



TECHNICAL REPORT 0-6877-2
TXDOT PROJECT NUMBER 0-6877

Communications and Radar-Supported Transportation Operations and Planning (CAR-STOP) – Phase 2: Final Report

Michael Motro
Taewan Kim
Rahi Kalantari
Jinsoo Park
Saharsh Oza
Joydeep Ghosh
Anum Ali
Preeti Kumari
Vutha Va
Nuria G. Prelicic

Robert Heath W. Jr.
Patricia S. Lavieri
Felipe F. Dias
Sebastian Astroza
Pragun Vinayak
Gopindra S. Nair
Joseph Hutchinson
Aupal Mondal
Chandra R. Bhat

January 2019; Published July 2019

<http://library.ctr.utexas.edu/ctr-publications/0-6877-2.pdf>



Technical Report Documentation Page

1. Report No. FHWA/TX-18/0-6877-2		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Communications and Radar-Supported Transportation Operations and Planning – Phase 2: Final Report			5. Report Date January 2019; Published July 2019		
			6. Performing Organization Code		
7. Author(s) Michael Motro, Taewan Kim, Rahi Kalantari, Jinsoo Park, Saharsh Oza, Joydeep Ghosh, Anum Ali, Preeti Kumari, Vutha Va, Nuria G. Prelicic, Robert Heath W. Jr., Patricia S. Lavieri, Felipe F. Dias, Sebastian Astroza, Pragun Vinayak, Gopindra S. Nair, Joseph Hutchinson, Aupal Mondal, Chandra R. Bhat			8. Performing Organization Report No. 0-6877-2		
9. Performing Organization Name and Address Center for Transportation Research The University of Texas at Austin 3925 W Braker Ln, 4 th Floor Austin, TX 78759			10. Work Unit No. (TRAVIS)		
			11. Contract or Grant No. 0-6877		
12. Sponsoring Agency Name and Address Texas Department of Transportation Research and Technology Implementation Office P.O. Box 5080 Austin, TX 78763-5080			13. Type of Report and Period Covered Technical Report September 2016–February 2019		
			14. Sponsoring Agency Code		
15. Supplementary Notes Project performed in cooperation with the Texas Department of Transportation and the Federal Highway Administration.					
16. Abstract This project designs a conceptual framework to harness and mature wireless technology to improve transportation safety, with a focus on frontal collision warning/collision avoidance (CW/CA) systems. The framework identifies components of the technology and its capabilities, and how these components can be integrated to improve transportation safety.					
17. Key Words Connected vehicles, traffic safety, machine learning			18. Distribution Statement No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161; www.ntis.gov.		
19. Security Classif. (of report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of pages 188		22. Price	



**THE UNIVERSITY OF TEXAS AT AUSTIN
CENTER FOR TRANSPORTATION RESEARCH**

Communications and Radar-Supported Transportation Operations and Planning (CAR-STOP) – Phase 2: Final Report

Michael Motro
Taewan Kim
Rahi Kalantari
Jinsoo Park
Saharsh Oza
Joydeep Ghosh

Anum Ali
Preeti Kumari
Vutha Va
Nuria G. Prelcic
Robert Heath W. Jr.

Patricia S. Lavieri
Felipe F. Dias
Sebastian Astroza
Pragun Vinayak
Gopindra S. Nair
Joseph Hutchinson
Aupal Mondal
Chandra R. Bhat

CTR Technical Report:	0-6877-2
Report Date:	January 2019; Published June 2019
Project:	0-6877 Phase 2
Project Title:	Communications and Radar-Supported Transportation Operations and Planning (CAR-STOP) – Phase 2
Sponsoring Agency:	Texas Department of Transportation
Performing Agency:	Center for Transportation Research at The University of Texas at Austin

Project performed in cooperation with the Texas Department of Transportation and the Federal Highway Administration.

Center for Transportation Research
The University of Texas at Austin
3925 W Braker Ln, 4th Floor
Austin, TX 78759

<http://ctr.utexas.edu/>

Disclaimers

Author's Disclaimer: The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official view or policies of the Federal Highway Administration or the Texas Department of Transportation (TxDOT). This report does not constitute a standard, specification, or regulation.

Patent Disclaimer: There was no invention or discovery conceived or first actually reduced to practice in the course of or under this contract, including any art, method, process, machine manufacture, design or composition of matter, or any new useful improvement thereof, or any variety of plant, which is or may be patentable under the patent laws of the United States of America or any foreign country.

Notice: The United States Government and the State of Texas do not endorse products or manufacturers. If trade or manufacturers' names appear herein, it is solely because they are considered essential to the object of this report.

Engineering Disclaimer

NOT INTENDED FOR CONSTRUCTION, BIDDING, OR PERMIT PURPOSES.

Project Engineer: Chandra Bhat
Professional Engineer License State and Number: Texas No. 88971
P. E. Designation: Research Supervisor

Acknowledgments

The authors would like to thank the undergraduate UT students that helped with testing and demonstrations: Daniel Guerrero, Justin Kinne, Aaron Logan, Sean Trembley, Jake Nimergood, Andrew Graff, Rachel Cooper, Andrea Vickers, Lauryn Spearing, and Christina Edgar. Graduate UT students Kevin Jo, Nitin Myers, and Javier Rodriguez also assisted in demonstrations.

The authors would also like to thank several CTR research staff members who helped throughout multiple steps of this project: Heidi W. Ross, Brandy Savarese, Jen Duthie, and James Kuhr.

There were three UT Electrical Engineering Senior Design projects that used CAR-STOP equipment. These projects performed work related to CAR-STOP, some of which is reported in this document. Tony DePalatis, Chris Duvarney, Steven Flolid, Nico Garofano, Adeesh Jain, and Zeyi Lin worked on “Object Detection Using Sensor Fusion in Vehicles.” Javier Zepeda, Matthew Edwards, George Tang, Gaurav Nagar, Hari Korosu, and Ben Miller worked on “CAR-STOP Collision Detection System Using the Nvidia Drive PX2 Platform.” Aditya Kharosekar, Chang Park, Evan Reid, Roberto Rioja, Steven Han, Trevor Eggenberger, and Vivian Tan worked on “Collision Warning System with Lidar and Camera.”

Table of Contents

Chapter 1 Introduction	1
Chapter 2 Implementation of Collision Avoidance Technology	4
2.1 Test Vehicles and Hardware	4
2.1.1 Sensing Devices	4
2.1.2 Vehicle Setup	7
2.3 Prototype Software	8
2.4 Prototype Computation	9
2.4.1 Parallel Computation	9
2.4.2 Alternative Platforms	11
Chapter 3 Perception Framework	13
3.1 Probabilistic Multi-Object Tracking	13
3.1.1 Explanation and Motivation	13
3.1.2 Tracking Formats Chosen	14
3.2 Object Representation	15
3.2.1 Host Vehicle	15
3.2.2 Relevant Objects	15
3.3 Occlusion	16
3.3.1 Partial Occlusion	16
3.3.2 Full Occlusion	17
Chapter 4 Perception with Specific Sensors	19
4.1 Vehicle Self-tracking	19
4.1.1 Location context	21
4.2 Ground Truth Methods	22
4.2.1 PPP-GNSS	22
4.2.2 Camera Localization	22
4.2.3 Overlooking Tower	23
4.3 V2V	25
4.4 Detecting and Tracking Relevant Objects	26
4.4.1 Radar	27
4.4.2 Camera	28
4.4.3 LIDAR	29
Chapter 5 Perception by Sensor Fusion	33
5.1 High-level Fusion with Correlation between Sensors	33
5.2 Low-level Fusion	35
5.2.1 Low-level Sensor Fusion of Camera and Radar	35

5.2.2 Shallow Low-Level Camera – LIDAR Fusion	37
5.2.3 Deep Camera-LIDAR Fusion – SSD-Fusion.....	38
5.3 Camera-LIDAR Calibration	42
5.3.1 Manual Sensor Calibration.....	42
5.3.2 Automatic Sensor Calibration.....	44
Chapter 6 Application to CA/CW	47
6.1 Prediction	47
6.1.1 Long-term Prediction based on Behavior	47
6.2 Collision Detection	50
Chapter 7 Distributed Decision-Making for Cooperative Driving	51
7.1 Previous Work and Goals	51
7.2 Model Improvements	51
7.2.1 Function Approximation	51
7.2.2 Deep Reinforcement Learning	52
7.3 Robustness to Sensor Error.....	53
7.4 Cooperative Highway Driving.....	54
7.5 Applying COIN to Real Scenarios.....	55
Chapter 8 DSRC-aided mmWave Vehicular Communication	60
8.1 A Machine Learning Approach to DSRC-aided Beam Alignment	61
8.1.1 System Model and Problem Description	61
8.1.2 Overview of DSCR-aided Beam Alignment Framework	63
8.1.3 Learning-to-rank Beam Recommendation.....	64
8.1.4 Online Learning for Beam Recommendation	67
8.2 Vehicular mmWave Reflection Measurements	71
8.2.1 Motivation.....	71
8.2.2 Reflection Measurement: Equipment and Setup.....	72
8.2.3 Measurement Results	74
8.2.4 Summary	77
Chapter 9 Radar-aided mmWave Infrastructure Communication	78
9.1 Introduction.....	78
9.2 Radar-aided V2I mmWave Communication	78
9.2.1 Positioning Using BS Mounted Radar	78
9.2.2 Radar-communication Congruence and Radar-based Precoding.....	80
9.2.3 Radar-aided 5G NR.....	82
9.3 Radar Equipment and Measurements	86
9.3.1 Equipment Purchases	86
9.3.2 Radar-communication Congruence Measurements	90
9.4 Summary.....	92

Chapter 10 Advanced Techniques to Further Enhance Joint mmWave Radar and Communication System.....	93
10.1.1 Introduction.....	93
10.1.2 Modify IEEE 802.11ad for Better Radar and Communication Performance	94
10.1.3 Designing New Joint Waveforms and Signal Processing Algorithms.....	98
10.1.4 Prototyping a Combined Communication and Radar Car-to-car Link	102
Chapter 11 Data Collection and Analysis Methodology	104
11.1 Equipment.....	104
11.1.1 Equipment Specifications	105
11.2 System Performance Measures	106
11.3 Experimental Design.....	106
11.4 Overview of Modeling Approach	108
Chapter 12 Urban Intersections	110
12.1 Identification of Primary Factors and Factor-levels and Identification of Test Bed Locations.....	110
12.2 Planning and Execution of Data Collection.....	121
12.3 Modeling Framework	125
12.3.1 Pooled Linear Model.....	125
12.3.2 Multi-level Linear Model.....	126
12.3.3 Pooled Discrete Model (Probit)	127
12.3.4 Multi-level Discrete Model (Probit)	127
12.3.5 Results.....	128
Chapter 13 Overtaking Maneuvers on Rural Roads.....	129
13.1 Experimental Design.....	129
13.2 Data Collection	131
13.3 Data Processing.....	134
13.4 Performance Evaluation.....	134
13.5 Conclusions.....	136
Chapter 14 Pedestrian Detection.....	138
14.1 Experimental Design.....	138
14.2 Data Collection	139
14.3 Performance Evaluation.....	140
14.3.1 Performance Models	140
14.3.2 Results.....	141
14.4 Conclusions.....	143
Chapter 15 ADAS Application Tests.....	144
15.1 Application 1: Construction Zone.....	144

15.1.1 Goals of the Demo	144
15.1.2 Modules Developed for the Road Construction Demo	145
15.1.3 Presentation of the Demo	149
15.2 Application 2: Pedestrian Crossing.....	149
15.3 Application 3: Highway Merging	150
Chapter 16 Conclusion	152
References	157
Appendix A – Vehicle Specifications.....	165
Appendix B – Pedestrian Ground Truth	167
Appendix C – Demonstration Videos.....	170

List of Figures

Figure 2.1: Pictures of the three project vehicles.....	4
Figure 2.2: Diagram of the steps of CW/CA	9
Figure 2.3: Snapshot of the MOT Challenge’s fourth video	11
Figure 2.4: Screenshot of modified DriveWorks display, with bird’s-eye view LIDAR points over the video with detected objects marked. The bottom left states the velocity of each object.	12
Figure 3.1: The parameters of a detected object.....	16
Figure 3.2: Example of partially occluded measurement, with minimum possible and maximum possible rectangles denoted with dashed lines. The gray regions are the regions that are undetectable due to the other objects.	17
Figure 4.1: Comparison of raw GNSS measurements (dotted) to tracked vehicle positions (green line), as well as imputed speed and heading.....	20
Figure 4.2: IMU readings from an urban road (left) and rural road (right)	21
Figure 4.3: ORB-SLAM on collected vehicle video: (a) in an unexplored area, (b) after returning to an explored area.	23
Figure 4.4: Pedestrian detection using tower camera	24
Figure 4.5: Overhead view of the experiment. Tower (far left - green), pedestrians (also on the left, red), and the approaching vehicle (far right - blue).....	24
Figure 4.6: Pedestrian detection using camera in approaching vehicle	24
Figure 4.7: Two examples of a vehicle tracking itself (green line) and being tracked by another vehicle via DSRC (red line). In the lower example, communication had only just been established—thus the speed is set to zero until more communication is received, and somewhat inaccurate in the short period afterward.	26
Figure 4.8: Car crossing in front, and four corresponding radar points.....	27
Figure 4.9: How cameras may determine an object’s distance, if elevation is known.....	28
Figure 4.10: Checkerboard pictures used to determine internal camera parameters	29
Figure 4.11: Diagram of LIDAR preprocessing. The right segment is partially occluded, so its length is unknown.....	30
Figure 4.12: Preprocessed LIDAR measurements superimposed onto images. Top: Pedestrian’s arm is segmented as a separate measurement. Bottom: Two pedestrians are merged into a single measurement (middle pair, lower laser).	31
Figure 4.13: Picture of LIDAR with bugs attached, taken October 8, 2017.....	32
Figure 5.1: Two example simulations with correlated sensor errors	34
Figure 5.2: Error of trackers on second sensor correlation simulation.....	35
Figure 5.3: Block diagram of the camera-radar sensor fusion system.....	36
Figure 5.4: Screenshot of the real-time sensor fusion system. Plot at the bottom left shows the detected radar points in front of the vehicle with clustered objects.....	37

Figure 5.5: Diagram of shallow camera-LIDAR fusion process	38
Figure 5.6: Framework of SSD-Fusion.....	39
Figure 5.7: Measurements from different sensors/fusions at a single timestep, camera view.....	40
Figure 5.8: Measurements from different sensors/fusions at a single timestep, top view	41
Figure 5.9: Example of bad calibration between a frontal camera and a rotational 64-layer LIDAR (KITTI dataset). Colored dots represent different distances measured by the LIDAR projected to the image’s coordinate system.....	42
Figure 5.10: The room used to calibrate sensors to create the KITTI dataset [5.10].....	42
Figure 5.11: Screenshot of the project calibration tool.....	43
Figure 5.12: Example of calibration results from the collected data showing the limit of hand-tuned approach. Logitech Pro Webcam C920 is used for the frontal camera and Quanergy M8 with eight layers is equipped as a rotational LIDAR sensor.....	44
Figure 5.13: Overall architecture and flow of the RegNet where inputs are images, depth point clouds, and initial calibration matrix <i>H_{init}</i> [5.13].	45
Figure 5.14: Example of calibration results using RegNet on KITTI dataset: (a) randomly de-calibrated sensors (b) true calibration (c) calibration using the output of RegNet.	46
Figure 6.1: Example trajectories from each extracted motion pattern.....	48
Figure 6.2: Predictions on pedestrians using the extracted motion patterns.....	49
Figure 6.3: Results of the vehicular collision detection study, with relative False Negative costs of 1, 10, and 100	50
Figure 7.1: The first architecture of the deep reinforcement learning model	53
Figure 7.2: Snapshot of simulated highway scenario	55
Figure 7.3: Scenario 1 screenshot	56
Figure 7.4: Scenario 2 screenshot	57
Figure 7.5: Scenario 3 screenshot	57
Figure 7.6: Scenario 4 screenshot	58
Figure 7.7: Scenario 5 screenshot	59
Figure 8.1: Illustration of the cellular infrastructure supporting several 5G V2X use cases enabled by mmWave communication.....	60
Figure 8.2: Simulated urban street canyon environment. The target vehicle is on the left lane at a longitudinal distance <i>d_l</i> from the RSU. The target vehicle has a roof-mounted antenna at 1.5 m high, and the RSU has its antenna at 7 m high.	62
Figure 8.3: (Left) A UPA with <i>N_x</i> by <i>N_y</i> antenna elements. (Right) The beam codebook for an 8x8 UPA. The beams are separated by their half-power (or 3 dB) beamwidth. We assume there is a grounded backplane so that there is no radiation toward the back (-z direction is this case). There are 87 beams to cover the entire field of view of this 8x8 UPA.	62
Figure 8.4: An illustration of the DSRC-aided mmWave beam alignment framework	63

Figure 8.5: Top- Nb recommendation beam pair recommendation system	66
Figure 8.6 Misalignment probability of the LtR beam recommendation method when using different kernels as compared to the baseline from [8.12]	67
Figure 8.7: A flowchart of the proposed online learning for beam pair recommendation	69
Figure 8.8: Flowchart illustrating the risk-aware greedy UCB algorithm	70
Figure 8.9: (Left) The effect of the two main parameters (Nb and Trisk) of the risk-aware greedy UCB algorithm. (Right) Performance comparison of greedy UCB algorithm with and without risk-awareness.	71
Figure 8.10: Horn antennae used in the measurement. (Left) 40° beamwidth horn antenna; (Right) 10° beamwidth horn antenna.	72
Figure 8.11: Reflection measurement setup. The measurements were conducted on the roof of a garage on campus.	73
Figure 8.12: Detailed geometry of the measurement setup	75
Figure 8.13: CIR tap power when rotating the receiver. Both the transmitter and receiver are at 0.5 m high.....	76
Figure 8.14: CIR tap power when rotating the receiver. The receiver is 0.5 m high, and the transmitter is 1.0 m high.	77
Figure 9.1 The V2I communication system with BS-mounted radar	79
Figure 9.2: The spectral efficiency vs transmit power, with 64 BS antennas and vehicle speeds of 140 kmh and 60 kmh.....	80
Figure 9.3: Relative path gain in both communication and radar signals.....	81
Figure 9.4: The performance result of the proposed beam alignments: (left) the case of left antenna array; (right) the case of top antenna array	82
Figure 9.5: Simulation scenario using a realistic mmWave channel model for V2I	84
Figure 9.6: Misdetection probability of the exhaustive search, position-aided search, and radar-aided covariance estimation search	85
Figure 9.7: The spectral efficiency of the proposed radar-aided covariance approach, exhaustive search, and position-aided exhaustive search	86
Figure 9.8: MmWave radar target simulator architecture.....	87
Figure 9.9: Radarbook (left) and Radarlog (right).....	89
Figure 9.10: The RSU (left), vehicle (center), and zoomed in corner reflector and Radarbook on the vehicle (right)	90
Figure 9.11: The radar-communication congruence measurements completed at four different locations on the vehicle (i.e., receiving tripod)	91
Figure 9.12: The radar and communication spatial spectrum.....	92
Figure 10.1: A multi-target scenario with two vehicles, namely vehicle R and vehicle T, within the mainlobe of the TX beam at the source vehicle. Both the vehicles are slightly separated in range, relative velocity, and direction with respect to the source vehicle.....	95

Figure 10.2: Mesh plot of the estimated 3D plot of the normalized-delay Doppler map shows that there are two dominant reflections from the recipient vehicle present at the 50th Doppler bin. Due to the fifty-eight times narrower mainlobe width in the Doppler domain as compared to the first scenario, the velocity resolution is 0.59 m/s in this simulation.....	95
Figure 10.3: Frame structure of an adaptive waveform design with variable preamble duration	96
Figure 10.4: Trade-off between radar CRB and communication MMSE with preamble duration	97
Figure 10.5: An illustration of an IEEE 802.11ad-based joint system of SRR with a wide FoV and V2V communication with a narrow FoV.....	97
Figure 10.6: Optimized communication rate decreases with target distance.....	98
Figure 10.7: Optimized radar recovery rate decreases with target distance	98
Figure 10.8: Uniform pulse approach, where a CPI consists of M equi-spaced frames placed at Nyquist rate. Here, each frame contains fixed preamble and data lengths.....	99
Figure 10.9: Virtual pulse approach, where a CPI consists of non-uniformly placed $M = M_1 + M_2$ frames.....	99
Figure 10.10: Optimized weighted average for different normalized communication weightings.....	100
Figure 10.11: Comparison between DoF-based optimization and CRB-based optimization for nested waveform	101
Figure 10.12: The source vehicle sends a waveform to the target communication receiver and use their echoes to estimate their locations. The TX front end (not shown) includes a single directional antenna with a notch in the direction of its receiver.	101
Figure 10.13: Optimized weighted average for different communication weightings	102
Figure 10.14: Proposed mmWave test bed set-up for joint communication and radar prototyping.....	103
Figure 10.15: Joint radar and communication test bed with horn antennas.....	103
Figure 12.1: Conflicting and non-conflicting maneuvers in four-way intersection.....	115
Figure 12.2: Blind approaches: (a) due to roadway geometry, and (b) situational condition.	119
Figure 12.3: Horizontal alignment of approach roads (orthogonal versus angled)	120
Figure 12.4: Vertical alignment (gradient) of approach roads (at-grade, upslope, downslope).....	120
Figure 12.5: Number of lanes on major approach road (two versus four versus six lanes).....	121
Figure 12.6: Example of maneuver plan indicating starting, ending, and waiting areas	122
Figure 12.7: Real-time visualization of the position of the experimental vehicles based on data exchanged through DSRC	122
Figure 12.8: Maneuvers considered when testing the effect of horizontal alignment	124
Figure 13.1: Diagram for overtaking maneuver	131

Figure 13.2: Location of data collection for overtaking maneuvers	132
Figure 13.3: Location of South Cone for overtaking maneuver	133
Figure 13.4: Location of North Cone for overtaking maneuver	133
Figure 13.5: Lead vehicle blocking view for overtaking maneuver	136
Figure 13.6: Percentage of hypothetical detection of oncoming vehicle as a function of sensor position—(a) Low acceleration: 1.0 m/s ² , (b) High acceleration: 2.5 m/s ²	136
Figure 14.1: Diagram of pedestrian detection scenario	139
Figure 15.1: Example of an experimental bird’s eye view display.....	144
Figure 15.2: Prototype sensing base station developed for the demo.....	145
Figure 15.3: Image of the developed app showing a bird’s eye view of the demo area.....	146
Figure 15.4: Designed simulated road construction area.....	147
Figure 15.5: Deployed road construction area	148
Figure 15.6: Image seen on computer screen during live demonstration of pedestrian crossing	150
Figure 15.7: Simplified images for highway merge demo	151
Figure 15.8: More informative visual display that overlays approaching vehicles on the road	151
Figure A.1: Diagram of wire connections.....	165
Figure A.2: Picture of car with sensors annotated	165
Figure A.3: Closer picture of camera attachment to front window	166
Figure A.4: Picture of adapters and connections in the back seat	166
Figure B.1: Vehicle camera snapshots 0.1 seconds apart	168
Figure B.2: Tower camera snapshots 0.1 seconds apart	169

List of Tables

Table 2.1: Sensing devices.....	7
Table 2.2: Breakdown of computation time for example LIDAR tracking.....	10
Table 2.3: Runtime of different computation methods on MOT Challenge video tracking.....	10
Table 5.1: Performance of tracking in first correlated sensor simulation.....	34
Table 7.1: Results of COIN simulations OMP	52
Table 7.2: Results of COIN simulations OLS	52
Table 7.3: Results of COIN with deep Q-Learning on four-way intersection.....	53
Table 7.4: Results of COIN retrained model under described scenarios.....	54
Table 7.5: Results of COIN simulations in different conditions	55
Table 7.6: Time of selected scenarios.....	56
Table 8.1: An example of an offline training data for a LtR beam recommendation.....	65
Table 8.2: Measurement configurations	74
Table 12.1: Exhaustive list of variables for urban intersection scenario	111
Table 12.2: Performance of instruments under different situations and conditions	117
Table 13.1: Summary results for overtaking maneuvers	134
Table 13.2: Overall performance of camera, radar and V2V communication.....	135
Table 14.1: Overall performance measures for pedestrian detection	140
Table 14.2: Performance modeling for “camera only” and “LIDAR only” alternatives.....	142
Table 14.3: Performance modeling for “shallow fusion” and “deep fusion” alternatives.....	142

List of Acronyms

AoA	angle of arrival
AP	average precision
CAN	Controller Area Network
CEF	channel estimation field
COIN	Collective Intelligence
CONOPS	Concept of Operations
CPI	coherent processing interval
CRB	Cramer Rao (lower) bound
CW/CA	collision warning/collision avoidance
DSRC	dedicated short-range communication
EIRP	equivalent isotropically radiated power
EMSE	excess mean squared error
FMCW	frequency modulated continuous waveform
Gbps	gigabit-per-second
GCP	Golay complementary pair
GCS	Golay complementary sequence
GNSS	Global navigation satellite systems
IMU	Inertial measurement units
LRR	long-range radar
LoS	Line of sight
LS	least squares
MMSE	minimum mean squared error
mmWave	millimeter wave
MRR	medium-range radar
MSE	mean square error
OBD	On-Board Diagnostic
PPP-GNSS	Precise point positioning global navigation satellite system
PR	perception-reaction
R-CNN	Region-based Convolutional Neural Networks
RF	radio frequency
ROI	region of interest
RSU	road-side unit

SBD	symbol boundary detection
SCNR	signal-to-clutter-plus-noise ratio
SCPHY	single carrier physical layer
SNR	signal-to-noise ratio
SRR	short-range radar
STF	short training field
V2I	vehicle-to-infrastructure
V2V	vehicle-to-vehicle
V2X	vehicle-to-everything
WLAN	wireless local area network

Chapter 1 Introduction

Historically, highway safety efforts have focused on passive safety systems (such as airbags and road barriers) that attempt to reduce the severity of crash outcomes. With the advancement of technology, however, efforts have expanded to designing advanced driver assistance systems (ADAS), which have great potential to proactively anticipate and prevent crashes. In order to inform a vehicle's driver (or autonomous system) of a potential crash, ADAS work to detect roadway objects and obstacles, predict the motion of the vehicles of interest, and compare the host vehicle's own course to that of nearby vehicles to prevent conflicts. Currently, most existing ADAS rely exclusively on data collected from sensors—radars, cameras, and LIDAR (Light Detection and Ranging)—to detect objects and generate predictions. However, sensors are limited, both in range and in the ability to detect objects outside of the sensor's line of sight. As such, vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, which can function along a longer range and are unconstrained by line-of-sight considerations, offer further opportunities for ADAS. By exchanging real-time position information, such communication systems can identify a potential collision course well before the vehicles enter each other's sensor range. The simultaneous use of communication and sensing technologies promises to accelerate the potential of ADAS.

The mechanisms of communication and sensing work in fundamentally different technical ways. This is because communication is used to precisely exchange information (for instance position and speed) between vehicles (or between vehicle and infrastructure), while sensors are used to estimate the location of objects and motion trajectory of other vehicles relative to the sensing vehicle. While sensor-based ADAS offer great advancement over current roadway safety systems, there are limitations in what such a system can offer. For instance, in the context of urban intersections, sensors may have limited line-of-sight, and sudden stops or accelerations between two vehicles with pure sensor-based ADAS can result in a downstream (in time) cascading negative safety impact on other vehicle movements. This is where communication, in addition to sensors, may offer a substantial benefit as it can send and receive information even when line-of-sight is blocked. Similarly, in overtaking maneuvers on rural roads sensors can help prevent some lower speed collisions between a passing vehicle and oncoming traffic. However, at high speeds, ADAS systems relying purely on sensors do not have enough time and space to brake within the window of time from detecting an oncoming vehicle. Alternatively, the longer range of communication between the passing vehicle and an oncoming vehicle can prevent vehicles from being on a collision course well before what can be discerned by radar and other sensing mechanisms alone. Communications may also aid sensing systems by providing a means for exchanging sensor data between vehicles or between vehicles and infrastructure.

At the same time, sensing mechanisms can also be beneficial to communications systems, especially because beam alignment (basically speaking, the direction in which communication signals are shot for optimal latency and packet conveyance), which can be very time-consuming

in a pure communications-based ADAS, can be done much more quickly by narrowing the cone of communications transmissions through information on the environment provided by sensors. Sensors are also essential to complement communications because pedestrians and bicyclists will not have the appropriate communication devices. In all of these instances, the optimal fusion of information from different sources is a key issue to preventing communication data overload or overall detection and prediction failures.

This project was proposed to advance research on the integration of cutting-edge developments in communications and sensing technology within transportation systems, with the goal to evaluate potential gains in traffic safety that the development and integration of these technologies may offer.

The project was divided into three main tasks:

- 1) Develop conceptual and functional frameworks for integrated wireless safety systems that incorporate information derived from both communication and radar platforms.
- 2) Advance and develop a new combined communication-radar paradigm for automotive applications using next-generation millimeter wave (mmWave) communication. Consider different performance metrics of radar and communication systems and investigate security issues in vehicular environments.
- 3) Conduct preliminary tests with real data, assess the performance of the proposed system, and develop a Concept of Operations (CONOPS) and requirements for technology deployment.

The first task of the project developed conceptual and functional frameworks for integrating communications and radar technologies. It begins in Chapter 2 by implementing a collision avoidance system using available technologies such as camera, radar, LIDAR and DSRC (dedicated short-range communication). In Chapter 3, we develop the perception framework, which handles how objects are represented, tracking and how occlusion is dealt with. Chapter 4 discusses how each specific sensor deals with perception. Chapter 5 contains the details surrounding the sensor fusion and its calibration. In Chapter 6 we apply these technologies and construct a collision warning/collision avoidance (CW/CA) system. Finally, Chapter 7 deals with the development of an algorithm for decision-making while driving and its application to five different scenarios.

The second task starts in Chapter 8 by providing a framework for DSRC-aided mmWave communication systems vehicular environments. Then, in Chapter 9, we develop a framework for radar-aided mmWave communication systems for vehicle-to-infrastructure (V2I) communications. Chapter 10 provides an overview of advanced techniques that can further enhance joint mmWave communication and radar systems.

In the final task, ADAS applications were developed for five specific scenarios: urban intersections, overtaking maneuvers on rural roads, pedestrian crossings or pedestrian/heavy areas,

highway merging from a standstill, and construction zones. We develop a data collection and analysis methodology to assess the performance of the ADAS, which is described in Chapter 11. We then collect and analyze field data to evaluate the ADAS performance in three of the applications: urban intersections in Chapter 12; overtaking maneuvers in Chapter 13; and pedestrian detection in Chapter 14.

Chapter 15 describes the design of fully operational collision warning prototypes for three of the ADAS applications: construction zones, pedestrian warnings, and highway merging. These prototypes were tested and demonstrated in controlled but realistic environments. These applications are the result of a collective effort involving all three tasks.

Finally, in Chapter 16 we present the overall conclusions of the project and discuss future research.

Chapter 2 Implementation of Collision Avoidance Technology

2.1 Test Vehicles and Hardware

Three vehicles were purchased to enable real-world testing: a Honda Accord sedan, a Toyota RAV4 SUV, and a Ford F-150 truck (Figure 2.1). This diversity of make and model enables the research team to test the sensing devices and computational units across a wide variety of common vehicles. The exact specs for these vehicles are given in Appendix A. In the sedan and SUV, connecting cables for the sensors were installed semi-permanently by ProCar Audio. This installation made the vehicles appear less cluttered, cleared space for spectating passengers, and protected against rainfall—our system is not waterproof, but will not be damaged by brief water exposure.



Figure 2.1: Pictures of the three project vehicles

2.1.1 Sensing Devices

The initial research proposal emphasized the comparison of automotive radar and DSRC vehicular communication. However, the field of autonomous vehicles has steadily advanced and it became clear that other sensors have unique strengths and may be readily available. With TxDOT’s approval, the research team purchased one of each type of common automotive sensor associated with advanced driver assistant systems.

Camera

Cameras are appealing sensors for several reasons, the first being their extremely low cost. High quality and weather-robust cameras are readily available for hundreds of dollars and even far cheaper cameras (such as those on cell phones) can still provide recognizable video. We purchased four Logitech C920 webcams. Despite having potential difficulties with lighting responsiveness, the webcam offers high resolution and easy real-time data access through a USB for a price of \$75 or less. It is reasonable to assume that current vehicular cameras used for forward collision warning or automatic cruise control are of a similar cost.

Another appeal of cameras is the intuitive connection to human drivers, who almost entirely rely on visual information to drive [2.1]. However, the translation from raw image data to meaningful information requires state-of-the-art algorithms and computational resources, with the

cost of the latter potentially offsetting the cheapness of the cameras themselves. These algorithms reliably locate objects within the image, but determining the distance and relative velocity of faraway objects is a separate and difficult task. This task requires either contextual knowledge (for instance the expected size of a truck relative to a car, or the relative distance of other objects such as buildings), or cooperation with other sensors. Finally, a single camera can only gather information in one direction. Multiple cameras are required for 360-view sensing. We have used a single camera at a time, focusing on applications where only objects in front of the vehicle are need to be detected.

Radar

Radar detects objects based on the reflections of a broadcast electromagnetic wave. Waves of the right frequency are less affected by precipitation or dust than cameras, and radar easily detects distances and even speeds of approaching objects. However, waves reflecting off an uneven ground or off other objects can cause false or misread object detections. Additionally, radar covers a similar horizontal range to cameras and an even smaller vertical range. The Delphi ESR is a standard radar for automated cruise control and was already purchased and used in Phase 1 of this project. The purchased version is the most open and flexible with respect to its output, but still returns processed detections rather than raw waveforms. This meant that our assessment of the radar's performance was constrained by the ESR's design choices.

LIDAR

Rotating LIDAR is capable of providing high-resolution distance data in 360 degrees, and is still instrumental to modern urban autonomous vehicles like those of Waymo and GM's Cruise. However, the LIDARs used by these companies cost tens of thousands of dollars, putting them far beyond the cost of other sensors and the reasonable cost for consumer vehicles. The cost of a LIDAR unit is roughly proportional to the number of vertical lasers – more lasers equates to more reliable detection at long distances and more information on the shape of an object. We purchased two Quanergy M8 units and a Velodyne VLP16, which have 8 and 16 lasers respectively. These sensors are relatively accessible at around \$5000 per sensor. However, they cannot capture the same level of detail that higher-laser LIDARs can, and so many conventional detection algorithms had to be modified for their use. LIDARs are also known have difficulty with weather and other forms of air obstruction [2.2].

DSRC

Dedicated short-range communications (DSRC) is the North American standard for V2V and V2I communication. As DSRC operates at the same frequency as Wi-Fi, the hardware necessary to send and receive communications is widespread and inexpensive. Basic GNSS (global navigation satellite systems, including the USA-created GPS) receivers are similarly widespread in modern cell phones and cars. However, these low-accuracy GNSS receivers can have

positioning errors of over five meters, which is problematic for most safety applications. The Cohda DSRC unit includes differential GNSS, which has an error within several meters [2.3]. Three on-board units and one roadside unit were purchased in Phase 1, and were used in this project for both V2V and V2I.

PPP-GNSS

The only way to assess the quality of detection for each sensor is to determine the position of each vehicle with a guaranteed high accuracy. The multiple approaches used in this project are described in Section 4.2. One option for precise vehicle positioning is PPP-GNSS (precise point positioning GNSS), the most accurate form of GNSS. We purchased four Emlid Reach receivers. One receiver is kept stationary, and the other three can precisely determine their positions while within a few hundred meters of the stationary receiver. Unfortunately, the Reach GNSS receivers we purchased were heavily affected by obstruction from trees and buildings, so they were only reliable positioning tools in certain situations.

IMUs

Inertial measurement units (IMUs) give information about acceleration and rotation, and like GNSS are readily available on most cars and smartphones. The Reach GNSS unit came with an MPU9250, which is a common low-cost IMU. High-accuracy IMUs [2.4] were substantially more expensive and were not considered. We primarily use IMUs to improve self-tracking at low speeds in urban environments, as they can for instance distinguish between GNSS drift and slow motion at a stoplight.







CAN Access

Autonomous vehicles are typically controlled through the vehicle's Controller Area Network (CAN). The configuration of this network is an international standard, but the accepted messages and commands vary from car to car and are typically not public. The exception is On-Board Diagnostic (OBD-II) information, which includes basic information used to diagnose or adjust the engine and other basic vehicular sensors. These basic sensors include the steering wheel angle, engine throttle, and speedometer reading. This information is similar to the acceleration and rotation returned by an IMU and could potentially replace IMUs.

While all vehicles provide this OBD-II information in some form, the way they do so can be kept informally secret. Some cars have access ports in hard-to-reach areas like the hood or multiple access ports. Some details of the communication format, such as baud rate, vary among car models and are not publicized for every model. We purchased two Panda OBD-II interfaces from comma.ai. We successfully connected to and read speed information from the Toyota RAV4, but were unsuccessful in reading anything from the 2017 Honda Accord. The next step to accessing OBD-II information, given insufficient online information, would be to take the car to a dealership for maintenance and ask/observe how they access this information.

Table 2.1 displays each variety of sensor and computer, its relative advantages and disadvantages, and the specific device purchased.

Table 2.1: Sensing devices

Sensor	Pros	Cons	Chosen Device	Price
Camera	Cheap, high-res data, details	No distance info, needs fast computing	Logitech C920 	\$75
Radar	Weather resistant	Noisy, low detail	Delphi ESR 2.5 	\$5000
LIDAR	360 view, high resolution	weather-prone	Quanergy M8, Velodyne Puck 	\$5000, \$8000
V2V	360 view, unobstructed, and accurate info	Needs other vehicles to have V2V and GNSS	Cohda MK5 (includes GNSS) 	(no fixed price)
PPP-GNSS	High-accuracy absolute position	Blocked by obstacles	EMLID Reach 	\$275 (locally operational unit, global units are > \$1000)
IMU	Only source of acceleration and turning info		MPU 9520	Included with EMLID Reach
Computer			Oryx Pro, Drive PX2 	\$1700, \$2500 (our offer from Nvidia, no fixed price)

2.1.2 Vehicle Setup

The computational requirements of each sensor vary substantially. The GNSS and DSRC devices performed most of the computational work internally, while LIDAR and especially cameras require computers that handle large input data and perform complex algorithms at a high

speed. As this project involved testing and demonstrating several setups, computers that were easily movable and removable were also useful. Three System76 Oryx Pros were chosen as some of the few laptops with high-quality GPUs and sufficient input ports. In addition, two Nvidia Drive PX2s, high-power GPU-based computers that are used by companies such as Tesla [2.5], were purchased—their use is discussed in Section 2.4.2.

As the cars purchased were hybrids, in-car battery access was more restricted than in traditional cars and could not source enough power for all the sensors during test runs. To overcome this, the team purchased extra deep-cycle vehicle batteries and placed them on the floor in front of the back seats of each vehicle. 400W adapters were sufficient to power all devices at once.

Some external equipment was also necessary for tests and demonstrations. Calibration checkerboards were hand-constructed to properly position sensors, as discussed in Section 5.3.1. A tower that connects to the trailer hitch of the pickup truck was used to emulate sensing infrastructure, for instance for the demonstration in Chapter 15, and to measure pedestrian positions as described in Section 4.2.3.

2.3 Prototype Software

The software for collision avoidance is divided into sensor code and application code. The sensor code handles each of the hardware devices covered above. If run on its own, it visually displays the information from a sensor. The application code covers the five collision warning applications that were the focus of the project: rural-road overtaking, pedestrian, intersection crossing, highway merging from a stop, and construction zones. Each application has its own folder with implementations of tracking, collision detection, and an audio-visual warning. The applications can be run as a simulation, or in real-time using the sensors and sensor code. The simulated version uses the same collision detection and display format as the real version and provides a useful test.

Most tasks in robotics can be divided into the subtasks of perception, prediction, and planning. Planning in CW/CA is generally simple, as the goal is simply to determine that the currently predicted future is dangerous enough to alert the driver (as diagrammed in Figure 2.2).

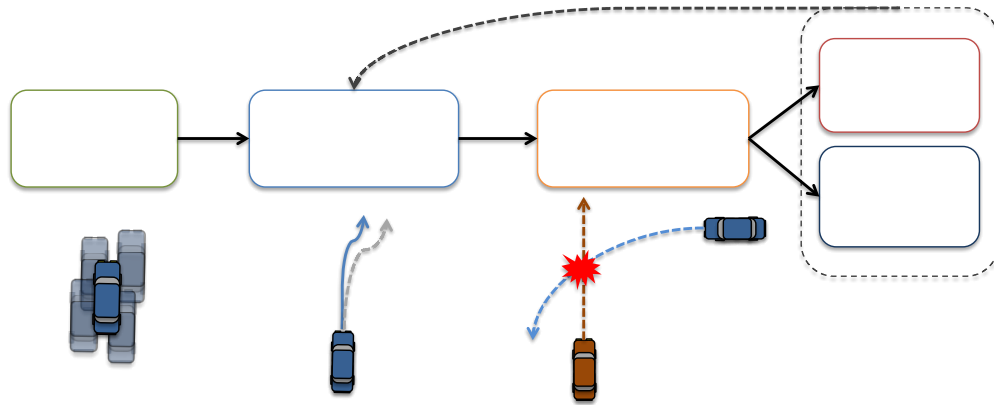


Figure 2.2: Diagram of the steps of CW/CA

2.4 Prototype Computation

The prototype collision avoidance system is performed primarily on the System76 laptops purchased for this purpose. The Python programming language was used with popular open-source packages such as OpenCV and Tensorflow [2.6]. The speed of the prototype varies depending on the sensor combination, the type of fusion, and the number of components passed by the tracker. It was designed to perform at 10 sensor updates a second or faster. Two design choices were investigated in detail but ultimately not adopted: parallel computation to increase speed and standard software platforms for easy modification and sharing.

2.4.1 Parallel Computation

By default, a program runs as a single thread and is performed in sequence by a single processor. Parts of the program that must be run simultaneously, such as storing the input from a sensor, are run as separate threads that communicate with the main thread. Much of the computation for multi-object tracking does not need to be performed sequentially. For instance, a matrix containing the likelihood of each object causing each measurement is used to determine which object is hypothesized to cause which measurement. This step is the most time-intensive for our LIDAR update algorithm, as shown in Table 2.2. Each object-measurement pair could be compared in parallel with no change to the rest of the update algorithm. Our laptop CPUs have four processors, so a highly parallel tracking algorithm could hypothetically run four times faster—in practice, the gain from parallelizing is less.

Table 2.2: Breakdown of computation time for example LIDAR tracking

Method	Parallelizable	Time (seconds)
Prediction	Y	15
Sensor Processing	N	22
Object-measurement Likelihood	Y	157
Data Association	N	4
Final Update	Y	7
Debug and Output	N	1

A GPU has hundreds or thousands of processors that perform simple parallel operations, such as operations that create or modify parts of an image. GPU computing is the key to rapid visual object detection and is used by our camera object detection system. We also implemented GPU-based multi-object tracking. The potential disadvantage of GPU computing is that the graphics processors are slower per sequential instruction than standard processors and have a smaller memory bank. Programs must ferry data back and forth between the CPU memory and GPU memory, incurring some time loss.

The different tracking implementations were tested on the fourth video of the Multi-Object Tracking Challenge [2.7]. Figure 2.3 shows a snapshot of this video. The video contains over thirty pedestrians at any point in time, roughly as much as a vehicle is expected to handle. Thus, the benefit of parallelization for this video should be roughly as much as for vehicular applications. The different computational implementations perform the same operations and reach the same estimates, the only difference being the speed at which they do so. Table 2.3 shows the runtime of a single update step for each method. Multi-threading on a computer with four processors improves the average tracking speed by roughly 50%. The prototype does not currently use multi-threading as our speed requirements are already satisfied, but multi-threading is a clear option should higher speed be needed. GPU computation is a less valuable option, especially considering that any camera-based object detection method will already be utilizing the GPU.

Table 2.3: Runtime of different computation methods on MOT Challenge video tracking

Method	Average Update Time (milliseconds)	Max Update Time (milliseconds)
Single-threaded CPU	28	69
Multi-threaded CPU	18	52
GPU	30	54

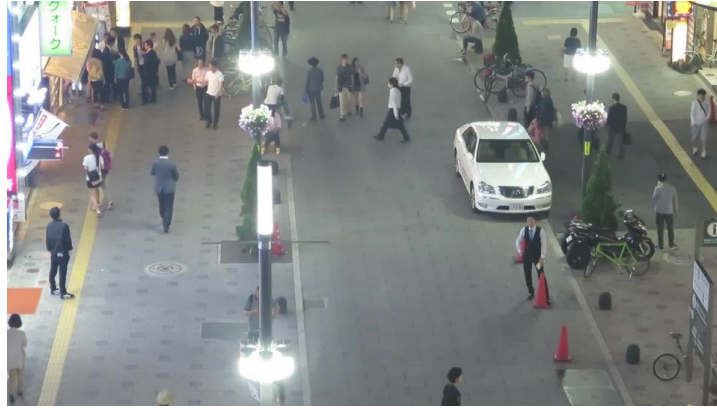


Figure 2.3: Snapshot of the MOT Challenge’s fourth video

2.4.2 Alternative Platforms

Several public codebases specifically for autonomous vehicles are in development. The educational startup Udacity has organized competitions for specific subtasks of autonomous driving [2.8]. Comma.ai has started OpenPilot, an open source framework for highway collision avoidance [2.9]. The Apollo project led by Baidu is the largest of these repositories, with over one hundred contributors and over one hundred partner companies [2.10]. It addresses all aspects of autonomous driving, from perception to planning and visual display, and at this point has several versions, each with its own focus application. Some companies are creating proprietary codebases. Nvidia has created a system called DriveWorks that facilitates use of autonomous technology on their car-oriented computer, the PX2 [2.11]. Two senior design projects for UT Electrical Engineering undergraduates in the 2017–2018 school year examined achieving collision avoidance with these public frameworks. One of the projects used the Apollo framework, while the other used Nvidia’s DriveWorks.

The Apollo framework, in its current form, proved difficult to use for two reasons. Firstly, most of the perception code is designed specifically for particular sensors. Rather than extending or modifying parts of the code to incorporate new sensors, researchers are currently encouraged to write their own perception code and connect it to the other parts of the framework. For collision warning applications, the perception subtask is a major part of the work, so the current Apollo has little to offer other than a standardized graphical display. The framework is also under rapid development—Version 1.0 was released summer of 2017, and four distinct versions have been released since then with thousands of small updates in between. The senior design team had to choose between modifying their code to accommodate the updated platform and maintaining obsolete and possibly erroneous versions. Ultimately, the collision warning system they developed did not actually use any part of Apollo. Apollo is currently not a practical platform for collision warning, but has evolved quickly enough to warrant future examination.

DriveWorks is a more stable framework and is marketed as being ready-to-use, even including on-call support from Nvidia. The senior design team completed a collision warning

assistant in this framework. Figure 2.4 shows a snapshot of their visualization, which is dense but combines all of the information DriveWorks offers. A variety of sensors can be practically incorporated into DriveWorks, though radar was still difficult to use. However, DriveNet is the piece of the DriveWorks that handles GPU computation such as camera-based object detection, and modifying or replacing this code was not well supported. The main draw of Nvidia's vehicle-oriented devices is their GPU and vision quality, so the limited flexibility of the DriveWorks in this regard does impede its usefulness.



Figure 2.4: Screenshot of modified DriveWorks display, with bird's-eye view LIDAR points over the video with detected objects marked. The bottom left states the velocity of each object.

Chapter 3 Perception Framework

We refer to the vehicle in which the CW/CA prototype is installed as the host vehicle, and the other traffic participants that must be avoided as objects (unless a specific type of object, such as a pedestrian, is the focus). Perception can be further divided into self-tracking, which determines the position and current motion of the host vehicle, and multi-object tracking, which locates all of the other objects relative to the host vehicle and determines their position and current motion. Multi-object tracking includes the challenges of detecting all the relevant objects (and conversely, not falsely detecting objects that are not actually present), determining these objects' location and shape, and determining their motion if relevant. These challenges are usually addressed simultaneously by maintaining a hypothesis of current objects and updating that hypothesis each time new sensor information is available. If the hypothesis includes the motion of all objects, then it is straightforward to predict their possible positions for the next several seconds. These possible positions can be used to detect a potential collision in the near future, which is the essential information for CW/CA.

The first step in perception is to process raw sensor data into useable information about the host vehicle and relevant objects. However, these algorithms are highly dependent on the sensor in question and on the desired format of the information. Thus, we skip the detection step and describe its various implementations in Chapter 4.

3.1 Probabilistic Multi-Object Tracking

3.1.1 Explanation and Motivation

Even with highly accurate sensors and predictive methods, the future state of a traffic environment will be uncertain. Drivers or pedestrians may change their motion arbitrarily, and new participants may enter the field of view. Similarly, all sensors suffer a certain degree of error in their ability to describe nearby objects. More cost-effective sensors generally have more error at a single point in time. Probabilistic multi-object tracking explicitly handles uncertainty in the objects and sensor readings by making the hypothesis a probability distribution across a set of objects. The sensor's reading at a single point in time is expected to follow a joint distribution with the current hypothesis, and updating the hypothesis with the given readings results in a less varied distribution, rather than a single answer. That is, given a hypothesis across the state of objects $P(X)$ and a model of the sensor output $P(Z|X)$, the update solves for $P(X|Z)$.

In probabilistic single-object tracking, X is a single object and Z is a measurement of that object. The field has been heavily studied for over fifty years, and is covered for instance in [3.1]. The most popular solution applies when $P(X)$ is a mixture of Gaussian distributions and $P(Z|X)$ is also Gaussian. One of the benefits of probabilistic tracking is that any number of sensors can be used at any time. Assuming that each sensor model is independent of the others, the hypothesis can be updated with each sensor, one at a time.

In multi-object tracking, X is a set of objects $\{x_1, x_2, \dots, x_n\}$, where even the number of present objects n is uncertain. The sensor output comes in a set of *measurements* $\{z_1, \dots\}$, wherein a single measurement (possibly) contains the sensor’s information for a single object [3.2]. Each object has a probability of detection $P_D(x)$, which is the probability that a measurement is created, and a distribution over a created measurement $P(z/x)$. There may also be false measurements, which are errors by the sensor and do not correspond to objects. These have a false distribution $P_F(z)$. The standard multi-object measurement model can be written:

$$P(Z|X) = \sum_{\text{matches}} \left(\prod_{\text{matched } i,j} P_D(x_i)P(z_j|x_i) \prod_{\text{unmatched } i} (1 - P_D(x_i)) \prod_{\text{unmatched } j} P_F(z_j) \right)$$

A specific matching between measurements and objects is referred to as a *data association*. The form of hypothesized objects, as well their motion and entry/exit from the scene, is the same for all sensors and described in Section 3.2. The measurement terms $P_D(x)$, $P(z/x)$, and $P_F(z)$ are specific to each sensor. One common concern, line-of-sight limitations for sensors, requires particular care and is discussed in Section 3.3.

3.1.2 Tracking Formats Chosen

The necessary complexity of the tracking framework can depend on the application. For instance, V2V applications do not need to worry about data association because DSRC messages include unique identifiers for each vehicle, and there is essentially no chance of accidentally getting a measurement that was not sent by a nearby vehicle (outside of malicious attacks). For the demonstration in Section 15.3, the V2V-based tracking framework stored a list of relevant vehicles and deleted one when it had not been detected for ten seconds. Each relevant vehicle was tracked accounting for GPS errors.

For most V2I or sensor-based detection and tracking applications, the project prototype uses a single hypothesis. This is represented as a multi-Bernoulli distribution for the set of tracked objects X . This stores $P(X)$ as a finite set of potentially existing objects, where the state of each object is independent [3.3, 3.4]. That is, we keep a fixed-size list of distributions $[P(x_1), P(x_2), P(x_3), \dots, P(x_N)]$ and existence probabilities $[r_1, r_2, r_3, \dots, r_N]$. Any number of objects $n \leq N$ can be tracked; if there is definitely only n objects, the existence probabilities past r_n will be very low. In practice, false detections add a certain number of possibly existing objects that should be considered. To update this type of distribution, a single data association is required. The theoretically best distribution update uses the marginal probability of each object matching each measurement, which can be exactly calculated [3.5] or quickly approximated [3.6]. However, we found that using the single most likely data association performed better—this has been corroborated by other researchers [3.7]. Using this data association technique, our update method is equivalent to a Multi-Bernoulli Mixture tracker [3.8] that maintains only a single mixture

component at a time. Single hypothesis techniques have been popular for autonomous vehicles such as [3.9, 3.10, 3.11, 3.12].

For particularly crowded scenes, the uncertainties in data association mean that the single most likely hypothesis may not ultimately be the correct one. Additionally, sometimes the distinction between two possible situations cannot be made yet, as with occluded objects. Multiple hypotheses can be stored at once while using the same list of object distributions $[P(x_i), \dots]$ but a more complex formulation of existence probabilities. The Multi-Bernoulli Mixture tracker [3.8] simply stores multiple lists of existence probabilities, each corresponding to a single hypothesis. When updating using a sensor's information, each of the current hypotheses must be considered and several of the most likely new data associations for each are calculated. However, single hypothesis tracking was considered sufficient for our current prototype applications.

3.2 Object Representation

An object's state includes its position on the ground, its shape, its current motion, and potentially other characteristics such as its identity (pedestrian versus cyclist, for instance). Shape and motion are generally unrelated as the shape can be assumed to stay constant. Boxes (or rectangles from a bird's eye view) are a popular choice of shape. Queries such as the nearest point on an object, the visible parts of an object, and whether two objects are occupying the same space (collided) are relatively easy to calculate for rectangles. Boxes also closely match the shape of cars, trucks, and large stationary objects such as road barriers or nearby buildings. Common relevant objects that are shaped differently—people, bicycles, and signs—are much thinner than cars. The error in meters of a rectangular approximation of these objects is therefore also small. Finally, objects that are only partially observed can be represented decently with rectangles.

3.2.1 Host Vehicle

In addition to detecting nearby vehicles and obstacles, an effective CA/CW system will need to know the heading, speed, and possibly driver action (wheel angle, braking) of its host vehicle. On the other hand, the host vehicle's shape and goal is known precisely beforehand. Thus self-tracking determines only the vehicle's current location and motion. We assume motion follows the bicycle model. This model assumes that vehicles move in periods of steady acceleration and angular velocity—roughly corresponding to the driver applying the gas pedal/brakes at a steady rate and keeping the steering wheel at a fixed angle. The same approach can be used for vehicles in V2V communication with the host, if they broadcast their shape (which is not part of the standard BSM).

3.2.2 Relevant Objects

There are many slight variations of how to model measurements from rectangular objects and handle the nonlinearity of orientation, as described in the research literature [3.9, 3.13, 3.14, 3.15]. We keep multiple potential samples for each object, each with a different fixed orientation.

Either one or two sides of the object are visible to a single sensor at a point in time. The height of each object is currently ignored, as it does not affect an object’s motion or potential danger. LIDAR is the only sensor whose detections may vary on object height. Relevant objects are assumed to be at least half a meter in height. The sensors and setup of the prototype will generally not detect objects smaller than this, nor do the applications usually require this. The velocity of the object is stored independently of the orientation, meaning each object has an east-west velocity and a north-south velocity (Figure 3.1). Vehicles can be assumed to move only in the direction of their orientation, but pedestrians do not have this restriction because their rectangular fit is a loose approximation. A global representation of vehicle velocity is ultimately about as accurate as an orientation-based one, while neither has the predictive power of a road-based model (as seen in Phase 1 of this project).

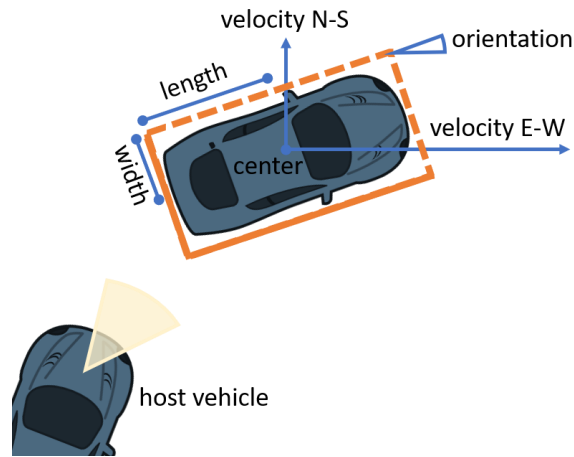


Figure 3.1: The parameters of a detected object

3.3 Occlusion

A key assumption of the standard measurement model is that, given a specific data association between hypothesized objects and observed measurements, each measurement is generated by its associated object with no dependence on other objects. This assumption is generally seen to conflict with the line-of-sight limitation of sensors. If an object is completely behind another object, or *occluded*, then it is unlikely to be detected by the sensor. This implies that the objects’ probabilities of detection $P_D(x)$ may be dependent on one or more other objects. Additionally, the sensor may detect part of an object but not its full extent.

3.3.1 Partial Occlusion

This phenomenon is considered for radar, LIDAR, and vision+LIDAR fusion methods. Vehicular communication returns complete information or nothing at all, and visual object detection methods do not find object distance accurately enough to determine whether there is

occlusion (though a human can recognize occlusion easily from images). Relevant objects are represented as rectangles in a bird’s eye view, so partially observed measurements are rectangles with an unknown extent. Figure 3.2 shows the representation of a rectangle measurement with minimum and maximum limits on its extent in every direction. This representation can conveniently limit objects to only be a certain size. To update objects, the midpoint between the minimum and maximum extent is used as a measurement of each edge, with a variance based on the difference between the two. This way, objects can still be represented with a singular (but uncertain) position and shape. For an object that is fully observed, the minimum and maximum extents will be nearly equal and the measurement is simply the detected rectangle.



Figure 3.2: Example of partially occluded measurement, with minimum possible and maximum possible rectangles denoted with dashed lines. The gray regions are the regions that are undetectable due to the other objects.

3.3.2 Full Occlusion

When an object is fully occluded, it will not be detected by a sensor. However, it may still be important to still store the object and predict its motion. Thus it is important for the multi-object tracking framework to reason about full occlusion. The typical formulation is to split the present relevant objects X into visible and occluded subsets X^V , X^U . Only visible objects may create measurements. The probability that a hypothesized object is occluded is determined by integrating across the other objects. The research works [3.16, 3.17, 3.18] all describe fast approximate methods for this formulation.

We discovered that a different descriptive model of occlusion is more realistic for many sensors, easier to compute, and compliant with the no-dependence assumption of the standard measurement model. Objects create measurements as in the standard measurement model, but the

resulting set of measurements Z is split into visible and occluded subsets Z^V, Z^U . This model is realistic for high-detail sensors whose measurements undergo heavy preprocessing, such as camera and LIDAR. This preprocessing often includes merging of overlapping measurements that are considered likely to have come from the same object, or removal of small measurements that may have been errors or are non-descriptive. Under this model, a hypothetical measurement is occluded if it is fully ‘behind’ any of the given measurements. The exact form of this check varies from sensor to sensor, but it is generally easier to perform than the equivalent check across hypothesized objects.

Chapter 4 Perception with Specific Sensors

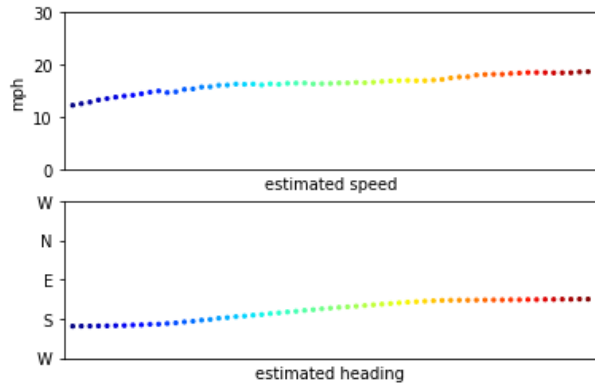
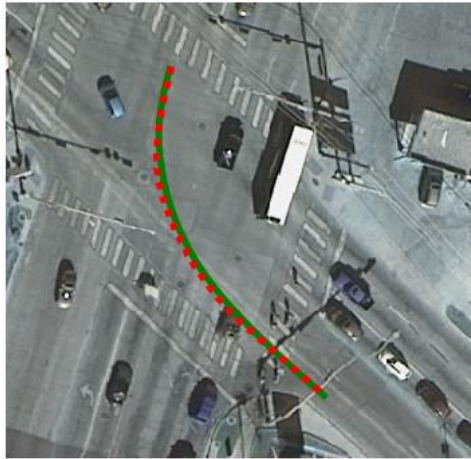
This chapter explains the methods developed to detect cars, pedestrians, or significant inanimate objects. Each sensor requires a different approach and is covered separately. Each sensor's subchapter also describes methods needed to set up the sensors and important considerations when using the sensor for detection.

4.1 Vehicle Self-tracking

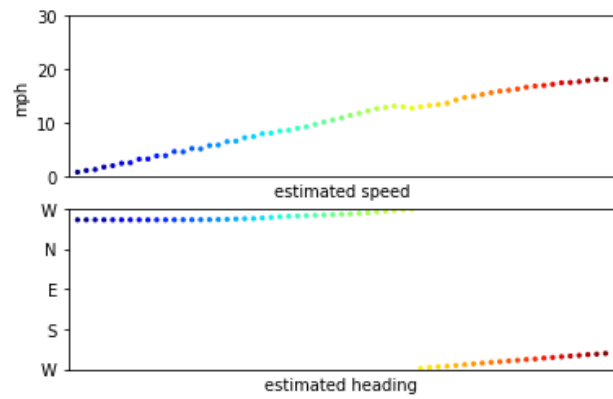
The host vehicle is tracked using GNSS location readings from the onboard Cohda DSRC and IMU motion information from the Reach unit. The Rao-Blackwellized particle filter single-object tracking algorithm offers high accuracy at a flexible speed. The tracking required some specialization depending on the application.

The application of the intersection collision avoidance should function with or without stuff. However, due to the amount of time needed to setup each test run, the research team only ran the algorithm at the start of the planned "maneuvers" of each test drive (details on these maneuvers are in Chapter 13). This often meant starting the algorithm as the vehicle was stopped at the intersection before the maneuver. Without dynamic car movements before the stop, determination of the heading (the direction the vehicle is facing) in these situations was difficult. The IMU's internal compass was often affected too strongly by the presence of other electronic devices to provide reliable heading information when the vehicle was stopped. As such, the team initiated the tracking algorithm at the most recent time before a maneuver that the vehicle was in motion. Additionally, the algorithm itself was altered to distinguish between turning and steady/stopped vehicles, so that the heading was kept constant during long periods of waiting at an intersection.

Figure 4.1 shows tracked positions, along with the estimated heading and speed, for a few example maneuvers. The background aerial images are from the US National Map database from 2011 [4.1]. The tracked position is given as a line, while the GNSS measurements are marked as colored dots for comparison. Note that the estimated center of the vehicle is tracked, while the GNSS antennae are to the side and back of the vehicles; hence the tracking lines are not completely aligned with the measurements.



(a)



(b)

Figure 4.1: Comparison of raw GNSS measurements (dotted) to tracked vehicle positions (green line), as well as imputed speed and heading

For the rural-road overtaking application, it was discovered that bumpy rides, caused by high speeds or less-maintained roads, increase the error of inertial motion detection. Figure 4.2 shows reported acceleration and angular velocity for ten-second periods on an urban road and high-speed rural road. Acceleration readings on the rural road are too varied to determine whether the vehicle is increasing or decreasing speed.

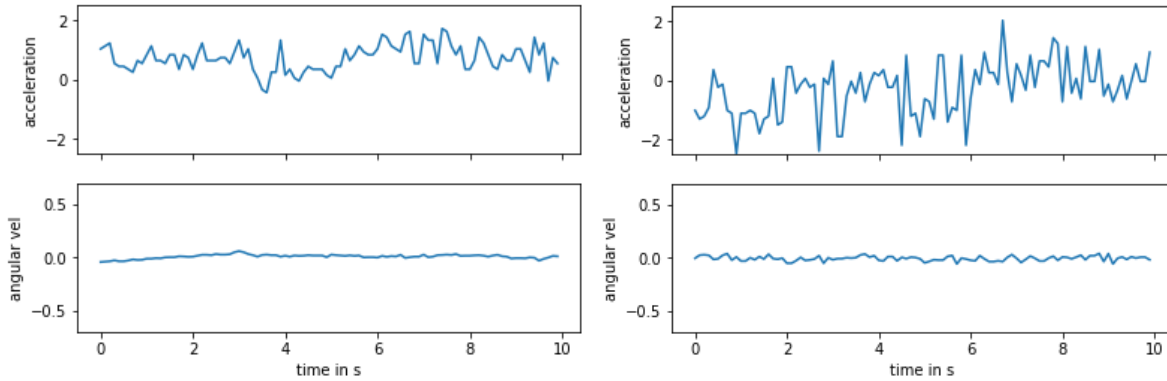


Figure 4.2: IMU readings from an urban road (left) and rural road (right)

4.1.1 Location context

Self-driving vehicles cannot operate without a high-resolution map covering all areas they might drive in. Collision avoidance technology does not make road decisions and can often function with low-resolution maps or without maps at all. Maps are not difficult to create for small areas, but difficult to create and maintain over large areas. Third-party map services like Google Maps API [4.2] are cost-effective solutions for large citywide or nation-wide mapping. This project did not use a third-party service, instead self-making maps for small test areas. The form of these maps varies by application.

- Overtaking – Maps are useful for correctly assessing oncoming vehicles on curved rural roads, although most drivers may consider overtaking on curved roads risky in the first place. These maps can be low-resolution because of the distances and speeds considered in this maneuver. Standard third-party services or even free crowd-sourced services like OpenStreetMap would be sufficient. This project tested only the overtaking warning on straight roads, with no map use.
- Highway merge – A map is useful to store the road curvature and the number of lanes. However, the map will likely only need to cover a small region near a designated merging zone, rather than a national or global map. Additionally, if the merge is taking place because of construction, the shape or number of lanes of the highway may be temporarily altered. Thus, a self-made map is more useful than a third-party map service. Our highway merge application requires a road model constructed out of a few key point coordinates, which can be obtained via GPS or a satellite image. Road models have been designed for a few example locations in Austin.
- Intersection – A map is useful to accurately characterize legal maneuvers such as left turns. For most applications, this map would need to be citywide if not nation-wide, and of higher resolution than for the overtaking maneuver, so third-party services are appropriate.

- Pedestrian – Even on normal roads, pedestrian crossing zones are not necessarily annotated on available maps nor strictly followed by pedestrians. For off-road locations such as construction zones, there will likely be no specific path taken by pedestrians. Thus, maps are not useful for this application.
- Off-road/construction zone – We assume that equipment and workers may be anywhere in the zone, so a map is not useful and was not created for this application. If a particular zone can benefit from a map of drivable or accessible areas, it would be simple to add this information to the infrastructure unit’s software.

4.2 Ground Truth Methods

Much of our analysis of the CW/CA applications was qualitative, but for certain tasks, a numerical verification of the perception algorithms was needed. Thus, we explored several ways to obtain highly accurate vehicle/relevant object positions. These methods differ from the perception methods discussed before in that they could be restricted to a small known area, and that they were only calculated after-the-fact on recorded data.

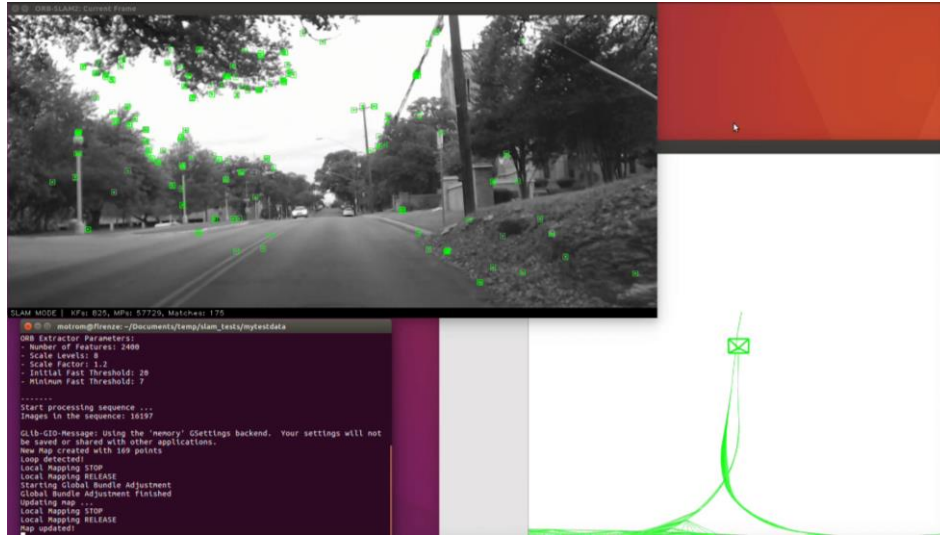
4.2.1 PPP-GNSS

The Emlid Reach units return normal GNSS output, latitude and longitude at five times per second. These position readings are more accurate because they are cross-referenced with a stationary Reach unit. The self-tracking algorithm is modified into a smoothing algorithm that bases each position estimate not only on past measurements but also on future measurements. Future measurements of up to one second were utilized. For the intersection application and tests, the IMU was used as in the self-tracking algorithm.

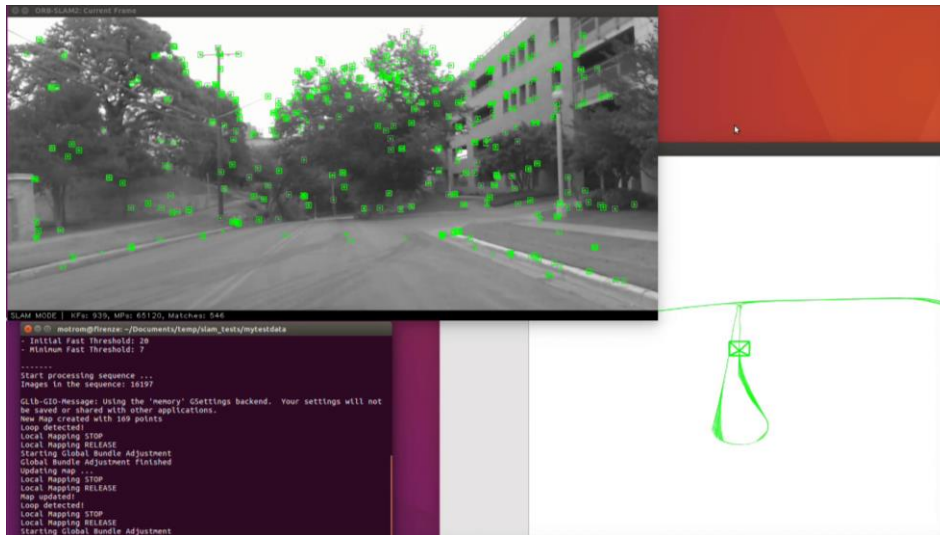
4.2.2 Camera Localization

Localization is a technique for self-tracking with a similar form to multi-object tracking. A large number of nearby objects are located, and the host vehicle’s motion is determined by assuming that most of these objects are stationary. Simultaneous localization and mapping (SLAM) is a long-term version of this technique that recognizes areas that have been seen before. Traditionally localization is performed with high-resolution sensors such as high-laser LIDAR or stereo cameras [4.3, 4.4, 4.5]. There are no conventional algorithms to perform localization or SLAM with the low-resolution LIDAR and radar used in this project. However, recently developed algorithms such as ORB-SLAM [4.6] perform successfully with a single camera. Figure 4.3 shows ORB-SLAM operating on recorded video from 27th Street in Austin. The right plot in each figure shows the estimated path of the vehicle, which at first is inaccurate. Once the vehicle turns and returns to a position it had previously seen, the estimates are corrected and appear to achieve lane-level accuracy. Localization techniques were ultimately not incorporated in our current prototype, as the achieved accuracy was not high enough without long-term location memory. Localization

may also fail when many moving vehicles or pedestrians are nearby. SLAM will however be useful for future CW/CA application testing, as it provides highly accurate ground-truth positions if applied in a limited, controlled area.



(a)



(b)

Figure 4.3: ORB-SLAM on collected vehicle video: (a) in an unexplored area, (b) after returning to an explored area.

4.2.3 Overlooking Tower

It is difficult to provide pedestrians with precise positioning devices. Therefore, stationary infrastructure was used to detect the pedestrians instead. We used a camera on top of a 2.5-meter tower to record video of the pedestrian crossing the road. Object detection code (SSD) was applied to the recorded video afterwards. The detections in video could then be transformed into locations

on the ground if the camera's orientation and the sloping/elevation of the ground were known. The vehicle approaching the pedestrian was tracked with our self-tracking algorithm utilizing post-processed GPS, as is performed in other tests. The ground positions of the pedestrians are converted to positions relative to the car's view. Figure 4.4, Figure 4.5, and Figure 4.6 show the three formats of pedestrian position. Minor details of ground truth generation are discussed in Appendix B.

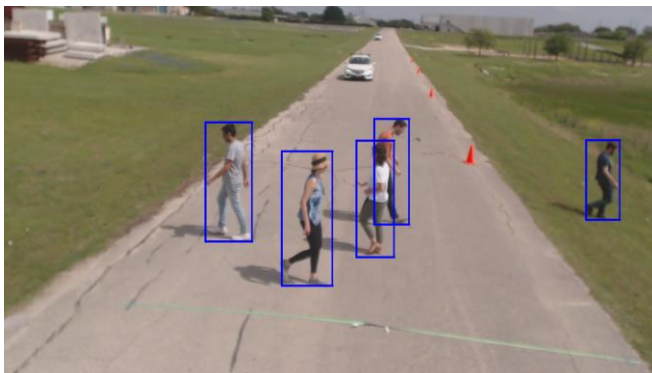


Figure 4.4: Pedestrian detection using tower camera

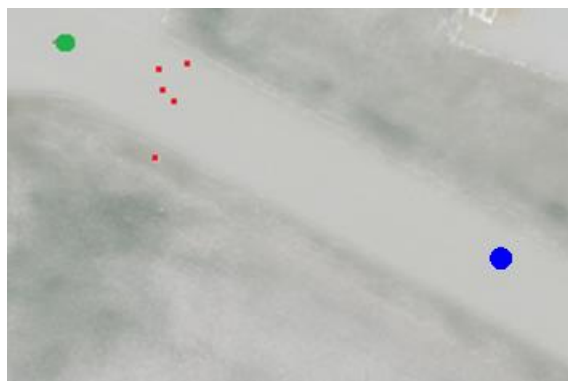


Figure 4.5: Overhead view of the experiment. Tower (far left - green), pedestrians (also on the left, red), and the approaching vehicle (far right - blue)



Figure 4.6: Pedestrian detection using camera in approaching vehicle

4.3 V2V

Most of the concern for vehicle-to-vehicle communication systems is on their completeness—that is, whether the adoption rate and the distance limits will ensure detection of every relevant vehicle to avoid collisions. In other ways, communication is an extremely simple and reliable technology: the chance of receiving a valid message out of thin air is essentially zero, meaning the reception of a message guarantees the presence of a vehicle. A well-designed system of unique IDs also ensures that any given message is attributed to the correct vehicle [4.7]. Thus, through monitoring of vehicle message broadcasts, standard single-target tracking algorithms like the one from the previous section (the Rao-Blackwellized particle filter) can be used to determine the heading and speed of each transmitting vehicle. The one notable difference between a vehicle tracking itself and tracking others via messages is that a series of messages must be analyzed to establish an accurate heading before course tracking and prediction can occur (in other words, prediction of an external vehicle movement cannot occur after receiving just one message). This may be somewhat negated if the broadcasting vehicle includes helpful information such as IMU readings or inferred heading and speed, but these are not guaranteed values. For this research, two types of V2V-based tracker were developed, one in which the complete tracked information of the other vehicle is received and one in which only GNSS positions are received. The former is an example of potentially beneficial communication not covered by the current BSMs. For the latter, the tracking algorithm was modified based on the number of messages received in the last ten seconds and each message was utilized as such:

- 1 Message: Place the vehicle at this GNSS position with no assumed movement but high uncertainty
- 2–3 Messages: Assume the vehicle is travelling in a straight line at a constant speed, infer the speed and heading from the points
- 4+ Messages: Perform tracking as in the previous section

Figure 4.7 shows an example of inaccurate V2V tracking that results when too few messages are received. Chapter 12 includes a detailed performance analysis for both variants of V2V tracking.

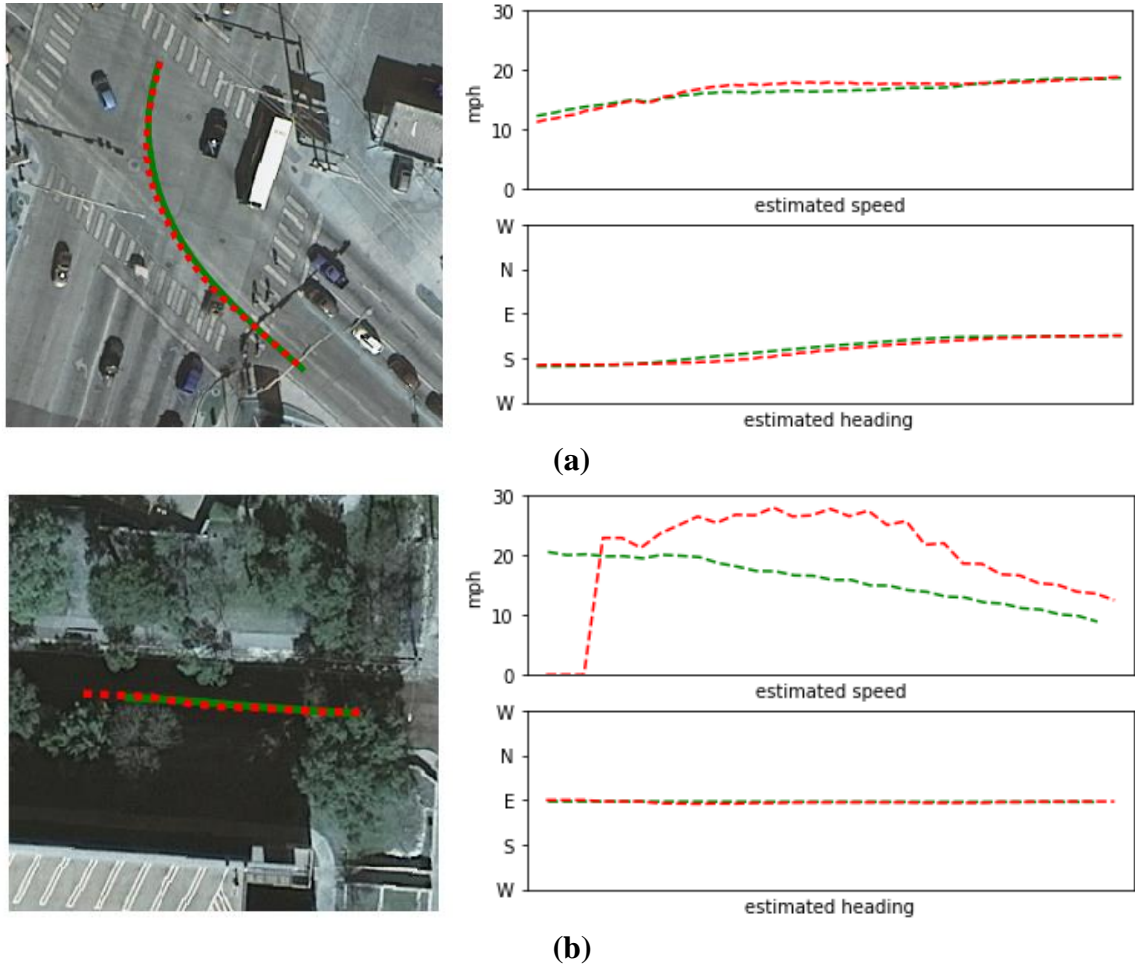


Figure 4.7: Two examples of a vehicle tracking itself (green line) and being tracked by another vehicle via DSRC (red line). In the lower example, communication had only just been established—thus the speed is set to zero until more communication is received, and somewhat inaccurate in the short period afterward.

4.4 Detecting and Tracking Relevant Objects

The tracking of other vehicles and objects is more challenging than the tracking of the host vehicle for several reasons. For instance, the number of objects present, if any, is unknown, and false detections from sensors may suggest the presence of objects that are not there. Another challenge is found when multiple objects are close by the sensing unit, the sensing unit must determine which measurement corresponds to which object before measurements can provide valuable information. Yet another challenge is that objects in the foreground can block all sensors' views of other objects.

To help overcome these challenges, this project combines communication with onboard sensing to work towards technology that will vastly improve roadway safety.

4.4.1 Radar

The Delphi ESR radar processes the reflected radio signal on its own and returns two-dimensional points corresponding to places of reflection (in other words, detected objects). Larger objects will likely be detected multiple times, as in Figure 4.8, and false and missed detections are frequent. Additionally, the Delphi unit returns measurements twenty times a second, while the other sensors in the prototype operate at ten times a second. Radar detections from two consecutive returns are combined, and then grouped into measurement-clusters. Specifically, the DBSCAN algorithm [4.8] clusters measurements based on distance between measurements. Each measurement-cluster is treated like a single measurement for the purposes of tracking and matched to an object. This approach does not distinguish between closely spaced objects well, as they may be grouped in the same measurement-cluster, but it works better on average due to mitigating false positives and negatives. Each point in a measurement-cluster is assumed to be generated from somewhere on the rectangular boundary of the object, with Gaussian error. It is not enforced that this boundary is the visible boundary of the car, because radar is known to occasionally reflect off the ground or nearby obstacles and hit the far side of objects [4.9]. Full occlusion is not strictly enforced for radar for the same reason. Instead, objects behind visible measurements have a certain probability, less than one, of being occluded.

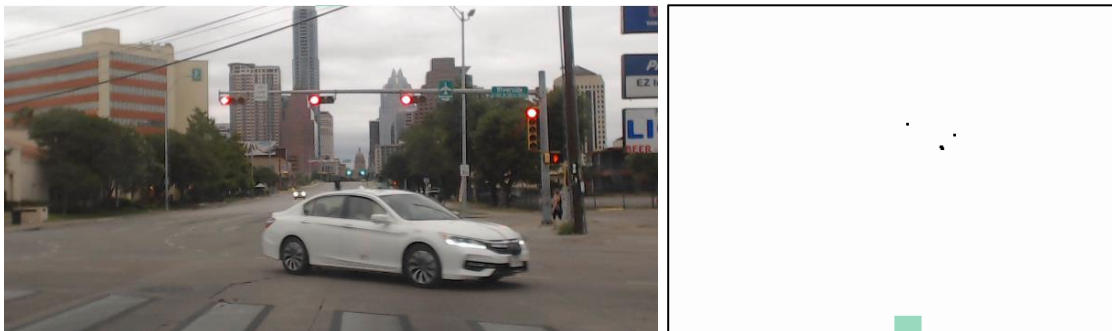


Figure 4.8: Car crossing in front, and four corresponding radar points

Radar units are more robust to weather and damage than LIDARs, as a thick, opaque cover can be used without blocking the sensor’s view. It should also be more robust to imperfect positioning or calibration, as the signals reflect off the ground and object height is not measured. If the radar were angled $.1$ radians downwards as opposed to directly horizontal, the ensuing error in position for a detected object would be at most half a meter. We have not explicitly tested the effect of poor radar calibration, however.

Radar clusters that were unlikely to be associated with any existing object are treated either as noise or as the first detection of a new object. The overall performance of the algorithm cannot be fully quantified, because we lack true position information for most objects around the vehicle, such as other vehicles in traffic or objects like trees/streetlights. Qualitative video of this and other detection tests is available in Appendix C.

4.4.2 Camera

Detecting objects such as vehicles or people within images is a classic problem in computer science, and deep convolutional networks represent the current state of the art. In Phase 1, a deep network was created in the Caffe software platform and used for joint LIDAR-camera object detection (see Phase 1 Final Report Chapter 3.3). In this phase, we used the You Only Look Once (YOLO) algorithm [4.10], which was trained on Microsoft’s Coco image database. This database includes thousands of annotated examples of cars, humans, buses, and bicycles and thus is more suited for the initial training of a vehicle detection algorithm than the data that we have collected [4.11]. The YOLO algorithm can operate on roughly three images per second on a decent CPU-based computer, but can handle ten to twenty images per second with a GPU, putting it at the necessary speed for vehicle safety applications. Version 2 was used for the pedestrian warning application, while the slightly faster version 3 [4.12] was used in the construction zone application. The low-level fusion method discussed in Section 5.2.3 uses a newly designed convolutional network, constructed in the TensorFlow [4.13] platform.

Note that visual object detection algorithms find the position of the object in the image—not their real position. Further context is needed to infer the distance of an object from its position or extent in an image. For instance, the angle of the bottom of the detected object (in image) can be used to estimate their position on the ground (Figure 4.9). This requires careful calibration of the camera and knowledge of the surrounding elevation. Its accuracy also depends on camera resolution and object distance: for our 720p camera, the error due to resolution alone is over 2 meters at 50 meters distance. This camera-only distance-estimation approach is tested on pedestrians in Chapter 14 and shown to be inferior to fusion with another sensor.

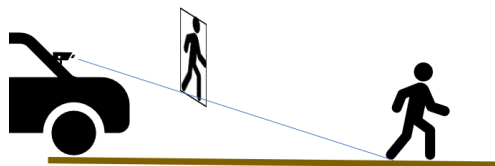


Figure 4.9: How cameras may determine an object’s distance, if elevation is known

Calibration

Any camera lens has a slightly different thickness corresponding to a slightly wider or thinner view, as well as some distortion. For cheaper cameras, the variation between lenses is significant enough that the values of focus and distortion should be calculated. This process called internal calibration, and was performed by recording pictures of checkerboards (Figure 4.10) and applying standard calibration code [4.14].



Figure 4.10: Checkerboard pictures used to determine internal camera parameters

External calibration instead refers to calculating a sensor’s pose, or exact position and angle on the vehicle. It is not discussed frequently in academic research or by autonomy-aiming companies, as carefully designed research prototypes or factory-produced autonomous vehicles will have precisely installed sensors. However, both after-market vehicular sensors and sensing infrastructure will require calibration, possibly more than once if the devices can be moved. This is necessary, for instance, for the distance estimation technique discussed above. Camera external calibration can be performed by setting up a test area with objects of known length, like the ground truth area in Section 4.2.3 and the construction zone application. The most challenging calibration for our applications was a camera-LIDAR calibration, where the camera’s pose was found relative to the LIDAR’s detected points. This step was necessary for performing fused camera-LIDAR tracking, and thus is covered in Section 5.3.

4.4.3 LIDAR

The way that LIDAR is processed typically depends on its type and resolution. Non-rotating LIDAR only provides information along a limited range. Alternatively, the high-resolution variants of fully rotating LIDAR provide too much data to be easily handled by a tracking algorithm. These are more often treated with image-style deep network object detection [4.15, 4.16]. The Quanergy M8 and Velodyne PUC units are fully rotating but only offer 8 and 16 vertical lasers respectively. Many of these lasers are aimed too high or low to detect grounded objects, so in practice a nearby pedestrian will be detected by 1 to 4 lasers. This is too few to apply to a traditional deep network.

Rectangular Segment Processing

We previously used raw LIDAR points as input to the tracking algorithm, as by Scheel et al. [4.17], but this method ran at less than a twentieth of the necessary speed. We instead preprocess the LIDAR output into object-like measurements, and then update the distribution based on these measurements. Much of the pre-processing can be performed as each packet of LIDAR output arrives, instead of all at once when a rotation is completed. First, contiguous segments of each laser’s returns are detected, and segments of three or more are treated as measurements. For a new measurement segment, the best fitting straight-line segment or rectangular segment is chosen using

a RANSAC minimum-area-rectangle approach, similar to [4.18] and [4.19]. The segment is classified as closed or open on each end, depending on whether the previous or following segments are directly in front of this segment. This determines whether the detected object is partially occluded such that the measurement does not cover the object’s full extent. Unlike with radar, it is assumed that the rectangle segments given by LIDAR correspond to the visible side of an object. Figure 4.11 illustrates the measurement-processing step.

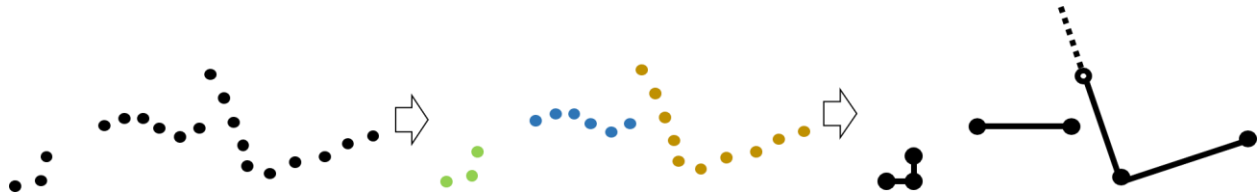


Figure 4.11: Diagram of LIDAR preprocessing. The right segment is partially occluded, so its length is unknown.

This method provides a much smaller set of relevant measurements to the tracking algorithm, while retaining most of the valuable information. There are a couple of practical issues with it. The first is that crowds of pedestrians are difficult to segment properly. Detected arms and legs may be classified as their own segments as seen in the upper image of Figure 4.12. However, nearby pedestrians may be merged into a single segment as seen in the lower image of Figure 4.12. This can be addressed by combining multiple laser’s detections. Because of the difference in vertical angle between two lasers, it is unlikely that both create two segments for a single pedestrian. Measurements that have been seen for more than one laser are considered definite measurements, while measurements that were only detected by one laser are possibly extraneous. The multi-object tracking paradigm can already account for less certain measurements by increasing the probability that these measurements are false positives.

A final challenge for LIDAR tracking is non-solid objects that may not be detected contiguously, such as bushes or a chain-link fence. While these objects do not need to be tracked carefully, the algorithm has no way of knowing this and wastes computational power tracking many small objects. Fusion with camera can solve this problem by specifically locating cars and humans.



Figure 4.12: Preprocessed LIDAR measurements superimposed onto images. Top: Pedestrian’s arm is segmented as a separate measurement. Bottom: Two pedestrians are merged into a single measurement (middle pair, lower laser).

Difficulties

The predominant cited weakness of LIDAR is not its cost but its weakness to weather or damage. The lasers used for sensing are weak, partially because at higher strengths they could blind an onlooker. It is commonly stated that rain will block LIDAR, and even exhaust from another car has caused false detections [4.20]. We have not tested the LIDAR in rain or heavy fog because our devices are not waterproof. We have found a similar, if less-mentioned, problem. While performing overtaking maneuver tests (as covered in Chapter 13), the LIDAR was plastered with small bugs, shown in Figure 4.13. This degree of obstruction did not have a noticeable effect on the LIDAR’s performance. However, this obstruction was caused by two hours or so of driving on a rural road. It is likely that LIDAR will require frequent cleaning if used in areas with swarming insects (like Texas [4.21]). Finally, LIDAR may disperse on and fail to detect semi-transparent or absorbent surfaces, such as car windows. This does not necessarily invalidate LIDAR as a sensor, as other parts of the vehicles are detected well—but this behavior should be kept in mind when using LIDAR to detect relevant objects.



Figure 4.13: Picture of LIDAR with bugs attached, taken October 8, 2017

Chapter 5 Perception by Sensor Fusion

Sensor fusion is a broad term referring to any use of multiple sensors to reach a single conclusion. While the form of each sensor's update is unique, they each update the same multi-object hypothesis. High-level fusion methods maintain each sensor's unique update algorithm, for instance by updating the current hypothesis separately for each sensor. There are also more complex high-level fusion methods that accurately characterize the relationship between sensors. In low-level fusion, sensor information is fused before updating the hypothesis. Some types of information, such as video, are complex to process and benefit from having outside information.

5.1 High-level Fusion with Correlation between Sensors

Sensors can update the hypothesis separately if the sensors' errors, including missed, inaccurate, and fake measurements, are independent of each other. This approach has been used in vehicular multi-object tracking in the research [5.1, 5.2, 5.3] among others. However, it makes sense that the independence assumption is not true for vehicle safety applications. Each sensor faces similar challenges in occlusion, weather, and small or strangely shaped obstacles. Errors from other inaccurate modeling assumptions, such as treating all objects as rectangular and processing all sensors at the same time, also manifest as shared sensor error. This is not to say that the independent errors assumption is the wrong choice for a practical sensor fusion application. The degree of correlation is difficult to quantify, and these correlations complicate the data association step as well. Instead of considering associations between sensor measurements and hypothesized objects, the system must consider three-way associations between sensor A's measurements, sensor B's measurements, and hypothesized objects.

Little published research addresses the importance of sensor correlation in automotive applications. Before implementing correlation into our platform, we simulated several multi-sensor systems to make sure that the addition was worthwhile. The first simulation is a single object moving in two-dimensional space. Two separate sensors report its position five times a second with a standard deviation of 2.5 meters, and 90% correlation in their error. The simulation is roughly based on a car tracking its position using GNSS in an area with poor satellite coverage. Figure 5.1 shows two examples of vehicle paths and measurements. The average positioning error, both considering and ignoring the poor correlation, is shown in Table 5.1. Even in this extreme simulation, the sensor independence assumption biases conclusions by a tenth of a meter or less.

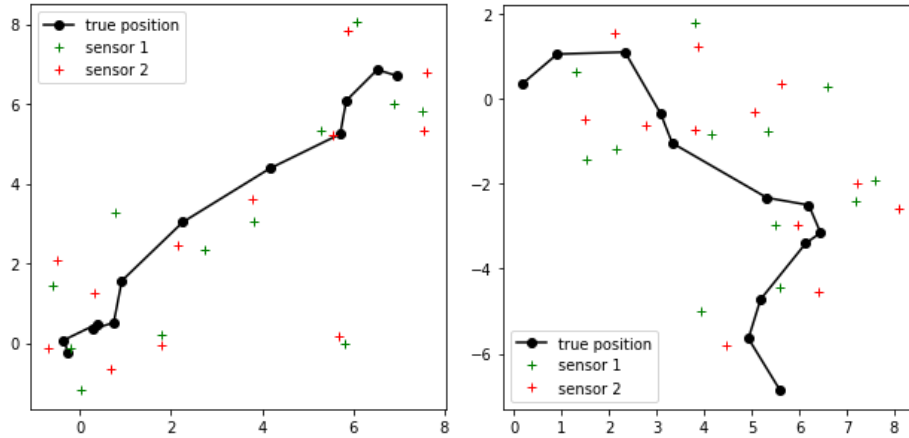


Figure 5.1: Two example simulations with correlated sensor errors

Table 5.1: Performance of tracking in first correlated sensor simulation

Accounting for correlation	RMS error in first second (meters)	RMS error in first twenty seconds
Yes	2.06	1.67
No	2.17	1.70

The first simulation can consider only positioning errors, not failed or false detections of objects. The second simulation tracks multiple objects in a fixed 3D area, with objects entering or leaving freely. Three dimensions were simulated rather than two to mimic previous simulations from [5.4], which were used to show the benefits of multi-sensor tracking. Three sensors detect the various objects with different strengths (each sensor gives accurate measurements in one dimension and weak measurements in the other two). Each object has a two-thirds probability of being detected by any sensor, but the sensor’s probability of detection is correlated such that an object that is detected by one is more likely to be detected by the others. Similarly, sensors are more likely to return false positive measurements in the same area. Figure 5.2 shows example results, and once again considering this correlation in the tracking algorithm does not significantly improve accuracy. Without evidence that sensor correlation has a major impact on tracking, the project prototype does not account for it.

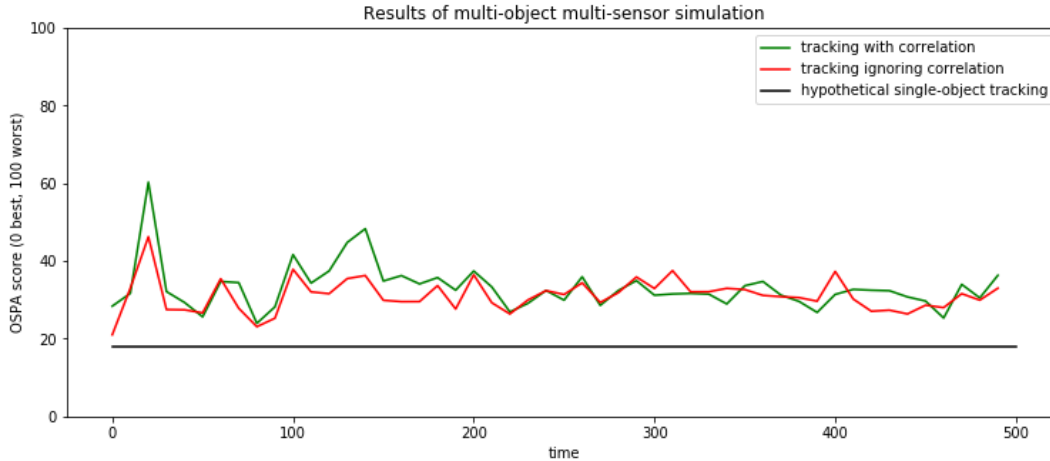


Figure 5.2: Error of trackers on second sensor correlation simulation

5.2 Low-level Fusion

When using high-level fusion, the update step for each sensor will face the same challenges as when only that sensor is used. For instance, measurements from a camera do not have a distance associated with them and thus may have uncertain association with objects. Similarly, objects newly detected by radar or LIDAR may be stationary objects and will be tracked anyways until sufficient verification from the camera that these objects are not significant. In low-level fusion, the information from sensors is combined into new multi-sensor measurements before tracking. Accuracy and detailed information are potentially lost in the creation of these fused measurements, but fused measurements may allow simpler and faster updates than either original sensor individually. We focused on low-level camera/LIDAR fusion, with the goal of creating multi-sensor measurements that correspond predominantly to cars and people and accurately report 2D position.

5.2.1 Low-level Sensor Fusion of Camera and Radar

The fusion between camera and radar is relatively high-level: objects are clustered and detected separately by radar and by camera, but the radar measurements and camera measurements are matched together before being used to update the hypothesis. This process is illustrated in the block diagram shown in Figure 5.3. Each subsystem and their respective inputs and outputs are described in detail in the following sections.

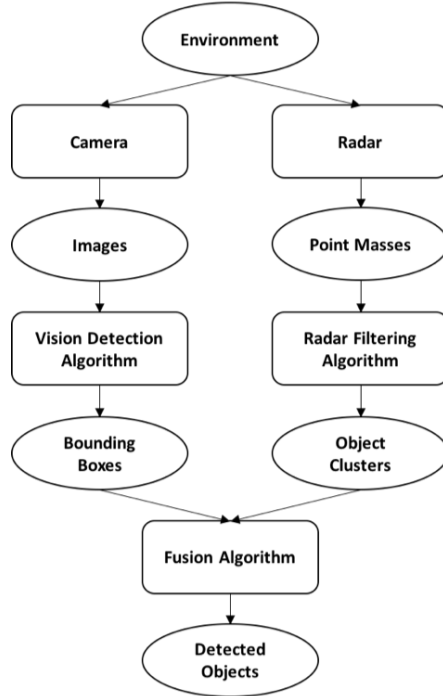


Figure 5.3: Block diagram of the camera-radar sensor fusion system

Sensor Fusion Subsystem

The goal of lower-level fusion is to combine the outputs of the radar and the vision modules to collect more comprehensive information about the vehicle’s surroundings than either could do alone. Camera detections lack any distance information, while radar detections lack any height information. The only information they both report is the lateral angle of objects with respect to the host vehicle’s orientation. We transform radar-reported angles into horizontal pixel coordinates for every object. After conversion, the heuristic equation is used to score each object from radar and vision. The best matching of radar-detected objects and camera-detected objects will maximize the sum of these scores—much like the data association step in tracking finds the hypothesis of highest probability. When the mapping is complete for each timeframe, we put bounding boxes around each object with its classification (from the vision subsystem) and distance (from the radar subsystem) from the vehicle in the output video feed.

$$Heuristic = \frac{\text{overlap of radar and vision object}}{\text{sum of widths of radar and vision object}} + \frac{1}{5\sqrt{\text{distance from radar}}} + 0.1 \cdot \mathbb{1}_{is\ subsumed}$$

Experimental Results

Tests were performed on recorded data from a moving vehicle through the school. 10 specific test time points were chosen. These test times included 30 objects in total, of which 27 were detected by vision and 21 were detected by radar. However, both camera and radar return

false or irrelevant detections, and the fusion method often matched false or irrelevant detections from one sensor with true detections from the other, resulting in a fusion accuracy of 41.38%. A screenshot of the running object detection system is provided in Figure 5.4. The test situations were challenging for radar due to adjacent vehicles and frequent partial occlusion.

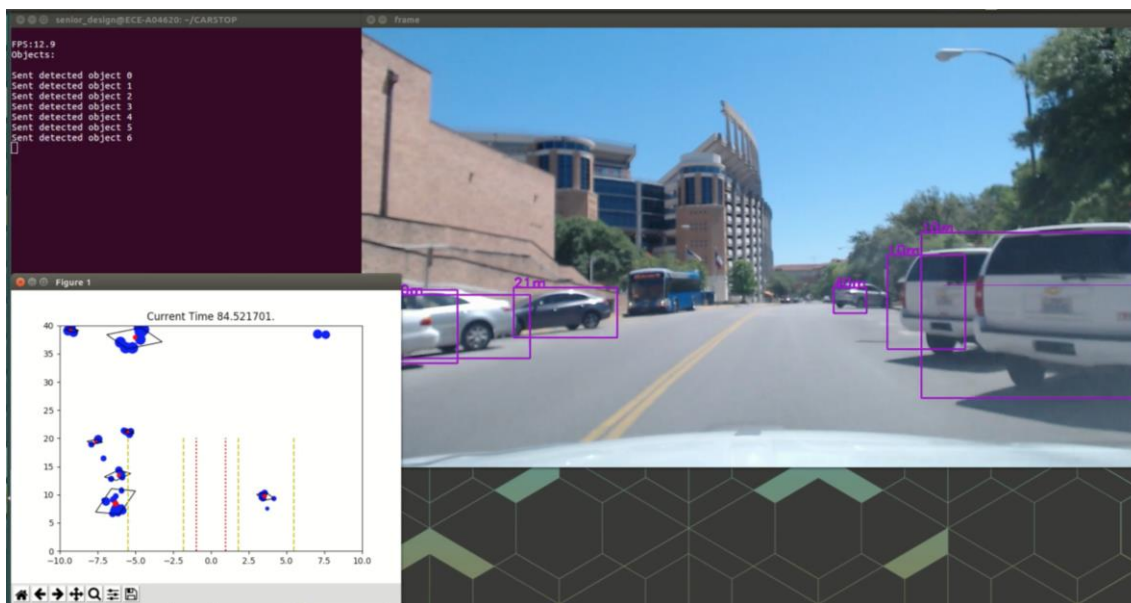


Figure 5.4: Screenshot of the real-time sensor fusion system. Plot at the bottom left shows the detected radar points in front of the vehicle with clustered objects. Video screen includes the bounding box for detected objects (vehicle/pedestrian) with estimated distances from the car using our fusion algorithm.

5.2.2 Shallow Low-Level Camera – LIDAR Fusion

Much like radar, LIDAR and camera measurements can be combined to give the camera measurements accurate distance. LIDAR has height information and a higher resolution than radar, but processing its raw points into object-like measurements is more computationally expensive. Instead, we adopted the fully low-level approach of fusing raw point detections with visually detected objects. Points are projected onto the camera image, both horizontally and vertically this time. Some points within a bounding box are most likely from the detected object, though others may be above or to the side of the object. The closest LIDAR point can however be considered to come from the detected object—provided occlusion is accounted for when the bounding boxes are created. Thus, the minimum distance point within an image bounding box is used to assign a distance to that box. This method is highly computationally efficient, beyond the simple assignment of distance. However, it loses information about the angle and shape of the object that a more complex LIDAR algorithm could potentially derive. Thus, this method is most useful for small objects such as pedestrians. The pedestrian detection application adopts this method, using

YOLO for the visual object detection step. The diagram in Figure 5.5 illustrates the shallow camera-LIDAR fusion process.

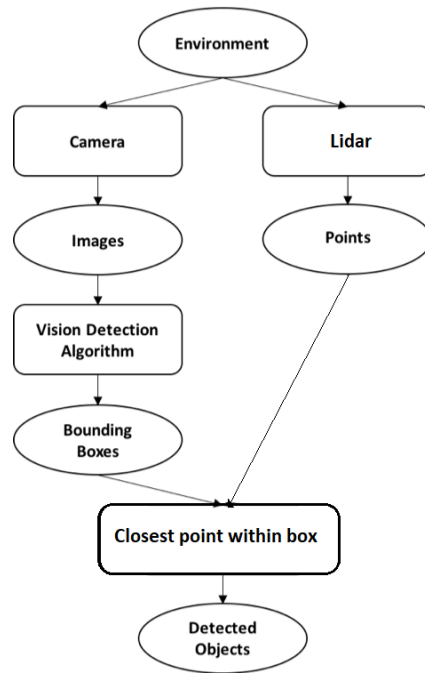


Figure 5.5: Diagram of shallow camera-LIDAR fusion process

5.2.3 Deep Camera-LIDAR Fusion – SSD-Fusion

The shallow fusion method is limited by the standalone performance and speed of a camera-based object detection. In practice, a visual system may fail to detect all relevant objects, for instance in areas with high lighting contrast. In addition to determining the distance of detected objects, radar or LIDAR could be used to improve the visual object detector. In Phase 1 of this project, we developed a method to improve the deep object detection network FRCNN by directly using LIDAR input [5.5]. The field of visual object detection has evolved in the last two years, and the current state-of-the-art networks cannot be modified in the same way. This fusion method was also limited to high-resolution LIDAR. We have since developed a fusion method called SSD-Fusion that operates on newer networks and works with medium-resolution LIDARs (around 16 lasers).

SSD-Fusion directly modifies the structure of the Single Shot Detector network [5.6], as shown in Figure 5.6. LIDAR points are converted into an image-like format where the ‘brightness’ of each pixel corresponds to the distance of the point. The vertical gap between each LIDAR laser causes vertical bands of pixels with missing values, which are handled in two ways. Initial max pooling sets missing values to the nearest value. The convolutional operation linking two layers of the deep network is set to be longer vertically than horizontally, so that information from the

nearest lasers can be utilized for every pixel. Along with this new input, a distance output is added to the network. The form of this output is equivalent to that of [5.7], though they did not use LIDAR as an input.

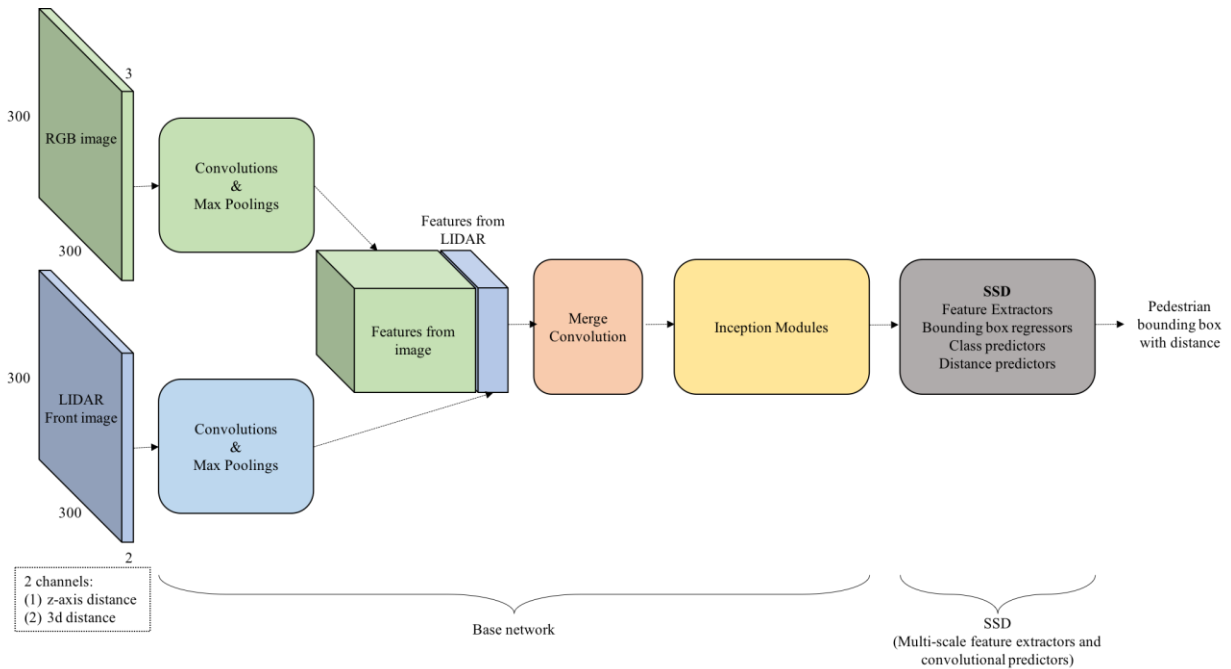


Figure 5.6: Framework of SSD-Fusion

Because the network is modified, SSD-Fusion must be trained with new data that includes video, recorded LIDAR readings, and accurate ‘true’ pedestrian positions. All of these were collected as part of the pedestrian detection tests discussed in Chapter 14.

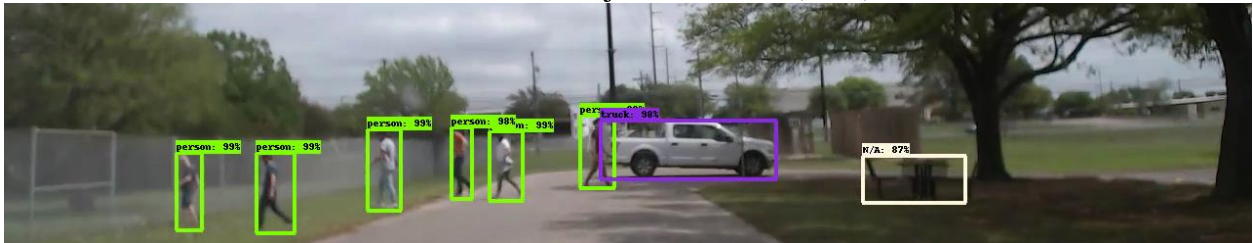
The accuracy of different fusion methods is determined with test pedestrian data in Section 14.3. Figure 5.7 and Figure 5.8 show a single time step of this data to demonstrate the discussed sensor measurements and fusion techniques. Figure 5.7 shows the camera’s view and results from two visual object detectors. Faster-RCNN with NAS is one of most powerful object detector networks [5.8], but is too slow for real-time applications. SSD is the faster detector used as the basis for SSD-FUSION. SSD only captures three of the pedestrians in this image, and the boxes that bound them are less exact than for FRCNN. Figure 5.8 shows measurements in a bird’s eye view grid where the host vehicle is at the bottom facing up, as denoted by the green icon. This grid covers 26.7 meters in the forward direction, and 20 meters to either side of the vehicle. The LIDAR measurements are shown as preprocessed line segments, as discussed in Section 4.4.3. All pedestrians except the furthest to the side have been captured by the LIDAR, but many other objects—such as the trees, benches, and fence—have been captured as well. The other measurements are shown as rectangles. The pedestrians detected by Faster-RCNN have their distance imputed by the method from Section 5.2.1. This overestimates the distance of the pedestrian directly ahead of the host vehicle, to the point where that measurement is off the grid. The shallow fusion with the slow but strong object detector detects all pedestrians and gives an

accurate measurement for all but one, who is far to the side of the vehicle. SSD-FUSION captures one more pedestrian that shallow fusion with SSD fails to detect. But it doesn't detect all six pedestrians and seems to estimate distance with several meters error. It is worth noting that on a computer with a powerful graphics processor unit (GPU), SSD and SSD-Fusion can perform on 25 images (and LIDAR rotations) per second, while Faster-RCNN can handle less than 10.

Scenario



Faster-RCNN Object Detection (slow)



SSD Object Detection (fast)



Figure 5.7: Measurements from different sensors/fusions at a single timestep, camera view

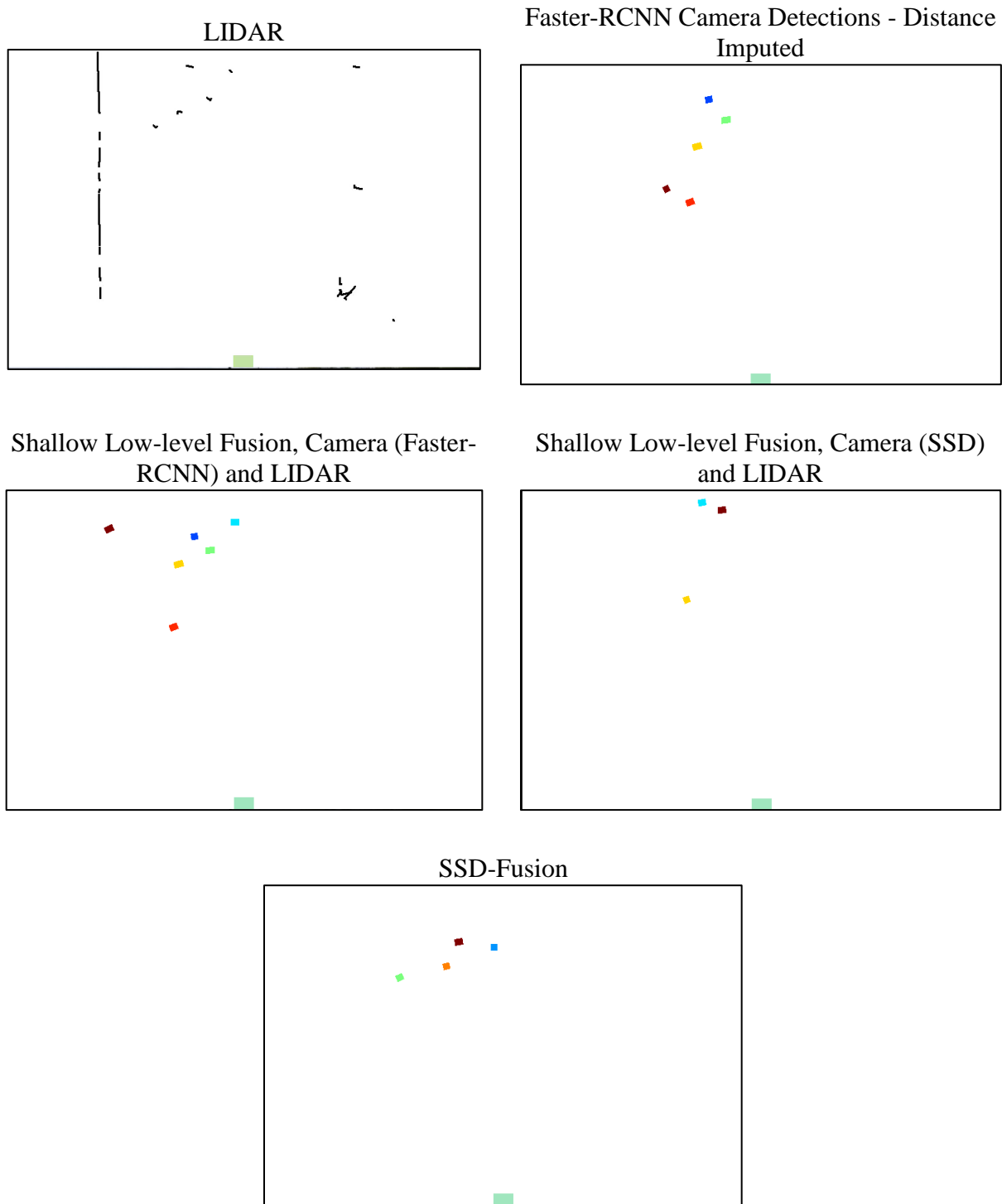


Figure 5.8: Measurements from different sensors/fusions at a single timestep, top view

5.3 Camera-LIDAR Calibration

Figure 5.9 shows an example of incorrect calibration between a camera and a LIDAR. The colored dots are points returned by LIDAR, with warmer colors corresponding to higher distances. The red car in front of the camera is also detected by the LIDAR, but the projected points are positioned to the upper left of the vehicle. This situation leads to a confusion in the overall CA/CW system because locations of surrounding objects cannot be accurately obtained.



Figure 5.9: Example of bad calibration between a frontal camera and a rotational 64-layer LIDAR (KITTI dataset). Colored dots represent different distances measured by the LIDAR projected to the image’s coordinate system.

5.3.1 Manual Sensor Calibration

One of the most traditional sensor calibration methods is to surround the sensors with checkerboards as in Figure 5.10. Checkerboards are easy to detect from both camera and LIDAR data and thus can be used to match the two. For example, [5.9] provides a software with GUI so that a user can select a planar region from LIDAR’s detection.



Figure 5.10: The room used to calibrate sensors to create the KITTI dataset [5.10]

We created a similar graphic system that displays camera video with LIDAR points overlaid and allows the user to adjust the sensors’ positions and angles until a qualitatively good match is achieved (Figure 5.11). Checkerboards may be used, but an ordinary image, for instance

of several people standing in front of the vehicle, is sufficient to roughly calibrate with this tool. For best results, multiple pictures/measurements with different views are used.

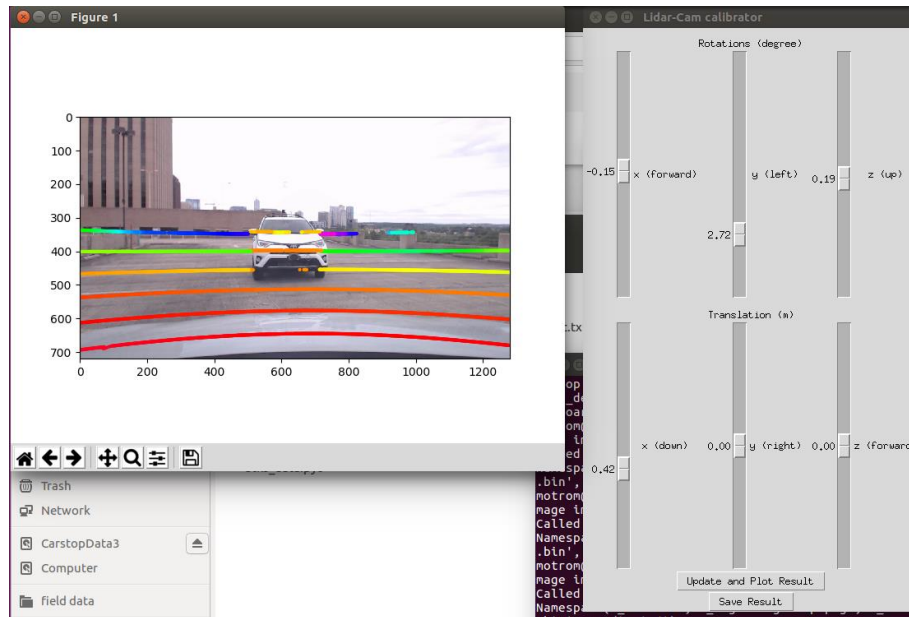


Figure 5.11: Screenshot of the project calibration tool

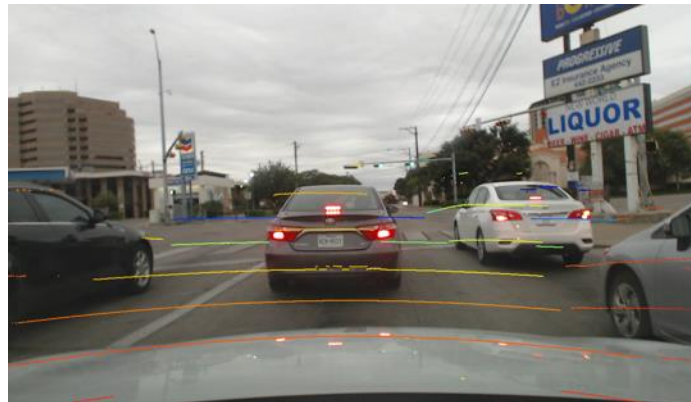
The downsides of manual sensor calibration are listed as follows:

- Takes time for physical preparation.
- The process of finding the best rotation and position is difficult and time-consuming.
- Cannot correct error during use. For instance, if external forces (such as a camera being bumped) cause a sensor to move slightly, the change would not be detected or fixed until another the checkerboard calibration is applied again.

The first and third downsides were not fully appreciated when collecting some of the intersection data. Figure 5.12 shows the projection of LIDAR onto the image plane in our collected intersection dataset using hand-tuned calibration parameters based on recorded measurements and trial and error. The dots are points detected by LIDAR, with warmer colors corresponding to closer points. From the figure, objects like the small tree in (a) and vehicles in (b) can be inferred from the LIDAR measurements, but the calibration still retains some error. Later test and collection runs with more careful calibrations were performed, as documented in Section 4.4.3.



(a)



(b)

Figure 5.12: Example of calibration results from the collected data showing the limit of hand-tuned approach. Logitech Pro Webcam C920 is used for the frontal camera and Quanergy M8 with eight layers is equipped as a rotational LIDAR sensor.

5.3.2 Automatic Sensor Calibration

To overcome the drawbacks of target-based extrinsic calibration methods, online algorithms and calibration approaches without fixed targets have been developed. In these methods, raw measurements from both camera and laser sensors are used in the calibration process by aligning certain attributes [5.11, 5.12]. This can be applied in any situation during vehicle operation, meaning it can also be applied at regular intervals to detect accidental sensor movement.

Recently, a deep-learning-based algorithm, RegNet, has been proposed that formulates the problem as a regression task for predicting calibration parameters [5.13]. Fusion of camera and LIDAR data is achieved by combining two neural networks in intermediate layers. [5.13] generated samples to train the network by randomly *de-calibrating* correctly calibrated data from the KITTI dataset. Figure 5.13 illustrates the overall architecture and flow of the RegNet algorithm.

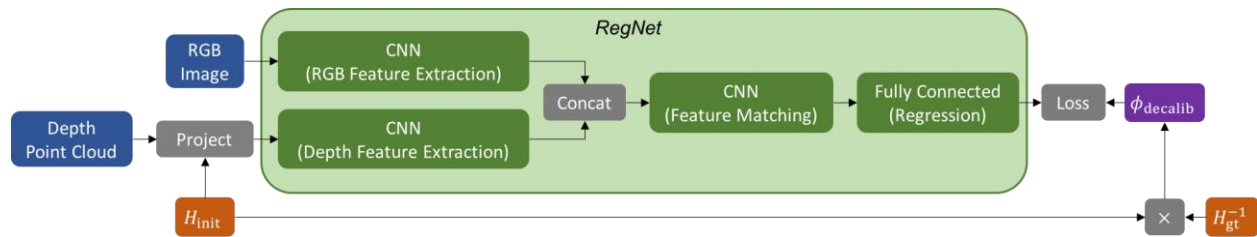


Figure 5.13: Overall architecture and flow of the RegNet where inputs are images, depth point clouds, and initial calibration matrix H_{init} [5.13].

We attempted to apply RegNet’s idea, which is trained on 64-layer LIDAR data, on our 8-layer LIDAR data with lower resolution. Code for the RegNet is not publicly available, so we implemented a TensorFlow module based on the RegNet research with some slight modifications to fit our sensors. Our implementation performed well on the 64-laser data, as shown in Figure 5.14, but poorly on 8-laser data. The manual method was considered more reliable and used for the demonstrations and tests.



(a)



(b)



(c)

Figure 5.14: Example of calibration results using RegNet on KITTI dataset: (a) randomly de-calibrated sensors (b) true calibration (c) calibration using the output of RegNet.

Chapter 6 Application to CA/CW

6.1 Prediction

Predictive models for vehicles on urban roads was a focus of the first phase of this project. Physics-based prediction models that rely solely on values such as speed and acceleration were found to be less reliable than models that considered the road shape and context. This comparison will not hold for all applications, however. Wrong-lane overtaking occurs predominantly on single-lane rural roads, and vehicles merging onto a highway are only in danger from vehicles on the rightmost lane of the highway. These applications also consider potential collisions more than several seconds in the future, where any model will be inaccurate. Simple and pessimistic models of approaching vehicles are most appropriate for CW/CA tools for these applications.

Pedestrians should also not be modeled based on road rules, as those rules are often broken. At least one high-profile autonomous vehicle crash involved a pedestrian crossing a road with no crosswalk [7.1]. Pedestrians are also difficult to predict because they can change direction much more freely than cars. We currently predict that pedestrians continue at their current speed, which is usually sufficient as pedestrians move much more slowly than cars. This fits with our current object representation from Section 3.2.2, which maintains object speed during tracking. We have also begun to study behavioral prediction that takes a more long-term view of the pedestrian's motion.

6.1.1 Long-term Prediction based on Behavior

Without reliable situation-based rules with which to predict pedestrian motion, the alternative is to find patterns in collected data. Motion patterns for vehicles at an intersection were discovered in Phase 1 of this project, by clustering different collected paths. Clustering works well at an intersection because any one crossing vehicle takes exactly one path, and therefore should match one cluster. Pedestrians may instead follow one motion pattern for an uncertain amount of time then switch to another. A Bayesian model was designed to extract motion patterns from pedestrian data. Pedestrian motion at each timestep is modeled as a weighted sum of motion patterns, where the weight for each pattern varies across people and time. The model also specifies which motion patterns are likely to switch to other motion patterns and when. Once the motion patterns and their switching behavior are known, the pedestrians can be tracked with a method similar to interacting multiple models. Actually determining the motion patterns requires time-intensive random sampling on a dataset of pedestrians.

The motion pattern prediction method was tested on pedestrian data from the 2015 Multi-Object Tracking Challenge (MOT Challenge) [7.2]. This data consists of natural (unscripted) video of a busy pedestrian walkway and annotated ground positions for each pedestrian at each time. Three significant motion patterns were derived, meaning only three were assigned high weights for any pedestrian. Example trajectories for a single pattern can be gathered by tracking pedestrians while assuming they follow this pattern. Figure 6.1 shows these example trajectories, which can

roughly be classified as large turns or loops, straight paths with sharp turns, and walking straight. The patterns are used for prediction in Figure 6.2. Six pedestrians are tracked using this method, and their future position 0.1 seconds ahead is predicted (red line) and compared to the truth (blue line). The x and y dimensions are split into separate plots, so that the true and predicted paths can be compared across time. Prediction with motion patterns matches extremely well with the ground truth.

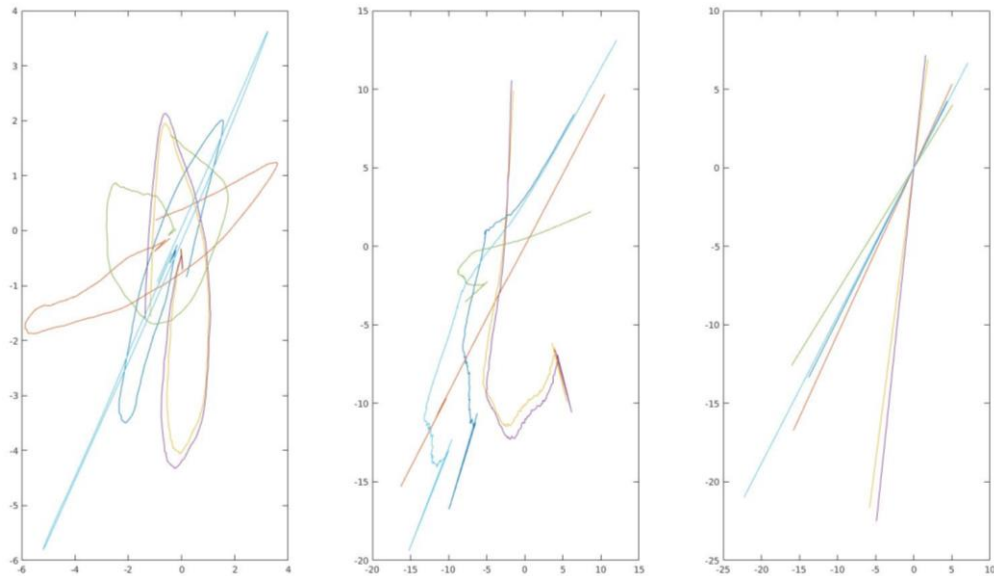


Figure 6.1: Example trajectories from each extracted motion pattern

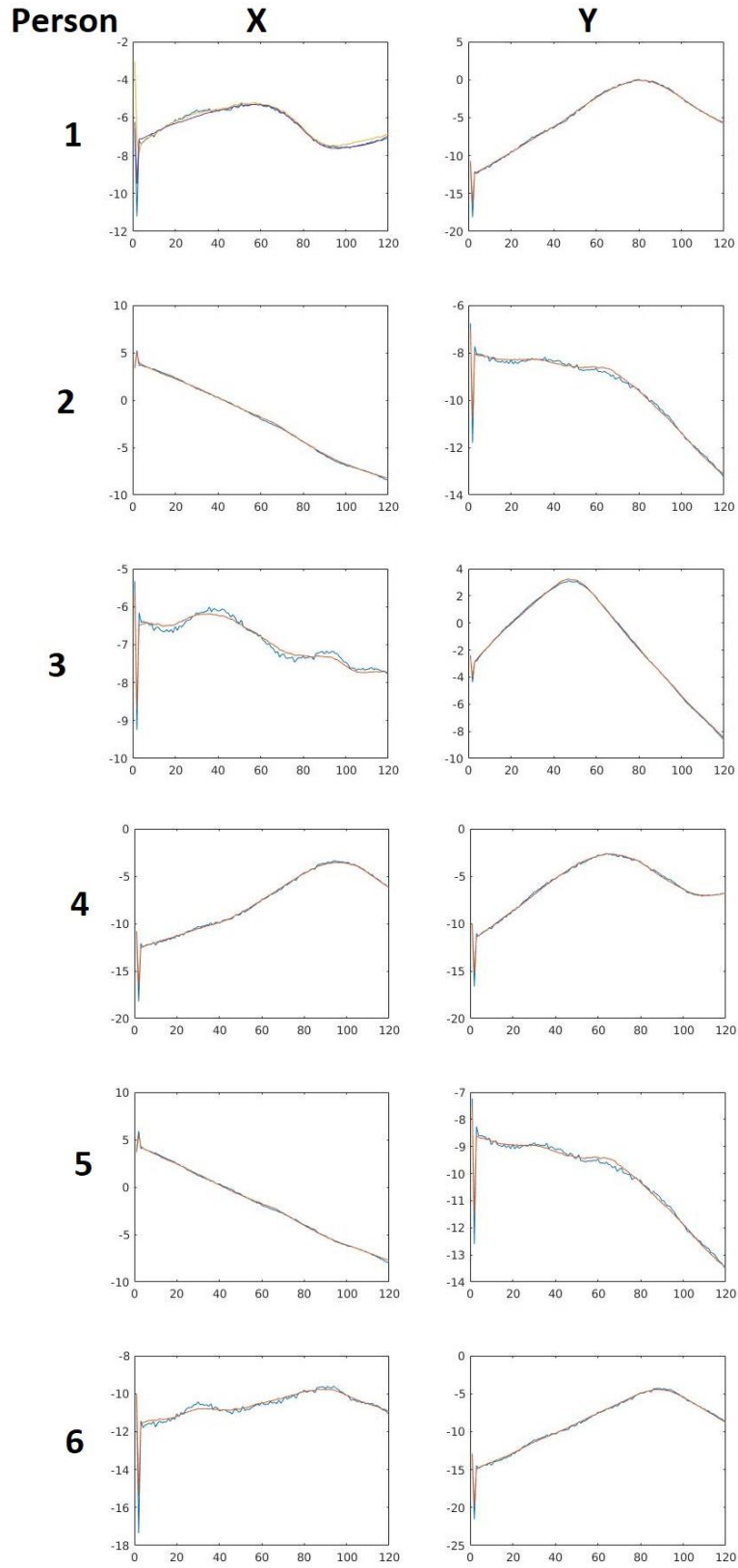


Figure 6.2: Predictions on pedestrians using the extracted motion patterns

6.2 Collision Detection

Section 2.3 of this project’s Phase 1 Report [7.3] discussed some of the challenges of advanced collision detection, given the uncertainty of predicting other vehicles’ future movements. Those results were expanded into a full study of the speed and accuracy of various approximations to collision detection, which was presented at the IEEE 2017 Intelligent Vehicles Symposium [7.4].

Figure 6.3 shows the results of this study. The quality of any method is determined by assigning a relative cost to False Negative errors (undetected collisions) as compared to False Positive errors (warnings in safe situations). The average overall cost added solely by inaccuracies in the collision detection stage—not including errors in the vehicle tracking and detection—forms the metric Expected Additional Cost. Runtime of each method is also calculated, as this task only comes into play after detection and tracking and therefore the system must operate very quickly to make real-time decisions. The principal conclusion of this paper was that Monte Carlo approximations, or the aggregate of several hundred random predictions forward in time, are the most reliable method to determine the probability of a collision in the near future. This method is easily combined with the particle-based tracking methods we have used.

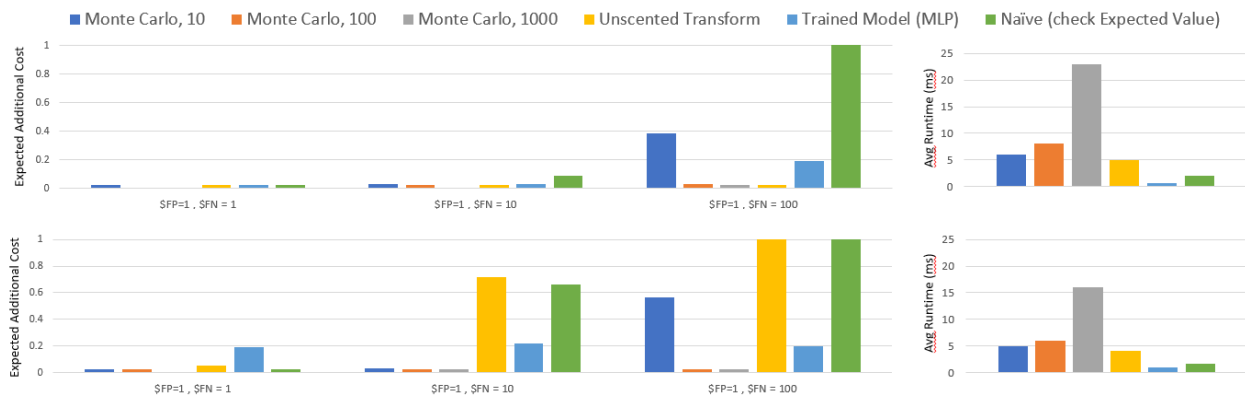


Figure 6.3: Results of the vehicular collision detection study, with relative False Negative costs of 1, 10, and 100

Our prototype systems also issue collision warnings when there is a nontrivial probability of a collision—a far simpler task than actual collision avoidance, which is left to the human. Collision avoidance requires both prediction of a collision and a choice among the options available to the vehicle. The following section continues our study of a method for connected vehicles to cooperate and achieve their goals while avoiding collisions.

Chapter 7 Distributed Decision-Making for Cooperative Driving

7.1 Previous Work and Goals

Currently, self-driving vehicles follow human road rules and conventions well enough to ensure safety. Once most vehicles become autonomous and connected, it will be possible for them to follow more subtle human driving conventions or to ignore them entirely. The question is how each vehicle might make driving decisions that fulfill their own goal but also allow the surrounding vehicles to fulfill theirs. We used the Collective Intelligence (COIN) [7.1] framework to design an algorithm allowing fully automated and connected cars to each navigate to their destination through an intersection. Managing a multi-agent environment is often very labor-intensive, as it requires tuning the interaction between the agents to make sure that they are cooperative. These techniques normally result in non-robust systems with limited applicability.

The COIN framework is designed to automatically learn cooperative strategies. It requires three specifications: states, actions, and rewards. A state is the information known to a vehicle at a single point in time. We assume autonomy, fully available road maps, and full V2V communication. Each vehicle's state includes its position and destination as well as the position of all nearby vehicles. Each action is a motion a single vehicle can make in a short time, namely speed changes, lane changes, and decision of where to turn. The reward is a handmade score for a vehicle's situation. Quick arrival at the destination has a high reward, for instance, while collision has a very low reward. COIN learns a function that takes the state and outputs the action of a single vehicle, such that these actions optimize the total reward of all vehicles.

The previous phase used COIN and simulations to train vehicles to cross a four-way, four-lane intersection [7.2]. In order of significance, the reward function penalized agents for collisions, exits in the wrong direction, and delay in crossing an intersection. Improvements to the learning technique are covered in Section 7.2. The robustness and generalizability of COIN are studied in Sections 7.3 and 7.4, and its performance on real driving examples in 7.5. Intersections and rural roads, both focuses of this project, are both used. COIN is not applicable to all focuses, such as pedestrian safety, because it only applies to agents in a communication network that can be expected to follow a set protocol.

7.2 Model Improvements

7.2.1 Function Approximation

The previous work used a discrete but large set of states and actions. It is infeasible to simulate every possible state and action; we utilized function approximation to estimate the best action for states that were missed in training. Function approximation methods include ranking models such as Monotonic Retargeting [7.3], feature selection techniques such as Orthogonal

Matching Pursuit (OMP) [7.4], and Orthogonal Least-Squares (OLS) [7.5]. Results of the OMP-based function approximator are reported in [7.2] and provided in Table 7.1, and results related to the generalized OLS function approximator are reported in Table 7.2. The improvement in generalized OLS is achieved by grouping the feature section at every step rather than one feature at a time.

The max speed of each car at any point could vary between 0 to 60 km/h (37.6 mph). Note that the average speed in a two-way maneuver is lower than that of four-way maneuvers. In the two-way maneuvers, all cars in a certain street were set to turn left, while in four-way maneuvers all car destinations were randomly selected. It makes sense that straight and right-turn maneuvers are still faster on average to perform, as they have no conflict with cars on the opposite road.

Table 7.1: Results of COIN simulations OMP

Maneuver	Average number of cars per episode	Collision (%)	Wrong path (%)	Average time (seconds)	Number of random episodes for testing
2-ways	45	0.65	2.5	14.7	20
4-ways	85	3.2	12.5	13.9	12

Table 7.2: Results of COIN simulations OLS

Maneuver	Average number of cars per episode	Collision (%)	Wrong path (%)	Average time (seconds)	Number of random episodes for testing
2-ways	40	0.73	2.1	12.3	10
4-ways	80	4.5	9.8	13.5	10

7.2.2 Deep Reinforcement Learning

Continuous states and actions give the agent a more precise knowledge of the environment. As a simple example, if the speed of cars is discretized to 10mph, the difference between a car’s speed and the speed of the car directly ahead of it is either given as -10, 0, +10, etc. Continuous states can hypothetically utilize smaller differences in speed to smoothly approach or follow an ahead vehicle. However, these states must be fed into a model in order to apply reinforcement learning.

Deep reinforcement learning—reinforcement learning with deep networks as the modeling tool—was first proposed in 2013 and has shown superior performance on many reinforcement learning problems [7.6, 7.7]. However, our multi-agent framework has some challenges that are not usually addressed with deep networks, namely a variable number of agents and partial observability—each agent uses information only on vehicles relevant to it, rather than using the entire state space of all vehicles. We implemented several deep architectures, with the first shown in Figure 7.1. The initial architecture has been inspired by Tampuu et al. [7.7] and Mnih et al. [7.6] with a few differences:

A fully connected network is used rather than a convolutional neural network (CNN) as each agent is communicating to its neighboring agents and each agent has partial observability of environment. The objective function is modified to accommodate the difference reward for each agent. The number of vehicles is variable at every time step, since cars enter and exit the simulated intersection. However, deep networks are trained on batches of fixed size. We used a fixed-size batch and collected enough samples, from different cars and different times, to fill the batch.

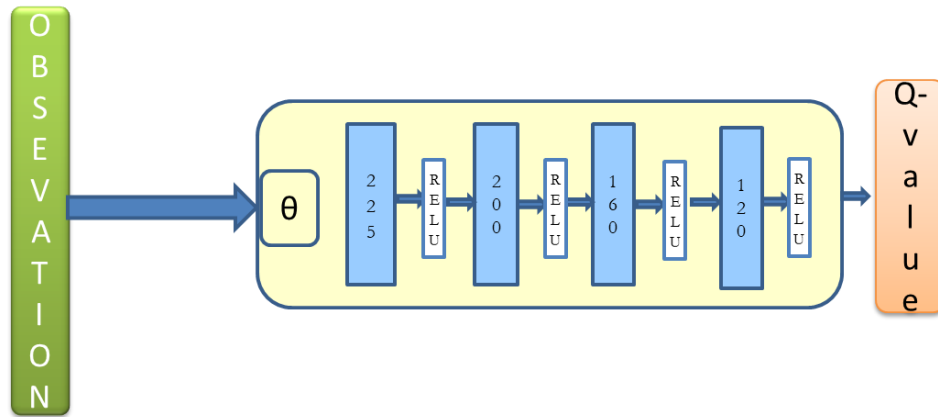


Figure 7.1: The first architecture of the deep reinforcement learning model

Another potential avenue for improvement is to give each agent more information at each time. This can be done by adding Recurrent Neural Network (RNN) or Long Short-Term Memory (LSTM) layers at the last layer of the fully connected network [7.8, 7.9]. By adding LSTM as the last layer of our model, features at every time step are used to provide each agent a sort of ‘memory’. This memory may allow agents to avoid the deadlock scenarios [7.2] that occur with simultaneous four-way incoming traffic at intersections.

Table 7.3 shows the results of both methods. The LSTM method performs better, but neither implementation of deep reinforcement learning outperforms the simpler models of Section 7.2.1.

Table 7.3: Results of COIN with deep Q-Learning on four-way intersection

LSTM	Average number of cars per episode	Collision (%)	Wrong path (%)	Average time (seconds)	Average speed (km/h)	Number of random episodes for testing
No	85	14.6%	2.1	14.3	50.3	20
Yes	85	9.7%	3.2	14.9	48.3	20

7.3 Robustness to Sensor Error

Because of the noisy nature of sensor readings, it is possible that vehicles will have inaccurate knowledge of other vehicle’s positions even with fully communicative vehicles. In order to examine the performance of COIN under noise, we add noise to each vehicle’s position

at each timestep independently. This noise is Gaussian with 1 meter standard deviation. Three training methods are tested for performance under noise.

1. The trained model of Section 7.2.
2. The model of Section 7.2 retrained in the noisy environment.
3. A retrained model using methods that are meant to handle noisy information.

The first model is effectively tested in situations it has not been trained for, so poorer performance is expected. Retraining the model using Q-learning [7.10, 7.11] should compensate for noise. However, normal Q-learning is known to be at risk in noisy environments, as it may learn wrong concepts (due to noisy input) early on and fail to converge. Method 3 applies G-learning [7.12], a recent modification to the Q-learning algorithm that regularizes the value estimates by penalizing deterministic policies in the beginning of the learning process.

Table 7.4 gives results on the four-way maneuvers. G-Learning results in a safer if more conservative model, and traffic moves relatively slower as vehicles need to give bigger headspace and side space on average for each vehicle to move safely.

Table 7.4: Results of COIN retrained model under described scenarios

Method	Average number of cars per episode	Collision (%)	Wrong path (%)	Average time (seconds)	Number of random episodes for testing
1	65	11.3	8.5	14.3	10
2	65	8.2	10.1	14.6	10
3	65	6.5	10.9	15.8	10

7.4 Cooperative Highway Driving

An important question for a cooperative driving algorithm is whether it can be trained to work under any condition (one intersection vs other intersections vs highway vs rural road) or whether a new framework will be needed for driving under each of these conditions. Our chosen states and actions for each vehicle are flexible in meaning and easily expandable to other scenarios. We have run the model, as trained in Section 7.2.1, on a simulated two-lane highway and one-lane rural road scenarios. The traffic simulator used for COIN does not allow wrong-lane overtaking, as it only allows each vehicle to use lanes aligned with their direction of movement. However, overtaking should be unnecessary in cooperative autonomous systems as all vehicles should stay near the maximum allowed speed (here maximum possible speed assumed to be 60 km/h), slowing down only in unusual situations. Table 7.5 contains results for each road.

Table 7.5: Results of COIN simulations in different conditions

Condition	Average number of cars per episode	Collision (%)	Wrong path (%)	Average time (seconds)	Average Speed (km/h)	Number of random episodes for testing
Highway	65	0.0	0.0	14.3	50.3	10
Rural road	65	0.0	0.0	13.0	55.4	10

The lower average speed of the highway scenario, compared to the rural road, is due to vehicles heavily favoring the right lane as shown in the snapshot of Figure 7.2. This is most likely caused by poor generalization of the current training. In the intersection scenario, the left lane is used for left turns only and the right lane was used for straight or right-turn maneuvers. A single state variable, “correct lane” can be used to determine whether a lane change is desired, but other variables such as “lane number” and “adjacent lane free” have overlapping meanings and are probably replacing “correct lane” in the decision model. This simple example shows that even with a state space that is generalizable to all roads, a distributed decision-making model must be trained on a variety of simulations to generalize correctly.

**Figure 7.2: Snapshot of simulated highway scenario**

7.5 Applying COIN to Real Scenarios

COIN’s decision-making capability can be verified by comparing natural human driving behavior. If the two are compatible, distributed decision-making systems could be used pre-autonomy (but with V2V) by recommending actions to the driver that are friendly to other cars, potentially resulting in a global speed improvement.

Given that current driving is at least partially cooperative, it is tempting to learn a cooperative strategy from real traffic data. However, real driving behavior cannot be directly used to train COIN for several reasons. First, the state of a vehicle can only be fully calculated if the position of all relevant vehicles is known well. This may include vehicles that are not visible to human drivers, as COIN is expected to function in a fully V2V-equipped environment. Secondly, COIN learns by slowly reaching a consensus approach to driving among all cars. Human drivers have a consensus of sorts, but it is not always present and impossible to modify. Finally, reinforcement learning requires negative examples such as collisions, and these are rare events in reality.

We ultimately decided that the only reasonable way to improve COIN with human data is to evaluate COIN’s decisions in real situations, then revisit the training simulations if a poor choice of some kind is detected. We located a few scenarios from the intersection tests of Chapter 12 in which only our test cars are significant from each other’s decision-making perspective. We have extracted all parties’ trajectories based on their GPS recordings. Future positions of each vehicle,

as returned by GPS, allow us to see what would have happened had one vehicle changed their action (assuming the other does not change theirs in response). This is how the safety of COIN’s recommended actions is assessed.

Five scenarios are described and analyzed below (Table 7.6). Videos of these maneuvers are available with the full collected data, shared with TxDOT part of the project deliverables. Recommended actions are given by the OMP trained model from Section 7.2.1.

Table 7.6: Time of selected scenarios

Scenario	Date	Vehicle	Location	Time in Video
1	May 21, 2017	Honda Accord	Congress and Riverside	10:26
2	Sept. 24, 2017	Honda Accord	Farm to Market Rd 973	3:27
3	Sept. 24, 2017	Honda Accord	Farm to Market Rd 973	3:36
4	May 21, 2017	Honda Accord	Congress and Riverside	8:21, 8:26
5	May 21, 2017	Honda Accord	Congress and Riverside	13:12

Scenario 1

The Accord intends to turn left and the RAV4 to go straight. The RAV4 has stopped at the intersection but the Accord has not. COIN recommends the following action for the Accord: keep lane, increase speed by 10 km/h, and turn left at the intersection. The actual driver of the Accord hesitates for longer (probably to check for other crossing vehicles), but COIN’s decision is safe, as the trajectories of the RAV4 movement show no collision will happen as a result of that action. Figure 7.3 shows the Accord’s view at the moment the action is decided.



Figure 7.3: Scenario 1 screenshot

Scenario 2

The Ford F-150 is followed by the Accord, as depicted in Figure 7.4. The F-150 stops at a stop sign before turning onto the rural road. The Accord is moving towards the F-150, although the F-150 has completely stopped. COIN's decision is to decrease speed.



Figure 7.4: Scenario 2 screenshot

Scenario 3

This scenario, as seen in Figure 7.5, occurs a few seconds after scenario 2. The F-150 starts to move again while the Accord stays stationary. The COIN recommendation for the Accord is to keep its lane, speed up by 10 km/h, and turn right. All COIN vehicles assume that all other vehicles on the road follow the same collective intelligence model. This is not a dangerous decision, as the Accord can slow down again during the next turn if the truck stops again. However, the scenario is a good reminder of the strict V2V requirement of COIN, since the presence of an eighteen-wheeler did not affect the Accord's decision.



Figure 7.5: Scenario 3 screenshot

Scenario 4

The driver's decision in this scenario again disagrees with COIN's recommendation. The RAV4 is traveling at 28 km/h while the Accord is at 3 km/h and can be considered almost stationary at the intersection. The COIN decision for the RAV4 was to increase speed by 10 km/h, while the decision for the Accord was to keep its current speed. COIN recommends that cars cross in the reverse order, which would have been equally valid and would in fact route cars to their destinations faster. These videos were taken in data collection scenarios (Chapter 12) where right-of-way was designated to the Accord. Five seconds from this point, COIN adapts and instead decides that Accord crosses first. A screenshot of the scenario can be seen in Figure 7.6



Figure 7.6: Scenario 4 screenshot

Scenario 5

The Accord is passing straight through the intersection while the RAV4 turns right onto the same lane. COIN recommends that both vehicles increase speed, which is what both drivers choose to do. This scenario is depicted in Figure 7.7.



Figure 7.7: Scenario 5 screenshot

Chapter 8 DSRC-aided mmWave Vehicular Communication

Multiple-input multiple-output (MIMO) communication at mmWave frequencies is one of the key technologies for 5G cellular networks [8.1]. It enables high data rate and low latency communications, required in several applications associated to the industry verticals that have driven the development of the recently released 5G New Radio standard [8.2, 8.3]. The automotive vertical is possibly the most novel and interesting one, with automakers, cellular operators and equipment vendors working together to make vehicular communications (V2X) operating at mmWave frequencies a reality. As illustrated in Figure 8.1, mmWave V2X is the only feasible approach to provide a solution to many 5G V2X use cases: intelligent navigation, cloud assisted driving or cooperative collision avoidance to name a few.



Figure 8.1: Illustration of the cellular infrastructure supporting several 5G V2X use cases enabled by mmWave communication

The main difference between a conventional MIMO communication system operating at lower frequencies and a mmWave MIMO system is that the communication has to be directional (the antennas cannot radiate the power in all directions, but have to steer the energy towards the receiver), which requires the usage of antenna arrays with a large number of elements. Given the small wavelength, these large antenna arrays can be integrated, however, in a very small area. The greater challenge is how to configure/reconfigure the antenna arrays quickly, so the antenna beam steering works well even in high mobility scenarios. Steering the beam pattern in the right direction requires knowledge of the propagation environment, which changes very quickly in vehicular settings. To overcome this difficulty, we have proposed the idea of using out-of-band information

obtained from other communication systems operating in parallel, or from other sensors, to provide prior information that can speed up the process of acquiring this knowledge about the environment.

Beam alignment is the process of finding this good pointing direction to focus the signal energy. In a vehicular setting, the high mobility requires frequent beam realignment. In this chapter, we describe our work on using a standard low data rate vehicular communication technology, DSRC (dedicated short range communications), operating in parallel at the vehicles and the road infrastructure, to provide this prior knowledge about the environment and speed up the beam alignment process. DSRC is a mature standardized communication protocol for vehicular communications. It has been tested over a decade and is ready for deployment [8.4]. It enables the exchange of messages between DSRC-equipped entities, e.g., vehicles and roadside units (RSU). One of the most important message types in DSRC is the basic safety message (BSM). A BSM contains two parts: Part I includes critical state information and is mandatory, and Part II has customizable contents [8.4]. Part I contains information that is useful for mmWave beam alignment including position information (longitude, latitude, and elevation), positional accuracy, dynamics (heading, speed, acceleration), and the vehicle size. DSRC can also be used for system control, such as requesting mmWave beam alignment to establish a link.

The research team initially considered an approach based on obtaining position information from cameras located at the RSU. Given the superior performance of the machine learning approach we developed recently, we chose to include in this final report only the DSRC-based approach.

8.1 A Machine Learning Approach to DSRC-aided Beam Alignment

8.1.1 System Model and Problem Description

We assume a V2I urban street canyon environment. Our approach uses machine-learning methods that rely on the training data. Therefore, the training data must capture the important features of the information we want to extract, in our case the propagation environment that defines the communication channel. To generate a sufficient number of different propagation environment/communication channels, we use a commercial ray-tracing simulator, Wireless InSite [8.5]. A snapshot of the simulated environment is shown in Figure 8.2. The street has two lanes. We simulated two types of vehicles: cars and trucks. The trucks have larger dimensions than those of cars. The target communicating vehicle is a car in the left lane. The target car's antenna is at 1.5 m, and the RSU's antenna is at 7 m. Depending on the location of the trucks on the right lane, the LOS path could be blocked.

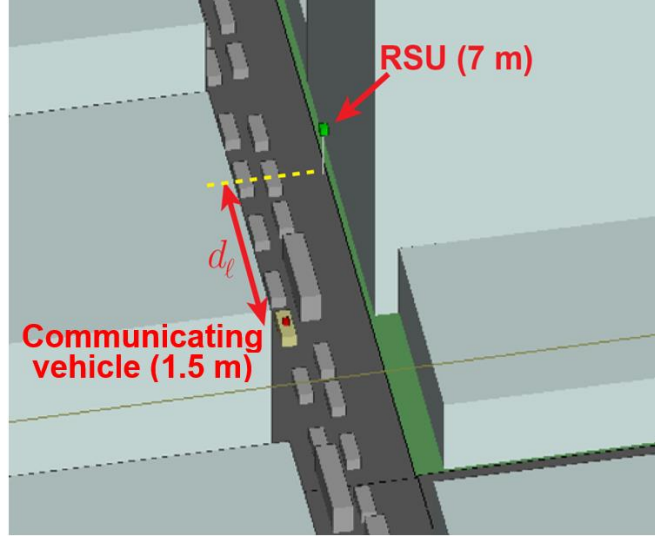


Figure 8.2: Simulated urban street canyon environment. The target vehicle is on the left lane at a longitudinal distance d_ℓ from the RSU. The target vehicle has a roof-mounted antenna at 1.5 m high, and the RSU has its antenna at 7 m high.

The ray-tracing output is used to parametrize a wideband geometric channel model for mmWave frequencies which is a widely accepted in the previous literature [8.6]. The obtained channels can be used to compute the received power and assess the strength of the mmWave link for a given pointing direction. We use uniform planar arrays (UPAs) in our simulations. The beam can be steered by controlling the phases of the antenna elements of the arrays. We quantized the pointing directions in the field of view of the array by the beamwidth of the beams. This provides us with a set of predefined beam directions, which is commonly called a beam codebook. We assume the array has a grounded backplane so that there is no radiation toward the back of the array ($-z$ direction in Figure 8.3). The beam patterns of the entire codebook for an 8×8 UPA is shown in Figure 8.3 (Right).

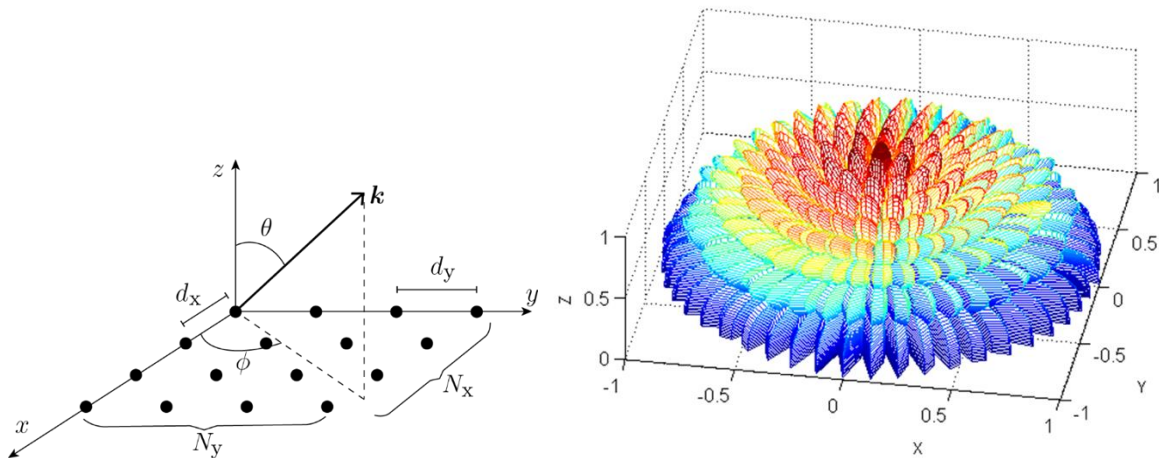


Figure 8.3: (Left) A UPA with N_x by N_y antenna elements. (Right) The beam codebook for an 8×8 UPA. The beams are separated by their half-power (or 3 dB) beamwidth. We

assume there is a grounded backplane so that there is no radiation toward the back (-z direction is this case). There are 87 beams to cover the entire field of view of this 8x8 UPA.

Finally, we describe the beam alignment problem. The beam alignment comes down to the problem of selecting the pair of transmit and receive beam (one of the beams shown in Figure 8.3 (Right)) that provides the highest received power. Without any prior knowledge, the alignment algorithm will need to test all the combinations of the transmit and receive beam pairs. If using the 8x8 UPAs shown in Figure 8.3 (right) at both the transmitter and the receiver, there are 87×87 combinations to be tested. Our proposed solution learns from past experience at a given location to determine which directions are likely to provide strong link connections. As will be shown later, our approach can achieve less than 1% misalignment probability by testing or training only 30 beam pairs as compared to 87^2 when doing the search blindly.

8.1.2 Overview of DSCR-aided Beam Alignment Framework

Figure 8.4 illustrates the proposed beam alignment solution when operated in the uplink (i.e., when the car transmits to the RSU). The proposed framework has two phases. Phase 1 is the training request. The car sends the request along with its position (e.g., included in a BSM) to the RSU. The RSU inputs the information into the recommender system to produce a list of promising beam pairs. It then responds back to the car with an acknowledgment (ACK) along with a list of recommended beam indices. Because Phase 1 happens before mmWave beam alignment and mmWave cannot yet support a reliable link, DSRC should be used. In Phase 2, the beam pairs in the list are trained. Once completed, the RSU sends back to the car the beam index that provides the highest received power.

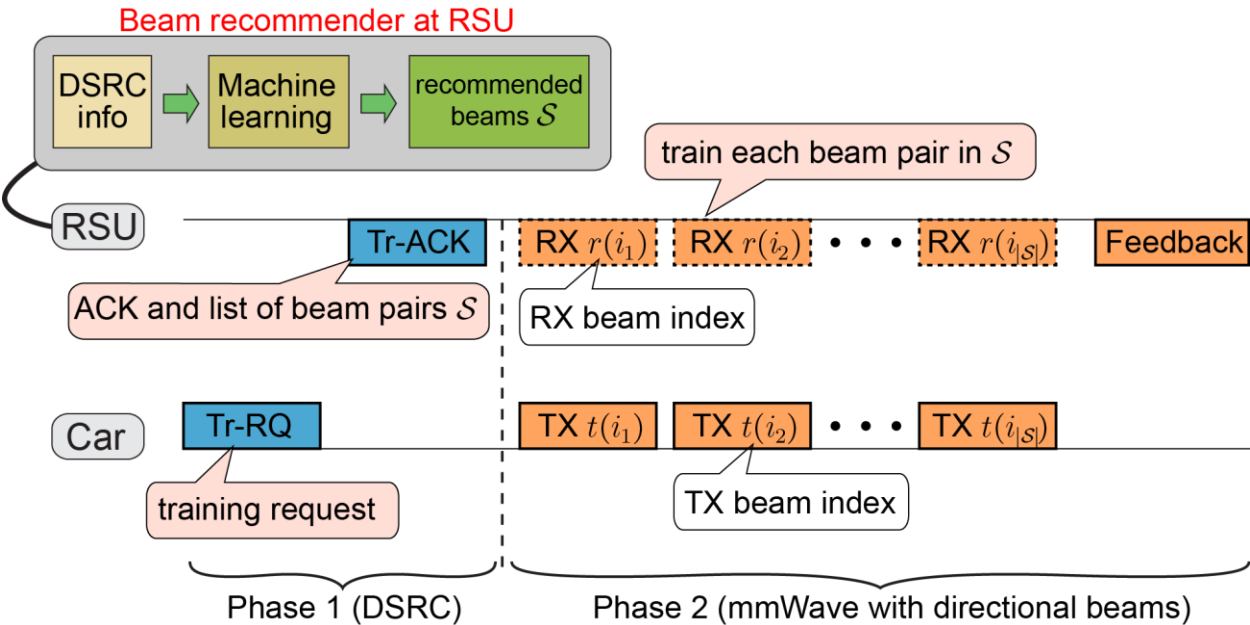


Figure 8.4: An illustration of the DSRC-aided mmWave beam alignment framework

In the downlink, Phase 1 changes slightly. The training request is initiated by the RSU. The car then responds with an ACK and its position. The RSU will use the position to query the recommender system to get the list of beam pairs. It sends back the list, which signifies the start of the beam training. That is Phase 1 in the downlink, which will use three transmissions. This is needed because the recommender system resides at the RSU. Phase 2 is kept unchanged. By having the car transmit during the beam training, the RSU can collect the beam measurements without any feedback from the car. These measurements can be used to update the recommender systems and are necessary when the recommender is trained online, as will be shown in Section 8.1.4.

The core of this beam alignment framework is the beam recommender system. The rest of this section, presents two types of beam recommender system. The first one is an offline learning approach. It provides a flexible and scalable framework that can take various types of information as the input beyond just the position as is currently the case. The second solution is an online learning method. It requires a minimal database (of beam measurements) for initialization. Its performance can be expected to improve over time since the online method keeps collecting new measurements and updates its learned parameters. This approach, however, can only use the position as the input and does not scale well when incorporated with various other types of information. Such an extension is an interesting future research direction.

8.1.3 Learning-to-rank Beam Recommendation

In this subsection, we present our offline learning for beam recommendation. We will start with a problem description, which is followed by some motivation. Then, we present the main idea of the solution. Next, we describe the proposed solution. Finally, we will show some numerical evaluation.

Problem: This is an offline learning problem with the goal of producing a ranked list of beam pairs that maximizes the likelihood of finding the best beam pairs (i.e., having the highest received power). The training data is assumed already collected (i.e., offline learning). Table 8.1 shows an example of the training data. Each row consists of the position of the car when conducting the beam measurements and the sorted beam pair indices and the received powers.

Table 8.1: An example of an offline training data for a LtR beam recommendation.

Each row is a set of beam measurements conducted by a contributing vehicle. It has the position of the car, and the beam pair index (top of the cells) and the received power (bottom of the cells) sorted by the received power strengths.

Position	Best	2 nd best	...	<i>k</i> -th best
x_1	5 -64.5 dBm	159 -69.2 dBm	...	346 -95.8 dBm
x_2	159 -70.4 dBm	263 -72.6 dBm	...	354 -97.1 dBm
...
x_N	5 -66.4 dBm	258 -68.1 dBm	...	2 -82.6 dBm

Motivation for the proposed solution: With the given training data in Table 8.1, a solution could be to discretize the position in grids and then use measurements (rows) from the same grid for statistical estimation of the parameters of interest, such as the probability of being aligned or the average received power of a beam pair. Indeed, this is the approach of our prior work [8.7]. This approach performs well, but it does not scale well with increasing context information. Here, only the position is used as the context. If we have additional context information, such as time of the day and other sensor data, the number of context grids increases exponentially with the number of types of context. Also, if the system has access to accurate context information (e.g., up to cm-level position information), the grid size should be small. This means the number of grid points becomes large and is another source of a scalability issue. This scalability problem is the motivation for developing our LtR method, which does not require discretizing the context. While the developed solution can incorporate different types of context, due to limited training data, our numerical results only use position information.

Solution idea: Here, we describe the main idea of the proposed solution. Common to most machine learning problems, choosing the model of the learning and the cost function are probably the two most important steps.

We choose the kernel-based model for our solution for two main reasons. First, our context is low-dimensional and thus it does not make sense to use a complicated model such as neural networks. Deep neural networks can learn features in high dimensional input such as pixels of images, but this is not necessary for simple processed context information in our case. Second, while not too complicated, we do not expect the linear model to work well either. The kernel-based model allows the introduction of non-linearity by the choice of the kernel function. Indeed, our numerical results show poor performance when using a linear kernel.

For the cost function for the learning problem, we choose a metric for assessing the accuracy of a ranked list. Note that common types of learning problems such as classification or

regression problems try to predict a scalar output. Here, the problem is to predict a structured output, specifically a ranked list. To evaluate the quality of a predicted ranked list, the metric needs to be aware of the ranks. One approach to reflect this is to introduce a discounting factor that has a low value for a low rank. This way, the metric will emphasize the correctness of the top-rank predictions. Our metric is designed along this idea.

Once the model and the objective function are decided, we can apply a standard optimization technique such as the stochastic gradient descent to learning the model parameters. In our case, an approximation is needed to compute the derivative of the objective function because the original objective function is not continuous.

LtR beam recommender solution: The proposed solution is illustrated in Figure 8.5. It shows how the recommender system works after the learning. The optimization for learning the parameters uses the model and the cost function as mentioned earlier. The output of the learning is the score predictor used in the second block (the green one) in Figure 8.5. The recommendation procedure takes the context information as the input (in this case, the position of the car). It then produces a set of query vectors corresponding to the pointing directions of interest. Each query vector (or feature vector) includes the transmit and receive pointing direction (azimuth and elevation) of a beam pair and the context. Additional context can be appended to this feature vector while using the same learning procedure. Therefore, the system is flexible for incorporating different context information. Next, for each feature vector, the system computes a score using the learned model. In the final part, the system sorts the predicted scores and outputs the beam pair indices that have the top- N_b highest scores.

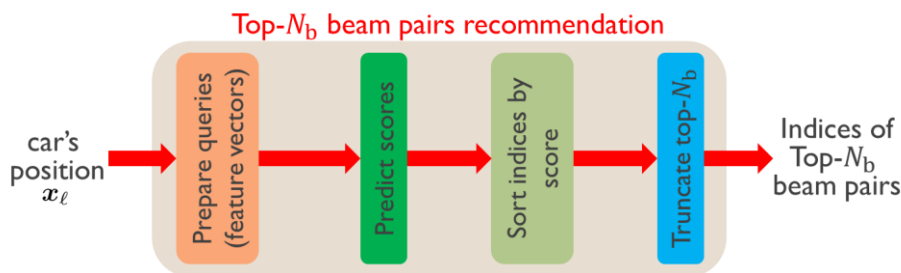


Figure 8.5: Top- N_b recommendation beam pair recommendation system

Numerical evaluation: Here, we evaluate and compare the proposed LtR beam recommendation method with the baseline from our prior work in [8.7]. We perform a five-fold cross-validation on 500 channel samples generated following the description in Section 8.1.1. This means we divide the data into five subsets of 100 samples each and we select one of the subsets as the test set and use the other four subsets as the training set. This is repeated five times; each time a different subset is selected as the test set. Each channel sample corresponds to an independent placement of the vehicles in the environment. This also includes the location of the communicating car, which is drawn uniformly from the range 27.5 m to 32.5 m from the RSU (d_ℓ in Figure 8.2). The baseline method treats this 5-m range as a location bin, while in the proposed LtR approach the actual position is included in the feature vector when computing the prediction. We assume

16x16 UPAs at both the RSU and the car. The beam codebook has 271 beams, and an exhaustive search would need to train $271^2 = 73441$ beam pairs. Our methods only need to train around 30 beam pairs to achieve the alignment with high probability.

Figure 8.6 shows the misalignment probability of the proposed LtR method when using different kernel functions as compared to the baseline. We can see that the linear kernel does not work at all. The radial basis function (RBF) kernels perform well. We proposed a slight modification to the standard RBF kernel, which provides the best performance as shown in Figure 8.6. With the modified RBF kernel, the proposed solution achieves <1% misalignment probability (i.e., >99% alignment probability) when training only 30 beam pairs. We can also see the performance gain over the baseline for a lower target misalignment probability at low target misalignment probability. For example, for a target misalignment probability of 0.1%, the training overhead reduction is >20% as compared to the baseline.

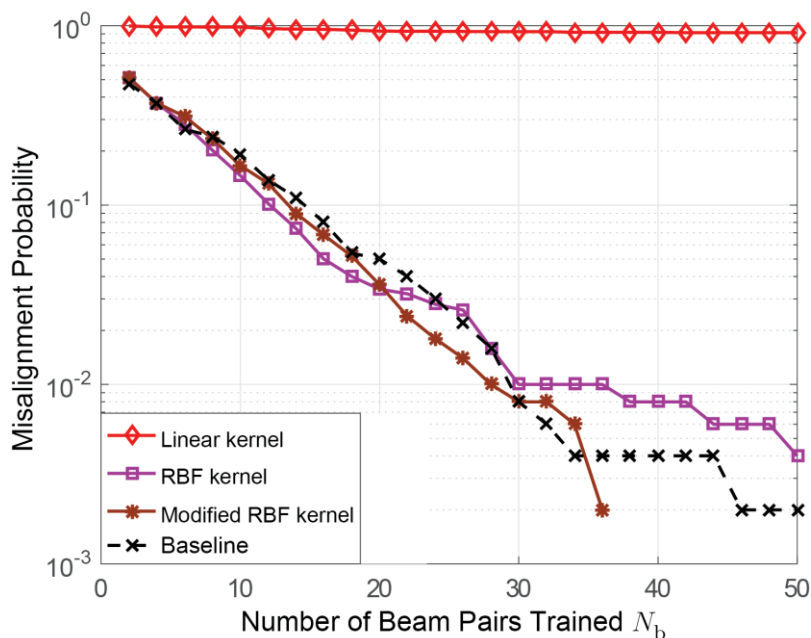


Figure 8.6 Misalignment probability of the LtR beam recommendation method when using different kernels as compared to the baseline from [8.12]

8.1.4 Online Learning for Beam Recommendation

In this subsection, we present our online learning solution for beam pair recommendation. The structure of the presentation is the same as in the previous subsection.

Problem: Our goal is to develop an online learning version of our prior approach from [8.7]. Here, we discretize the position as opposed to the LtR method in the previous subsection. The challenge here is that there is no full database up front as in the offline learning setting (as in [8.7] and the LtR method). The algorithm only uses a small initial database for initialization. In its canonical form, our problem is this: given a position, what beam pairs should be recommended based on the observations seen so far? In other words, this is a sequential decision-making

problem. Such problems can be solved using the MAB framework. Our beam alignment method (Figure 8.4) allows the training of $N_b \geq 1$ beam pairs, and so we can cast our problem as a multiple-play MAB problem (similar to betting on multiple numbers in a roulette game). The online learning algorithm has to exploit its current knowledge as well as enhance its knowledge of the learning model. This is known as the explore-exploit dilemma and is intrinsic to online learning problems. Note that exploration comes with the risk of poor performance since the algorithm tries uncertain actions to learn. Our goal is to design an algorithm that understands this risk and tries to minimize the risk during the learning.

Motivation for the proposed solution: There are several advantages of learning online. First, offline solutions cannot be deployed until the database is collected. Furthermore, the performance of an offline learning solution depends entirely on the accuracy of the database that is not updated. Online approaches keep collecting new observations during operation, making it possible to improve the database. In addition, online learning approaches open up opportunities for tuning the system based on the observed response. For example, as opposed to using the same beam codebook in every environment, an online learning method can be designed to tune the codebook to fit the environment and maximize the beamforming gain (i.e., the received power).

Solution idea: There are two main features in our solution. One is to solve the multiple-play MAB with low complexity and the other is to minimize the risk during the learning.

For solving the multiple-play MAB problem, it is important to select a good reward signal and exploit the structure of the reward signal. In our case, we use the alignment probability as the ideal average reward. There are two main reasons for this choice. First, we have shown in [8.7] that using the alignment probability for beam pair recommendation is optimal (in the sense of minimizing the misalignment probability). Thus, this is a natural choice to try to approach in the online approach. Second, the additivity of the alignment probability allows a low complexity solution when computing the recommendation. Additivity here means that the alignment probability of a set of beam pairs is equal to the sum of the alignment probability of each individual beam pair. This means that we can select one beam pair at a time when doing the recommendation (called greedy selection), which makes the problem much simpler. Note that such a subset selection problem is intractable in general. Our proposed solution uses the upper confidence bound (UCB) of the alignment probability as the selection metric. A UCB has two parts: the expected reward based on the observations so far and the uncertainty. ‘Uncertain’ means that the system has tried the beam pair only a few times, and it does not have enough information to reliably estimate the alignment probability. If a beam pair has a high UCB value, it means that the alignment probability of the beam pair either is known to be high or is uncertain. The former enforces exploitation, while the latter enforces exploration. We call this selection the greedy UCB selection.

We now discuss the other main feature of the solution: risk-awareness. The greedy UCB selection is oblivious to the multiple-play setting. This means that it could select too many beam pairs whose expected rewards are highly uncertain (thus, large UCB values). This means that the risk of misalignment is high. Observing this, we introduce the notion of risk to each beam pair and make the algorithm reject beam pairs selected by the greedy UCB method with a probability

proportional to the risk. We define the risk of a beam pair as the probability that its received power is below a certain threshold as compared to the received power of the strongest beam pair. This allows us to use a binary signal to estimate the risk during the operation. Now, when a beam pair is rejected, we used the expected rewards (i.e., without the uncertainty terms) to select a replacement. We call this risk-aware selection the risk-aware greedy UCB algorithm.

Online learning solution for beam pair recommendation: The proposed online solution operates as shown in Figure 8.7. The algorithm starts with a training request detection loop. When detecting a request, it decodes the car’s position and input to the beam recommendation algorithm. The algorithm reads the learning parameters corresponding to the position and produces a recommendation list. The list of beam pairs is then sent to the car and the beam pairs in the list are trained. The beam measurements are used to update the learning parameters and the algorithm returns to the training request detection loop.

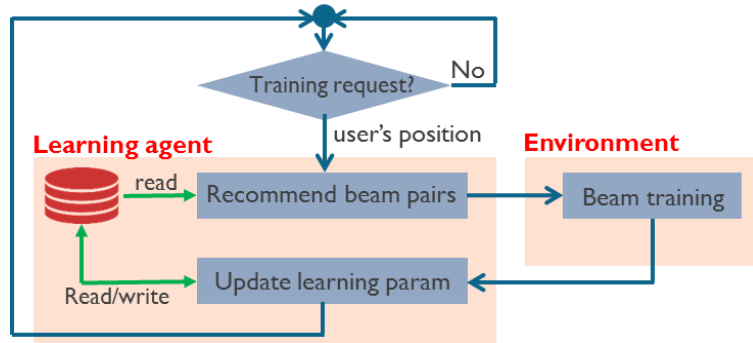


Figure 8.7: A flowchart of the proposed online learning for beam pair recommendation

We now describe the risk-aware greedy UCB algorithm for producing the recommendation. Figure 8.8 illustrates the procedure for recommending N_b beam pairs. First, it computes the UCB values of all beam pairs. Then, the algorithm enters a loop, where in each iteration it selects a beam pair and add it into the recommendation list. The loop starts by checking if N_b beam pairs are already selected or not. If yes, it outputs the recommendation list and terminates. If no, it selects a beam pair with the largest UCB values among the pairs not yet in the current recommendation list (i.e., the greedy UCB selection procedure). Next, it estimates the risk of the selected beam pair using the parameters of the beam pair stored in the memory. It then makes a random rejection with the rejection probability proportional to the risk. This is done by drawing a Bernoulli random variable (i.e., tossing a coin) with the risk as its parameter. If rejected, a replacement beam pair is selected using the expected reward (as opposed to the UCBs, which contain uncertainty). The selected replacement is then included into the recommendation list. If not rejected, then the original greedy UCB selection is included into the recommendation list. The algorithm then returns to the start of the selection loop. The loop iterates until N_b beam pairs are selected.

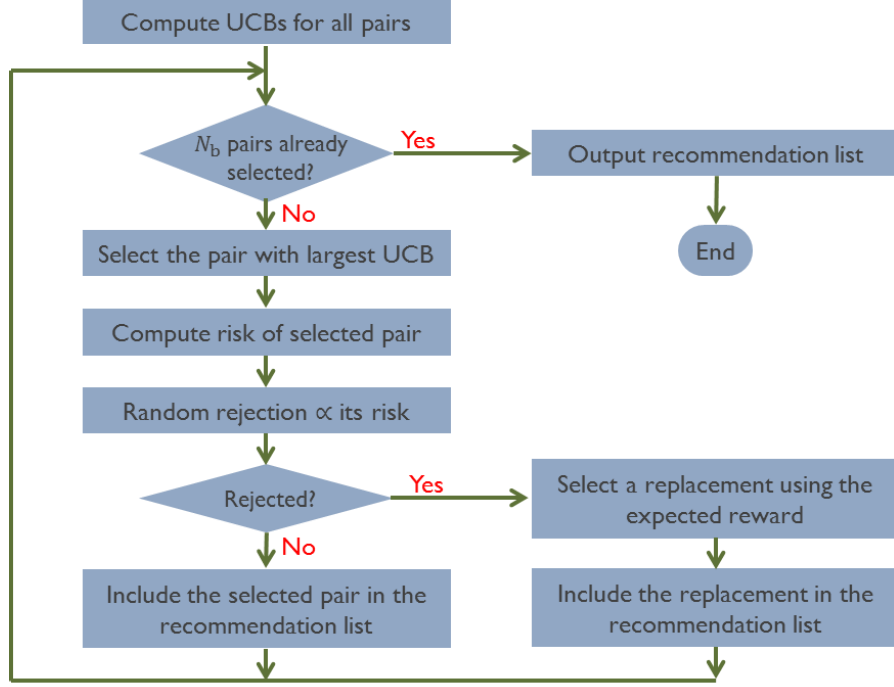


Figure 8.8: Flowchart illustrating the risk-aware greedy UCB algorithm

Numerical evaluation: We show the learning behavior of the proposed online learning system over time. We again use the channels generated following the description in Section 8.1.1. We generated 10,000 channel samples for the evaluation here. To eliminate the dependence of the ordering of the channel samples during the online learning, we run the online learning algorithm 100 times, where in each time we randomly permute the channel samples. As the performance metric, we use the 3 dB power loss probability averaged over the 100 simulation runs. The 3 dB power loss probability is the probability that the received power after conducting the proposed beam alignment is within 3 dB of the optimal alignment.

Figure 8.9 shows the learning behavior of the proposed risk-aware greedy UCB algorithm. The algorithm has two main parameters: the training budget N_b and the risk threshold Γ_{risk} . We note that our algorithm does not require that N_b be fixed, but for simplicity, we assume the same N_b during the learning here. Figure 8.9 (Left) shows the performance for $N_b = 10, 30$ with different Γ_{risk} . We can confirm that using a larger training budget N_b leads to a lower 3 dB power loss probability, i.e., more accurate beam alignment. The learning seems to have two phases: the fast improvement phase in the early time steps and the slower improvement phase after that. For $N_b = 30$ beam pairs and $\Gamma_{\text{risk}} = 5$, this phase change happens at around time index 500. The slower learning phase starts when the algorithm has identified high-risk beam pairs (with some certainty) and learns those beam pairs at a slow pace using the rejection mechanism. For the risk threshold, the algorithm is not sensitive to the choice of Γ_{risk} as long as it is not too large (e.g., 40 dB). The main reason for this behavior is due to the effect of the replacement selection (see Figure 8.8). Even if a good beam pair is rejected due to risk overestimation (when using a small Γ_{risk}), it

will likely be picked up by the replacement selection. Using a too large Γ_{risk} essentially reverts the algorithm back to not being risk-aware and results in poor performance.

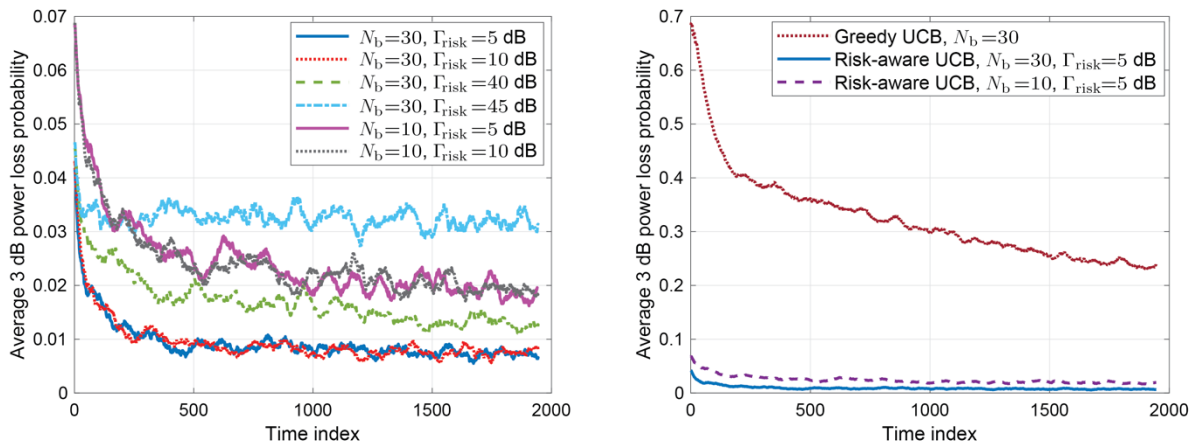


Figure 8.9: (Left) The effect of the two main parameters (N_b and Γ_{risk}) of the risk-aware greedy UCB algorithm. (Right) Performance comparison of greedy UCB algorithm with and without risk-awareness.

Figure 8.9 (Right) compares the performance of the online learning with and without risk-awareness. The performance without risk-awareness is an order of magnitude worse than that of the risk-aware case. This is because, without risk-awareness, the algorithm aggressively samples uncertain beam pairs (large UCB values). The risk-awareness feature balances the selection of high-risk beam pairs and encourages the algorithm to exploit more. While the risk-awareness component slows the learning of those high-risk beam pairs (i.e., obtaining accurate their learning parameters), the instantaneous performance at each time index is much better. This means that the risk of large power loss is distributed more evenly among the cars at different learning stages (each time index corresponds to the beam alignment of a different car), and the algorithm does not place too much burden on cars during the early stage of the learning.

8.2 Vehicular mmWave Reflection Measurements

In this section, we present our measurement campaign aiming for assessing the usability of the reflection path via a neighbor vehicle. We start with the motivation. Then, we describe the measurement system and the setup of the measurement scenario. Finally, we report the results.

8.2.1 Motivation

The reflection path off a neighbor vehicle could be an alternative path to the LOS path under blockage situation, and it can help make mmWave link connection more reliable. Blockage is an important characteristic of mmWave propagation. When vehicles move together in the same direction on a highway, blockage by a neighbor vehicle can last for a long time (e.g., several

seconds). Under such a long blockage, it is crucial to find an alternative path to keep the mmWave link connection. Indeed, in our simulation studies using the ray-tracing simulator conducted so far for the beam alignment, we observed that reflection paths play an important role. Due to the complexity of the simulator, we cannot model all the details of the vehicles (we used boxes made of metal in our simulations), and so it is important to understand such reflective paths in actual measurements.

Furthermore, such a reflection path could have an important implication for V2V mmWave communications. With DSRC, vehicles can obtain information about their neighboring vehicles, including their position and sizes. If the reflection path can serve as an alternative to the LOS path and the position information is accurate enough, a vehicle can determine if its LOS path is block and compute the pointing direction for the reflection path directly from the geometry.

8.2.2 Reflection Measurement: Equipment and Setup

In this subsection, we present the channel measurement equipment and the details of our measurement campaign.

Measurement equipment: There are three main components in our channel sounder system: a National Instruments (NI) mmWave transceiver system, a Rubidium frequency standard, and a rotating table. The NI mmWave transceiver system supports all the processing from baseband, intermediate frequency (IF) to the radio frequency (RF) that radiates signals in the environment. The transceiver system is a wideband system with a symbol sampling rate of 1.536 Giga-samples per second. This means it has a time resolution of 0.65 ns. Our system uses radio heads that support the frequency bands in between 71-76 GHz. The Rubidium frequency standard is used to provide the exact timing to the transmitter and the receiver so that they are tightly synchronized during the measurement. Such synchronization is needed for post-processing for identifying the different paths. Finally, the rotating table is used to obtain measurements at different pointing direction (we use directional antennae).

We used two types of horn antennae in the measurement campaign: a wide-beam of $\sim 40^\circ$ antenna and a narrow-beam of $\sim 10^\circ$ antenna. The two antennae with the radio heads are shown in Figure 8.10. The wide-beam antenna is used at the transmitter to illuminate the whole car (the reflector) and we rotate the narrow-beam antenna (in the azimuth plane) to measure the channels in the angular domain.

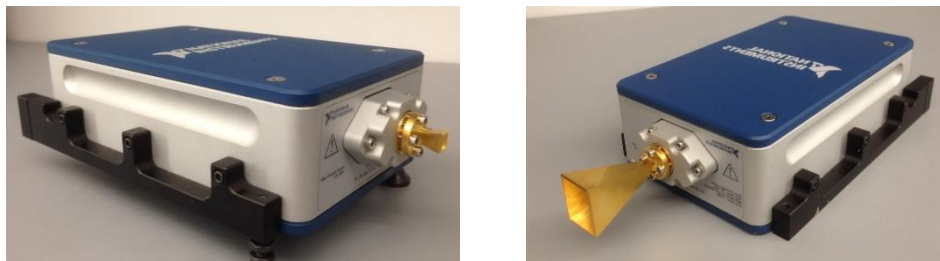


Figure 8.10: Horn antennae used in the measurement. (Left) 40° beamwidth horn antenna; (Right) 10° beamwidth horn antenna.

Measurement scenario: Since our goal is to measure reflection off a car, the measurements can be done in a static setting. The measurement setup is shown in Figure 8.11. The measurements were conducted on the top floor of a garage on campus. As mentioned earlier, we use a wide-beam antenna at the transmitter and its pointing direction is fixed. The receiver uses a narrow-beam antenna, and the receiver radio head is placed on the turntable, which we rotated to measure in the azimuth angular domain. The angular sweep starts from the LOS direction with a rotation step of 5° until it reached 70° away from the LOS. This sweep range covers the entire reflecting car.

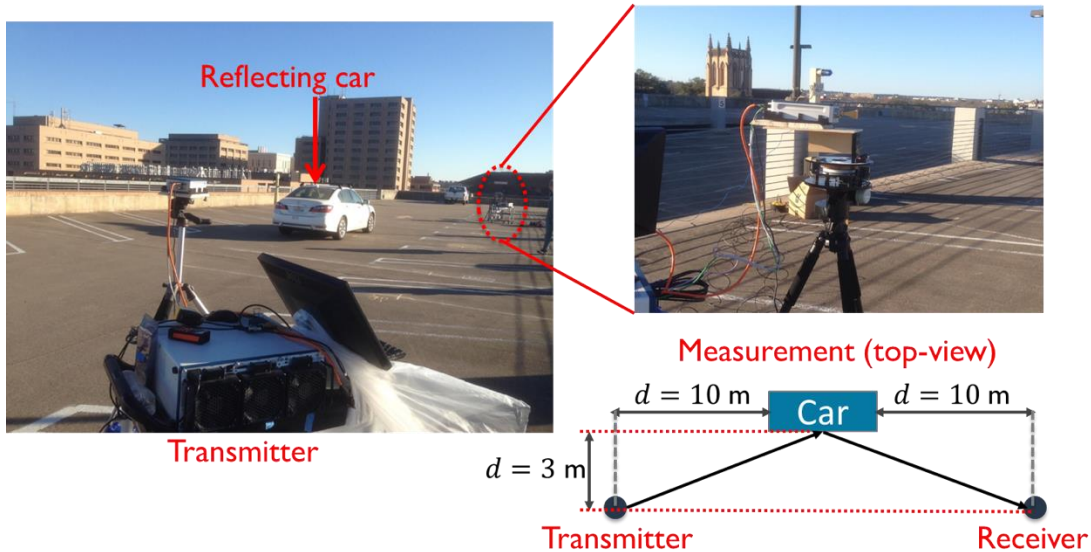


Figure 8.11: Reflection measurement setup. The measurements were conducted on the roof of a garage on campus.

In the measurement campaign, we measured 14 configurations (as listed in Table 8.2) that cover different antenna heights and antenna polarizations. We used three heights: 0.5 m, 1.0 m, and 1.5 m. These three choices imitate the cases where the antenna is mounted on the bumper, the side mirror, and the roof of the car. The antenna polarization is the axis of oscillation of the electric field as the electromagnetic wave radiates from the antenna. It is well known that propagation at mmWave preserves the polarization more than propagation at a lower carrier frequency. This means that antenna polarization mismatch can result in a large loss in the received power. To measure this loss, we use two configurations for the antenna polarization: horizontal-horizontal (H-H) and horizontal-vertical (H-V). We also measured the case when there was no reflecting car. The purpose is to measure the background response of the environment.

Table 8.2: Measurement configurations

Configurations	Rx height	Tx height	Polarization (Rx-Tx)	Note
1	0.5 m	0.5 m	H-H	With car
2	0.5 m	0.5 m	H-V	With car
3	0.5 m	1.0 m	H-H	With car
4	0.5 m	1.0 m	H-V	With car
5	0.5 m	1.5 m	H-H	With car
6	0.5 m	1.5 m	H-V	With car
7	1.0 m	1.0 m	H-H	With car
8	1.0 m	1.0 m	H-V	With car
9	1.0 m	1.5 m	H-H	With car
10	1.0 m	1.5 m	H-V	With car
11	1.5 m	1.5 m	H-H	With car
12	1.5 m	1.5 m	H-V	With car
13	1.5 m	1.5 m	H-H	No car
14	1.5 m	1.5 m	H-V	No car

8.2.3 Measurement Results

Before presenting the measurement results, we start with a description of how we identify the reflection path. We identify the reflection path using both the delay and the angular domain. For the delay domain, we can use the LOS path as the reference. The LOS path is the first peak in the power of the channel impulse response (CIR). Then, by computing the delay difference corresponding to the path length difference between the LOS and the reflection path, we can find the delay tap corresponding to the reflection path. Specifically, using the detailed geometry of the measurement setup shown in Figure 8.12, we can compute the path length difference to be about 0.7 m. The sample period of the system is 0.65 ns, and so using the speed of light of 3×10^8 m/s, we can translate the path length difference to a delay of about 3.6 samples. Thus, we expect the tap corresponding to the reflection path at 3 to 4 samples away from the LOS tap. Similarly in the

angular domain, from the geometry, we expect the peak of the reflection (highest received power) when we rotate the receiver close to 13.5° .

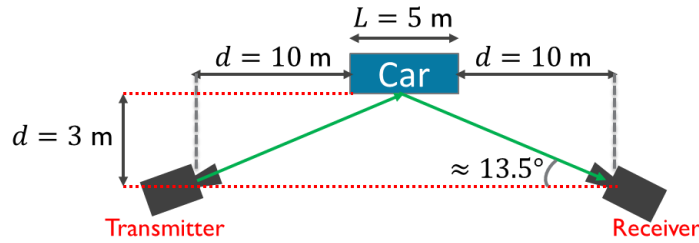


Figure 8.12: Detailed geometry of the measurement setup

We now present the results. We start with the case where the transmitter and receiver are of the same heights (configurations 1-2, 7-8, and 11-12). Figure 8.13 shows the results when the height is 0.5 m. The dashed curves are those with matched polarization, and the solid curves are those with polarization mismatch. We can see that the loss due to polarization mismatch is large. It is up to about 20 dB for the LOS path and about 15 dB for the reflection path. The smaller loss for the reflection path is likely because the reflection changes the polarization of the incident waves. In the matched polarization case, we can see that the received power of the reflection path is about 5 dB below the LOS. Such a loss of this magnitude will reduce the data rate, but it likely will not enough to break the mmWave link. This means the reflection path can be used as a backup link. We observed a similar trend for the case of 1.0 m high. For the 1.5 m case, the trend is a bit different. The car reflection path is slightly stronger than the LOS received power when the polarization matches. One difference from the previous two cases is that the reflection point on the car is likely on the glass window area, which is made of different material than the car body. One hypothesis is that the material in this area provides a better reflection than the car body. Further measurements are needed to make a conclusive statement on this.

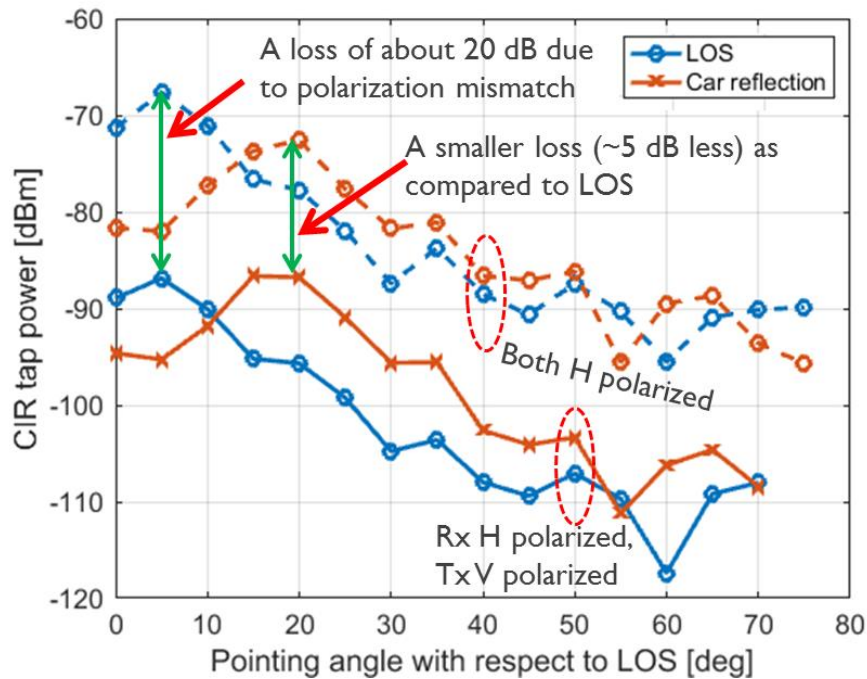


Figure 8.13: CIR tap power when rotating the receiver. Both the transmitter and receiver are at 0.5 m high.

We next describe the cases when the transmit and receive antennae are of different heights. Figure 8.14 shows the measurement results for the case when the receiver is 0.5 m high, and the transmitter is 1.0 m high. We can see that while the trends are still the same, the gaps here are larger than the case when the transmitter and the receiver are of the same height. With matched polarization, the gap between the reflection path to the LOS path increases to about 10 dB. With mismatched polarization, the reflection path is about 10 dB stronger than the LOS path. In the difference cases measured, it seems that the cause of this increase in the gaps is mainly because the LOS received power changes by a large value for the different measurement configurations. The received powers of the reflection path seem to be of similar values for these configurations. One possible explanation for this behavior is that the LOS possesses a two-path effect while the reflection path does not. In the LOS direction, besides the direct LOS path, there is also the ground reflection path. The two paths can add constructively or destructively, depending on the exact phase difference between the two paths causing the received power fluctuation. Further measurements are needed to confirm the correctness of this explanation.

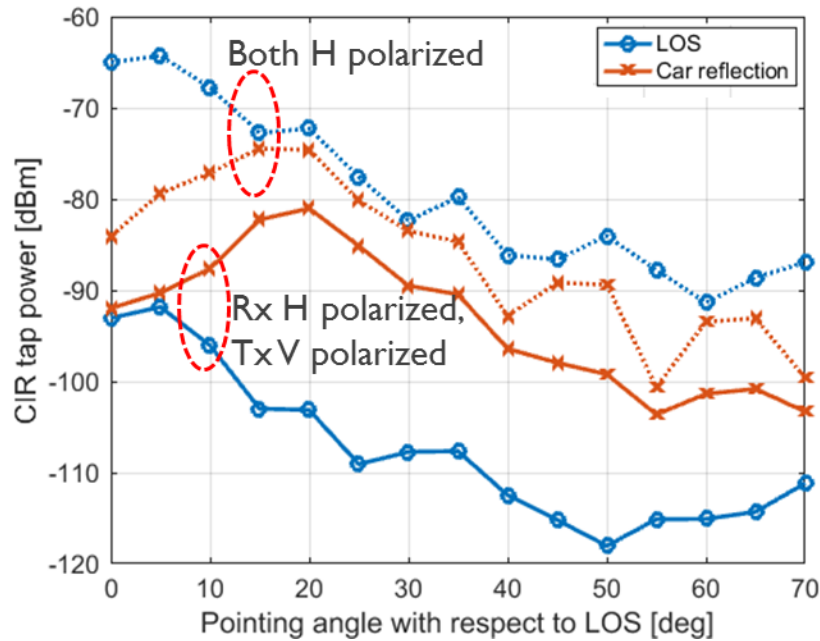


Figure 8.14: CIR tap power when rotating the receiver. The receiver is 0.5 m high, and the transmitter is 1.0 m high.

8.2.4 Summary

We first summarize the main results from the measurements. When the transmit and receive antennae are of the same height, the reflection loss is within about 5 dB as compared to the LOS path. This is a promising result showing that the reflection path is a viable alternative to maintain mmWave link connection in the LOS blockage situation. The loss due to polarization mismatch is large. The loss can be more than 20 dB for the LOS path and slightly less at around 15 dB for the reflection path. For the case with different antenna heights, the gap between the LOS and the reflection increased up to around 10 dB for the matched polarization case.

We now discuss several issues that are not fully explained by the current measurement results. The reflection path was slightly stronger than the LOS path when both the transmit and receive antennae are at a height of 1.5 m. The gaps in the received power between the LOS and the reflection path increased when the antenna heights are different. When both antennae are at 1.5 m high, the increase could be due to the different material of the reflection point on the car. It can also be explained by the two-path effect on the LOS received power. The two-path causes the LOS received power to decrease (the LOS path and the ground reflection add destructively), leading the LOS received power to become smaller than the power of the reflection path. When the antennae are at different heights, the increase in gap could also be explained by the fluctuation in the received power of the LOS path due to the two-path effect. While qualitative explanations of the behavior are possible, further measurements are needed to confirm the correctness of these explanations.

Chapter 9 Radar-aided mmWave Infrastructure Communication

9.1 Introduction

Radar is one of the sensors typically mounted on a vehicle or an RSU that can be used to obtain the side information for in the communication system. This information can be the position and velocity that can help in configuring the mmWave beams. In this chapter we will introduce the concept of radar-aided vehicular communication, identifying how radar can be a useful source of information that helps configure the mmWave link.

We start this chapter by showing radars mounted on BS can be a source of high precision positioning that can reduce the mmWave beam-training overhead. Then, in Section 9.2.2, we show the congruence in radar and communication spatial spectrum, and using the congruence configure the mmWave precoders/combiners without in band training. Subsequently, in Section 9.2.3, we outline a modification to the 5G NR based on eigenvectors of radar covariance, which can eliminate the initial access overhead.

We outline the equipment bought for measurements and prototyping radar-aided mmWave communications in Section 9.3.1. In Section 9.3.2, this equipment is used to establish the congruence of radar and communication in real world settings. Further, in Section 9.3.3 we use the equipment to study the impact of radar radiations on mmWave communication.

9.2 Radar-aided V2I mmWave Communication

9.2.1 Positioning Using BS Mounted Radar

One way to reduce the array configuration overhead is to exploit side-information about the mmWave channel. Vehicle location has been used as side-information in the past [9.1]. In light-of-sight (LOS) vehicle-to-infrastructure (V2I) communication links, the location of the vehicle can be used to reduce the beam-training overhead. Specifically, the beamforming codebook can be reduced to a few beams around the estimated vehicle location. The reduction in codebook depends on the accuracy of localization, i.e., more reduction with accurate localization.

The location of a vehicle typically comes from the global navigation satellite system (GNSS). As an alternative, a BS-mounted radar—referred to in this report as the BS-radar—can determine the precise location of the vehicle. Based on the Cramer-Rao lower bound (CRB) on the angle estimation of an off-the-shelf radar [9.2] we show that a BS-radar can localize a vehicle better than GNSS. Specifically, we consider a V2I communication setup shown in Figure 9.1. The height of the roadside lamp i.e., BS is h_{BS} , and the height of the vehicle is 1.6 m. Assuming that two BSs are separated by inter-site distance (ISD), and focusing only on the coverage area of a single BS, we consider a road segment of length ISD. The distance between the BS and the closest point on the road is d .

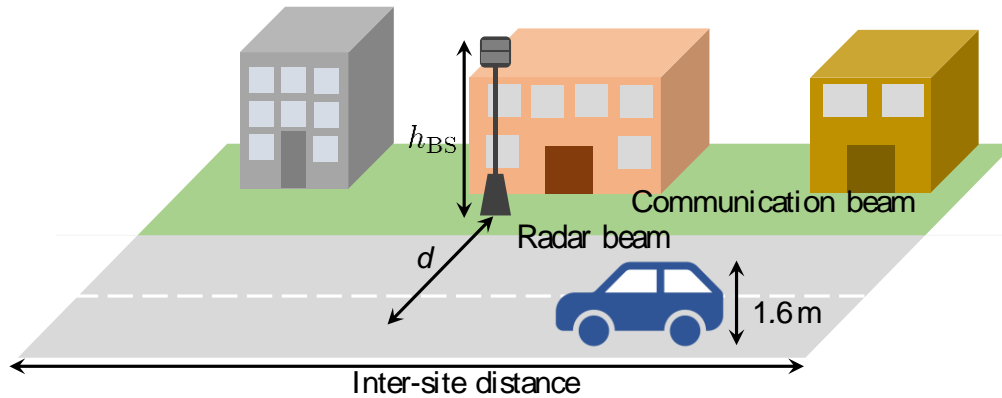


Figure 9.1 The V2I communication system with BS-mounted radar

Typically, a vehicle obtains its location from GNSS. The GNSS receivers are relatively inexpensive and ubiquitous. The location information from GNSS, however, is erroneous. The error in the location information of the GNSS system is modeled as a disk of radius r . The GNSS error radius r is around 10 m [9.1]. The location of the vehicle can be shared with the BS using current vehicular communication standards. A basic safety message (BSM) is broadcasted every 100 ms [9.3]. To understand the impact of erroneous information, we consider an example based on the setup in Figure 9.1 The V2I communication system with BS-mounted radar with ISD 100 m, h_{BS} 5 m, and d 10 m. Assume that the vehicle is at the broadside of the BS array, and is moving with the speed of 140 kmh (i.e., the speed of vehicles on a highway), the true location and the reported location can have an offset of 10 m. Assuming 64 BS antennas, and DFT codebook, 10 m uncertainty implies that we need to test 48 codewords out of the 64 codewords in the DFT codebook. The training overhead reduction using GNSS, therefore, is $(1-48/64)\%=25\%$.

For radar based localization, we consider that the radar is colocated with the communication array at the BS. One example is the INRAS Radarbook, which can operate at 24 GHz and 77 GHz. The transmit power of the Radarbook can go up to 11 dBm. The Radarbook has 8 receive antennas and 4 transmit antennas. The transmit antennas are placed such that the overall aperture of the antenna array is that of a 29 antenna SIMO radar. Doing simple calculations based on the CRB of the radar, we can show that for radar, the AoA estimation uncertainty translates to 2 codewords and training overhead reduction using radar is $1-2/64 \%=97\%$. A vehicle at the broadside of the BS-radar is an optimistic situation, and in practice the accuracy may not be this high. Thus, the savings may be less than 97%, but the example demonstrates the benefit of the BS-radar for vehicle localization.

We assume ISD=100 m, $h_{BS} = 5$ m, $d=10$ m, and 73 GHz operating frequency. We generate 1000 channels by randomly dropping the vehicles on the road and provide simulation results averaged over these random drops. For simulation results, we study the spectral efficiency (SE). The SE as a function of transmit power is shown in Figure 9.2. From the results, we can observe the radar location based beam-training can do better than exhaustive search beam-training and

GNSS based beam-training for both speeds of the vehicle. That said, as the speed increases the SE of the GoB decreases, as a larger fraction of the coherence time is spent training. In comparison, the SE of the radar location based beam-training is not sensitive to the coherence time as the coherence time is very large compared to the training time for even high vehicle speeds.

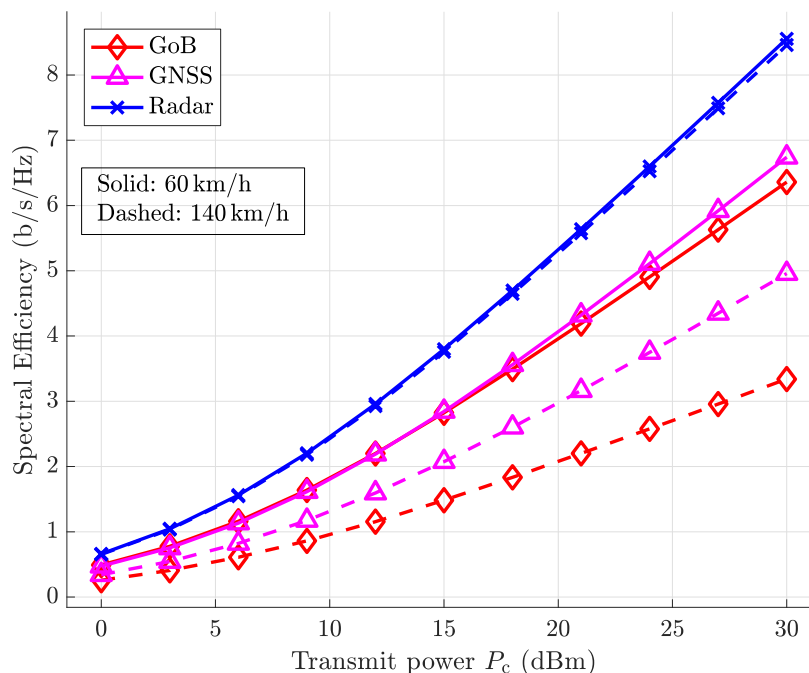


Figure 9.2: The spectral efficiency vs transmit power, with 64 BS antennas and vehicle speeds of 140 kmh and 60 kmh

9.2.2 Radar-communication Congruence and Radar-based Precoding

To enable mmWave in vehicular communication systems, we propose using radar mounted on the road infrastructure to aid in configuring the mmWave communication link in a V2I scenario. In other words, we seek to exploit the information obtained from the radar operating in an mmWave band to extract the channel information of another mmWave band for communications. This approach goes beyond the downlink/uplink reciprocity that has been studied in the past for frequency division duplexing (FDD) communication systems, since this configuration works even when the carrier frequencies may be very different, and the environment, not the channel, is sensed.

For our experiments, we consider an mmWave V2I communication system, where mmWave base stations serve as infrastructure for V2I communications. A radar operating in another mmWave band is mounted on the base station. In real-world environments, depending on the beam width and the distance between the vehicle and the BS, the mmWave beam (radar as well as communication) generated at the BS in a given direction might illuminate only a small fraction of the vehicle. To guarantee that the mmWave communication beam reaches the receiver module at the vehicle, we propose using a variety of antennas on board the vehicle. To achieve this

diversity, we assume that N phased arrays are placed at different points in the vehicle. This approach increases the probability that at least one path from the BS reaches one of the antenna arrays in the vehicle; it can also be used to prevent dramatic effects of blockages in the communications with the vehicle.

The objective of the system is to steer the mmWave communication beams between some of the arrays in the vehicle from the BS, relying on information extracted from the radar mmWave signal. For the mmWave V2I communication system, we consider a hybrid MIMO (multiple input multiple output) architecture, since a multi-beam antenna pattern must be designed to support the multiple receivers. Additionally, since an analog beamforming architecture allows beam steering only in one direction and an all-digital solution requires too much power consumption, we implement a hybrid processing architecture. This hybrid architecture splits the MIMO processing between the analog and digital domains to operate with a number of transceivers smaller than the number of antennas and reduce power consumption. There are similarities with the multiuser communication problem in an mmWave cellular network when the different users (in our vehicular case the different antennas in the car) are located at nearby positions, and this is a commonly applied solution.

Following the above we have designed two protocols for beam alignment in a vehicle-to-infrastructure (V2I) scenario. These protocols are based on a blind design of the precoders/combiners, which does not require channel state information, only the covariance of the received signal. An mmWave radar signal is used to estimate the covariance of the received signal in a different frequency band. Wireless insite [9.4] based ray-tracing results confirm that the main DoAs for the radar and the communication signals are similar, as can be shown in Figure 9.3, for the antenna placed at the top of the car in a ray tracing simulation.

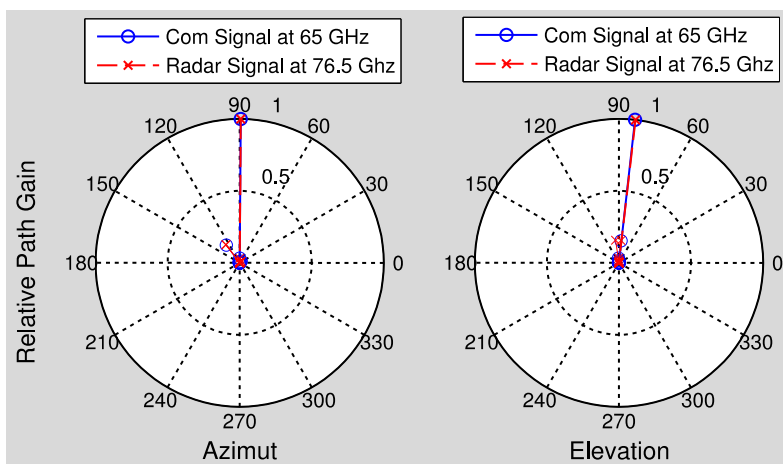


Figure 9.3: Relative path gain in both communication and radar signals

The spectral efficiency obtained for the uplink between two of the transceivers in the vehicle and the BS in a simple scenario is shown in Figure 9.4. The perfect covariance and the sample and compressed estimates provide similar results for the front left antenna, but there is still

a significant gap with the upper bound that could be further reduced with new designs of the combiners/precoders. Here, the upper bound refers to the achievable rate using the perfect channel state information (CSI), which can be obtained only when full beam training is available. However, as mentioned previously, the beam search over the entirety of surrounding space leads to severe training overhead, and so is practically unachievable. Our approach, leveraging the similarity of covariance between the radar and the communication channels, can achieve most of the achievable rate of perfect CSI at high SNR. This implies that our proposed radar-aided mmWave communication system can significantly reduce the overhead without a non-trivial loss of the achievable rate. We propose as future work to design the RF combiners using the maximization of the combined signal power, while finding an alternative criterion for the design of the baseband component. The compressive estimation of the covariance based on LS does not provide good results in some cases, like in the top antenna array, and a better estimator could also be designed.

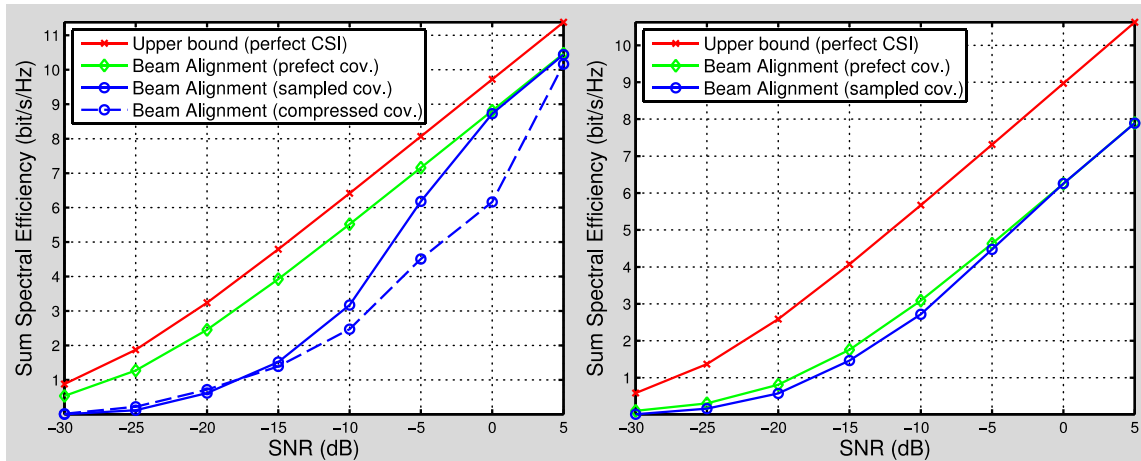


Figure 9.4: The performance result of the proposed beam alignments: (left) the case of left antenna array; (right) the case of top antenna array

9.2.3 Radar-aided 5G NR

In this section, we outline the idea of using radar information in the design of 5G new radio (NR) supporting V2X communication. In preparation for the approval of the first release of the 5G standard, we have been working to determine how to incorporate radar information to speed up beam training and make it feasible for high mobility applications. The main idea of the proposed work is to design the radar-aided mmWave communication from the NR standard point of view, i.e., its direct application in the 5G NR standard. To discuss the proposed ideas, we will first outline some preliminaries about the 5G NR beam-management.

The 5G NR supports frequencies above 6 GHz, and as such, procedures for beamforming and beam management are included in the standard. The basic procedure for beam management in 5G NR is based on an exhaustive search. The transmitter sweeps all the beams, and the receiver measures the received signal on all the transmitted beams and determines the best receive beam

based on the signal-to-noise ratio (SNR) [9.5]. The downside of the exhaustive search is that a significant fraction of time-frequency resources is utilized for link configuration. This implies that these resources are not used for data transmission, resulting in low spectral efficiency.

A simple modification to this process is to aid the NR beam-configuration with position—to restrict the set of transmit beams and receive beams based on the position of the transmitter and receiver. This can lead to significant gains in comparison with the exhaustive search, but even this modification has two limitations.

- i) The position information may not be entirely reliable, as the mechanisms to obtain the position information are inaccurate. This implies that the subset of the beams even after incorporating the position information may be large, not resulting in significant gains. This has been discussed in detail in an earlier section.
- ii) The restriction of beams based on the position assumes that the transmitter and receiver have an LOS. Therefore, directly restricting the beams based on position is not applicable to the non-LOS cases, i.e., a serious limitation.

We propose an alternative way to incorporate radar information into the 5G NR standard. Specifically, we first perform the radar covariance estimation based on the ideas in [9.6], and then use the radar covariance to configure the mmWave link. We provide the details of the proposed procedure below:

Step 1 (*Spatial channel covariance estimation*): First, the radar system operating at the mmWave band is used to estimate the spatial channel covariance. The spatial channel covariance is a second order statistic that is shown to be useful in mmWave link configuration [9.6]. The reason that the spatial covariance is beneficial in mmWave link configuration is twofold. As the spatial covariance is a second order statistic, it varies relatively slowly in comparison with the channel state information. Thus, obtaining, and updating the channel covariance information requires considerably less overhead, which is critical for a successful link configuration strategy. Secondly, due to the relatively sparse nature of the mmWave channels, the use of spatial channel covariance instead of the instantaneous channel information does not incur significant penalty in terms of link configuration performance. Note that typically the spatial channel covariance estimation is performed in-band, thus reducing the channel training overhead but not eliminating it. Our approach is different in the sense that we obtain the spatial channel covariance with a radar that is operating in an adjacent band. This is feasible as the channel's spatial characteristic does not change considerably with frequency. Thus, from an in-band training overhead point of view our approach completely eliminates any training overhead.

Step 2 (*Eigenvector-based precoding*): The beamforming at the transmitter can be performed in several ways. The simplest way is to use a grid of beams, i.e., to point the transmit beam in several carefully chosen directions sequentially and measure the receive

signal at the receiver. There are, however, more sophisticated approaches to use the channel state information. One such approach is the eigenvector-based precoding. This precoding ensures that all the significant directions in the channel are illuminated. We propose to use the eigenvector-based precoding due to its superior performance in comparison with the grid of beams, especially in the channels with multipath.

Step 3 (Exhaustive search at the receiver): Assuming that the radar is mounted on the RSU, it can help only the transmitter, and not the receiver, i.e., the user-equipment in this case. As such, at the user-equipment end an exhaustive search is still required. However, there are only a few antennas at the user-equipment end and as such the grid of beams' exhaustive search is still feasible.

We provide the simulation results for the three approaches: i) exhaustive search (i.e., the 5G NR protocol), ii) position-aided search (i.e., the exhaustive search but the number of beams is restricted based on the position information), and iii) the proposed approach that we call the radar covariance-aided search. The position-aided search was discussed earlier. The simulation scenario is shown in Figure 9.5.

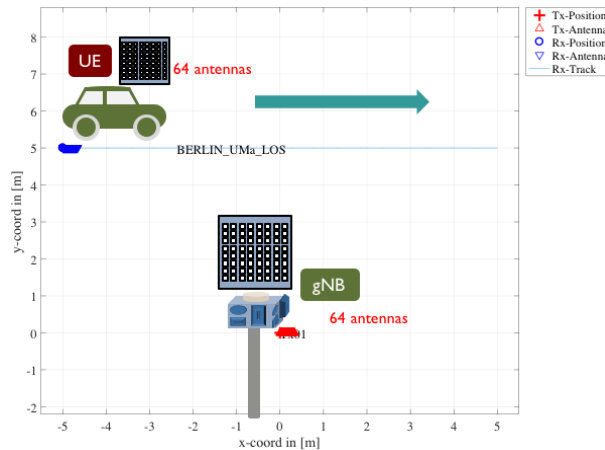


Figure 9.5: Simulation scenario using a realistic mmWave channel model for V2I

For simulations, we assume that the BS and the mobile terminal have 64 antennas each. We assume that there is a LOS channel between the mobile terminal and the base station. This can be considered a relatively favorable scenario for the 5G NR beam-search. We generate the channels according to 3GPP standard using Quadriga software [9.7].

The first metric used for simulations results is the *mis-detection probability*. When the user terminal enters the cell or is turned on, it looks to be associated to a certain BS. If the signal received from a BS is above a threshold the receiver can successfully associate with the BS. As such the mis-detection probability means that the post-combining signal was not above the threshold. We provide the mis-detection probability results in Figure 9.6. We can see that the mis-

detection probability of the proposed position-aided approach is sometimes better than the exhaustive search (saving a lot on the training overhead at the same time). The mis-detection probability of the proposed radar-aided approach is also comparable to the 5G NR exhaustive search protocol.

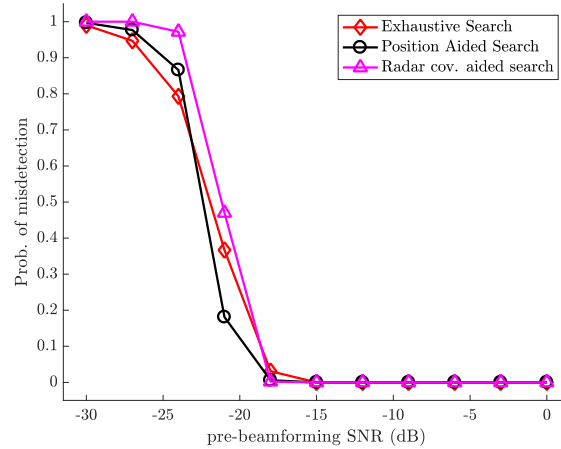


Figure 9.6: Misdetection probability of the exhaustive search, position-aided search, and radar-aided covariance estimation search

The second metric used to compare the proposed approach is the spectral efficiency. The spectral efficiency is defined as

$$SE = \mu\beta E[\log_2(1 + \text{SNR})],$$

where $\mu = (1 - \text{misdetection probability})$, which incorporates the fact that some devices are not detected in the spectral efficiency, $\beta = (1 - \frac{\text{training overhead}}{\text{channel coherence time}})$, is the fraction of the time left for data transmission. Note that the training time for the exhaustive search is 1.28 seconds, while for the position-aided approach it is 40 ms, and for the radar covariance-aided approach it is 20 ms. The results are plotted in Figure 9.7, for three channel coherence times: 10, 3, and 2 seconds. We can see from the results that the impact of varying coherence times is clear only in the exhaustive search, as it takes a long time for completion. For the other two strategies, the training time is very small in comparison with the channel coherence time and as such the three plots for the other two approaches coincide. Finally, the position-aided approach provides better spectral efficiency than the exhaustive search in all the tested cases. The radar-aided covariance approach is further better than the position-aided approach, justifying the proposed design.

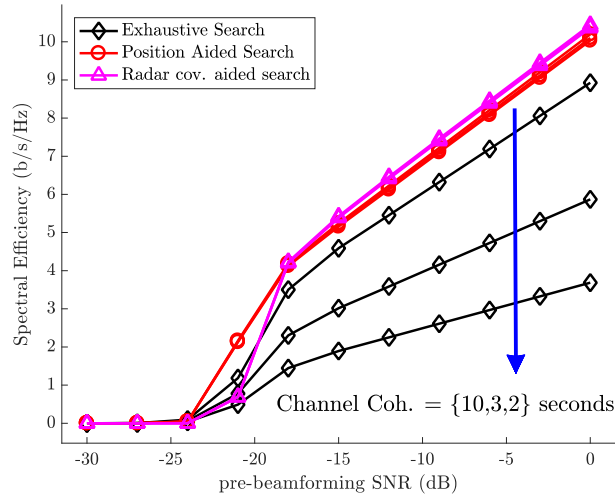


Figure 9.7: The spectral efficiency of the proposed radar-aided covariance approach, exhaustive search, and position-aided exhaustive search

9.3 Radar Equipment and Measurements

9.3.1 Equipment Purchases

The research team has a software-defined mmWave radar test-set up, which consists of several pieces of equipment obtained from National Instruments, as described in Table 9.1. The radar target is simulated for the device-under-test (DUT) (e.g., Delphi radar) using vector signal transceiver (VST) and mmWave transceiver, as shown in Figure 9.8. First, the DUT transmit signal is received, down-converted, and digitized. Then, the digitized DUT signal is modified based on the input target parameters (e.g., radar cross section, range, and velocity) received from the controller PXIe-8133. Finally, the modified up-converted and analog DUT signal is transmitted from the mmWave transceiver, which is later received by the DUT. The target height and rotation angle are, however, simulated using a DUT position control device, which consists of two Ethernet integrated steppers, one electronic clinometer, and one tripod, as described in Table 9.1.

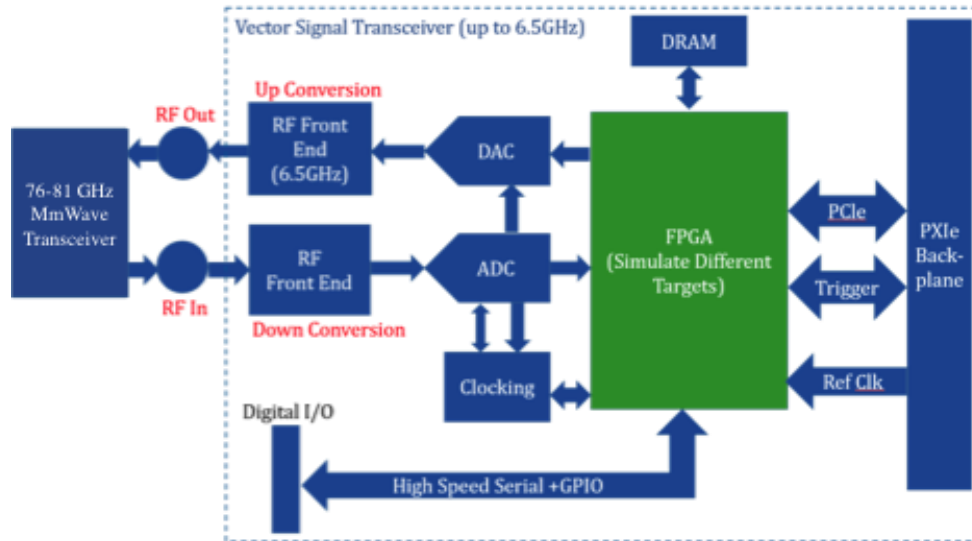


Figure 9.8: MmWave radar target simulator architecture

Table 9.1: National Instruments (NI) Test Equipment for mmWave Radar Prototyping

Model	Equipment	#	Capabilities
PXIe-1078	9-Slot 3U PXI Express Chassis with AC	1	It consists of 5 hybrid slots and 3 PXI Express slots with up to 250 MB/s per-slot bandwidth and 1.75 GB/s system bandwidth. It has a 4 slot-wide system controller slot, which can accept either an embedded controller or remote controller, and eight peripheral slots.
PXIe-8133	1.73 GHz Quad-Core PXI Express Controller	1	It is a high-performance Intel Core i7-820QM processor-based embedded controller with the 1.73 GHz base frequency, 3.06 GHz (single-core Turbo Boost) quad-core processor, and dual-channel 1333 MHz DDR3 memory. This processor delivers up to 2 GB/s of dedicated slot bandwidth, with the overall system bandwidth of up to 8 GB/s.
PXIe-5646R	6 GHz Vector Signal Transceiver (VST)	2	It features the flexibility of a software defined radio architecture by combining a vector signal generator and vector signal analyzer with FPGA-based real-time signal processing and control.
VRTT-77G-M1	77 GHz Vehicle Radar Test Transceiver	1	It is a phase-coherent millimeter-wave transceiver with 76-81 GHz frequency range and 3 GHz instantaneous bandwidth. It has extremely low noise performance. The equipment contains two horn antennas for transmit and receive.
ISM 7411E	Ethernet Integrated Stepper	2	It combines a stepper motor and a sophisticated stepper drive in a single device.
DUT Position Control	Custom made device under test (DUT) position control	1	It is a basic DUT position control device, which is used to set the rotation angle and height of the DUT, such as a Delphi radar. It consists of two ISM 7411E, one AccuStar electronic clinometer, and one Induro CLT203 Classic Series 2 Stealth Carbon Fiber Tripod

To aid in our real-world testing, the research team purchased Radarbook and Radarlog [9.8, 9.9], which are a state-of-the-art radar evaluation platform and a state-of-the-art data-capturing environment, respectively. The specifications of Radarbook and Radarlog are as follows:

- Radarbook (INRAS Radarbook n.d.) is a radar evaluation platform for rapid prototyping at 77 GHz band. Its baseband evaluation board contains eight analog channels, Altera Cyclon III FPGA, and USB module. Its 77 GHz frontend is a patch antenna array with four transmitter and eight receiver antennas that enables MIMO radar processing. Software support is provided for basic functionality to control front end using MATLAB along with Python GUI for raw data processing such as beamforming.
- Radarlog is a radar data-capturing environment for complex data analysis and algorithm design with a fully configurable processing chain in Python and Matlab. Using Radarlog, as many as sixteen analog receiver channels can be sampled at 65 MSPS per channel, processed streamed, and stored via a USB 3.0 interface with a data rate up to 1.6 GBit/s. Additionally, Radarlog can be operated with a USB server which enables to stream data to a HDF5 compatible file and simultaneously visualize the sampled or processed data. Software support is also provided for basic functionality to control the baseboard and raw data processing. The 77 GHz front end of Radarlog with two transmitter and sixteen receiver channels enables MIMO radar processing for frequency modulated continuous-wave (FMCW) waveforms.

Radarbook and Radarlog (shown in Figure 9.9) are compatible with an existing 77 GHz NI radar target emulator already in the research lab. The Radarbook and Radarlog were implemented for further use. This equipment enables prototyping for a radar-aided mmWave communication system. Radarlog, for example, can create range-velocity map information of multiple targets operating in real-world testing scenarios, which can be exploited to help mmWave communication link configuration. Also, Radarbook, which uses RF-frontends at 77 GHz, supports different scenarios needed to investigate the performance of a radar system in different types of applications such as traffic monitoring, vehicle detection, and drone detection. These products are used to test joint communication and radar in Section 9.3.2, and can be used for other mmWave prototypes in the future.

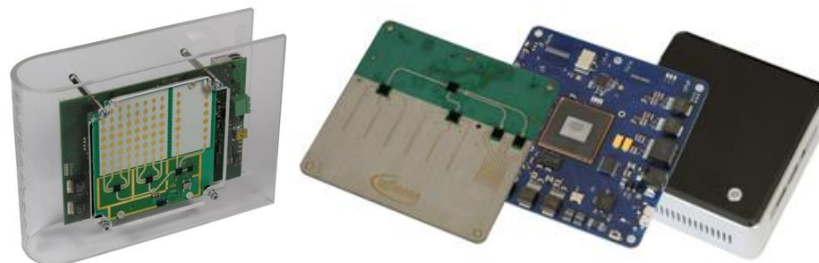


Figure 9.9: Radarbook (left) and Radarlog (right)

9.3.2 Radar-communication Congruence Measurements

The objective of measurements is to observe the similarity in the radar information and the communication channel. Specifically, for precoding at the BS, the power azimuth spectrum of the radar and communication channel needs to be similar for radar information to be useful for the communication system. We had established the radar-communication congruence using ray-tracing in an earlier section. Here, we demonstrate that the congruence can be observed in real world measurements.

We mimic the V2I scenario where a RSU is equipped with communication systems' BS and the radar. The car acts as a receiver for the communication signal and as a reflector for the radar signal. The setup we have considered is shown in Figure 9.10. The transmitter (and radar) and the receiver (and the reflector) are mounted on variable height tripods. The tripod itself is not a strong reflector in comparison with background and other reflectors in the scenarios. As such, we place a corner reflector on the tripod to have a strong reflection for the radar signal. We use the Radarbook to make the radar measurements. For the communication channel, we transmit using one Radarbook (acting as a RSU and is mounted on one tripod) and receive using the other Radarbook (that acts as a vehicle). The Radarbook is set to have a center frequency of 76 GHz for the communication channels' measurements and 77 GHz for the radar measurements. With this, the frequency differences between the communication and the radar band are taken into account.

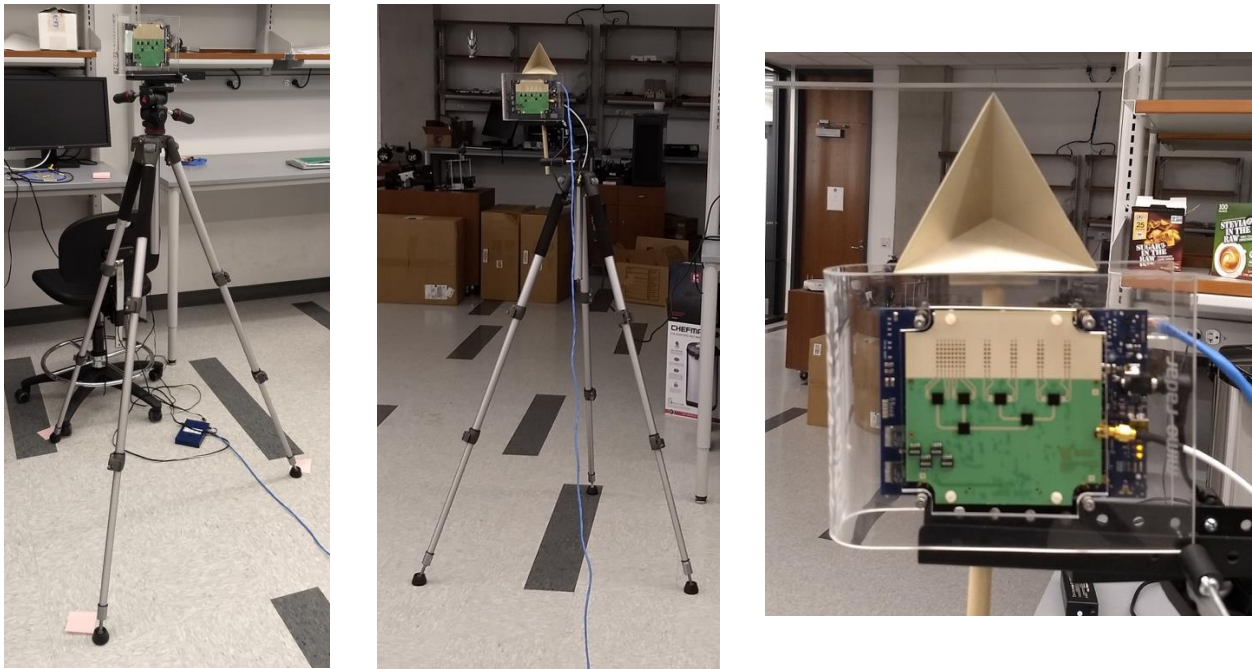


Figure 9.10: The RSU (left), vehicle (center), and zoomed in corner reflector and Radarbook on the vehicle (right)

It is not straightforward to have real-time measurements as the tripod is moving, but to show the congruence we make sets of measurements at different locations and observe the

variation in the radar and communication channels' power azimuth spectrum. The four locations of the tripod are shown in Figure 9.11. Before discussing the results for the communication and radar congruence, we would like to highlight two signal processing strategies that are used to obtain the result.



Figure 9.11: The radar-communication congruence measurements completed at four different locations on the vehicle (i.e., receiving tripod)

Background subtraction: There is a strong background in the radar observations. This background is static, and is part of all the observations. As such, this background can be estimated easily by taking the mean of all the radar measurements. In V2I applications, the background can be obtained easily by taking measurements without traffic (e.g., late-night or other no activity periods). As such the knowledge of background can be assumed. Once the background is available, it is prudent to take out the background before observing the congruence between the radar and communication channels.

Spatial spectrum estimation: The Radarbook has 4 transmit antennas and 8 receive antennas. The spacing of the transmit and receive antennas is such that the Radarbook acts as a SIMO radar with 29 antenna elements. The spatial spectral estimation with 29 antennas is easy and Fourier processing suffices to get a reasonable spatial spectrum estimate. The communication channels' measurement at the receiver using Radarbook, however, are obtained using 8 antennas. Simple Fourier processing does not give a good estimate of the spatial spectrum. As such, to obtain the communication channels' spatial spectrum, we use MUSIC algorithm [9.10].

The results for this spatial congruence experiment are shown in Figure 9.12. We can see from the result that as the receiving tripod (used as a surrogate for the vehicle) is moved, both the radar and communication spatial spectrum change. There, however, is always an agreement between the radar and communication spatial spectrum. This implies that the radar and communication spatial information is consistent, and radar information can be used to configure the mmWave communication links.

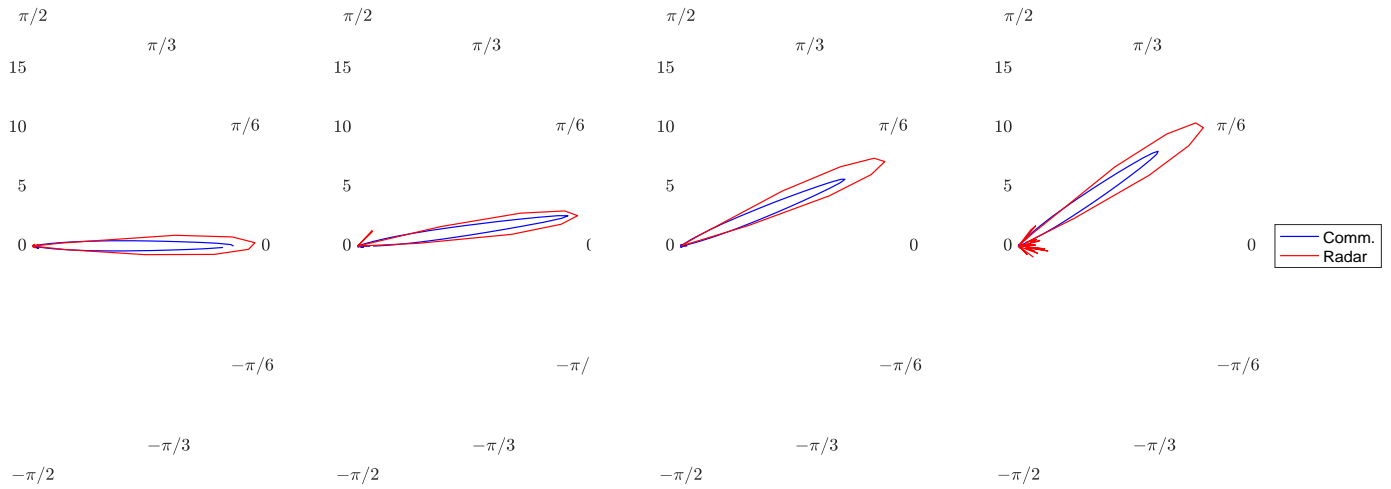


Figure 9.12: The radar and communication spatial spectrum

9.4 Summary

In this chapter, first, we established the BS-mounted radar can provide better positioning information in comparison with traditional GNSS-based positioning. This position information can reduce the training overhead of grid-of-beams beam-training by over 90%. Then, we established the spatial congruence of radar and communication using ray-tracing, and used radar covariance to obtain the mmWave communications' precoders/combiners. We then outlined modifications to the initial access procedure of the 5G NR to highlight that radar covariance-based eigenvalue precoding has misdetection probability similar to exhaustive search with a much lower training overhead.

We outlined the purchased equipment for prototyping and measurements, including Radarbook and Radarlog. Using the equipment, we performed measurements to establish the congruence of radar and communication spatial information.

Chapter 10 Advanced Techniques to Further Enhance Joint mmWave Radar and Communication System

10.1.1 Introduction

In this project, we developed an automotive joint communication-radar (JCR) system at the mmWave band. Millimeter-wave (mmWave) spectrum is a solution to realize high data rate communication and high resolution radar sensing. An mmWave joint communication-radar (JCR) system that leverages the same waveform for hardware reuse will enable significant advantages in terms of cost, size, power consumption, spectrum usage, and adoption of communication-capable vehicles. Traditional mmWave radars for consumer applications employ heavy analog pre-processing due to the use of low-speed analog-to-digital converters (ADC) and frequency-modulated continuous-wave (FMCW) technology [10.1]. An mmWave JCR system with fully digital time-domain processing can now be realized by leveraging mmWave communications hardware with high-speed ADCs and phase-coded waveforms. The mmWave JCR system, however, incurs trade-off between communication data rate and radar detection/estimation performance.

Prior mmWave JCR waveforms that achieved high data rate were based on the IEEE 802.11ad single-carrier physical layer (SC PHY) modulation frames [10.2]. Although the waveform proposed in [10.2] simultaneously achieved a cm-level range resolution and a Gbps data rate by exploiting the preamble of a single frame for radar, the velocity and angular estimation performance as well as the radar range of operation was limited.

In this work, we developed several alternative techniques to enhance communication and radar system performance. In particular, we proposed novel waveform designs, beamforming algorithms, processing techniques, and a MIMO JCR architecture with low-resolution ADCs. To quantify the trade-off between radar and communication, we proposed both a rate-based metric and a minimum mean square-based metric. Additionally, we constructed different optimization problems for the JCR system design in different vehicular scenarios. The simulation results demonstrate that the alternative techniques achieved a significant improvement in the radar estimation performance with only a small degradation in communication data rate.

Additionally, we developed an mmWave proof-of-concept test bed to demonstrate and evaluate the performance of our proposed joint communication and radar system at 71-76 GHz band with 2 GHz bandwidth. Our single-input-single-output software-defined JCR test bed leverages recently developed National Instruments (NI) mmWave communication platform that allows fully programmable interface. We also carry out indoor and outdoor measurements in different vehicle scenarios. The results suggest that mmWave communication system with 2 GHz bandwidth can achieve centimeter (cm)/ decimeter (dm)-level range accuracy using mmWave JCR test bed, which matches our theoretical analysis.

This chapter is organized as follows: First, we summarize the modifications in the IEEE 802.11ad-based radar as well as novel waveforms and signal processing algorithms for achieving

improved mmWave JCR performance. Then, we describe our mmWave test bed for the proposed joint communication and radar system at 71-76 GHz band. Lastly, we present our measurement results that demonstrates the validity of our theoretical results.

10.1.2 Modify IEEE 802.11ad for Better Radar and Communication Performance

In this project, we explored advanced processing and system design techniques to further optimize the performance of the IEEE 802.11ad-based radar developed in Phase 1. In particular, we explore alternative estimation techniques, waveform structure, and beamforming algorithms for improved target estimation performance and communication rate.

Alternative Processing Techniques

For achieving high range and velocity resolution in a multi-target scenario, we propose transmitting a super-frame with multiple fixed length frames in a coherent processing interval (CPI). Then, we the channel estimate can be extracted from the radar received signal by leveraging the conventional WLAN processing techniques. Lastly, we implement classic pulse-Doppler based radar processing algorithm on the obtained channel for enhanced velocity estimation in a multi-target scenario [10.3].

To demonstrate range and velocity resolutions of the joint system, we consider an example scenario with two scattering centers (either representing two different vehicles or belonging to the same vehicle) in different CPI duration. In this scenario, one of the target vehicles is a recipient vehicle, vehicle R. The second target vehicle, vehicle T, is considered within the beamwidth of the source vehicle and is separated in range, relative velocity, and angle-of-arrival (AoA)/angle-of-departure (AoD) as compared with the recipient vehicle by $\delta\rho$ and δv , and $(\delta\phi, \delta\theta)$ respectively, as shown in Figure 10.1. For vehicle R, we choose range as 14.31 m and velocity as 30 m/s, which falls in the typical operating span of LRR range and velocity specifications. We consider that the AoA/AoD corresponding to the vehicle R is $(90^\circ, 90^\circ)$, which is likely to happen when the vehicle R is in the same lane as the source vehicle for applications such as cruise control. For vehicle T, we consider $\delta\rho = 4.26$ m, $\delta v = 30$ m/s, and $(\delta\phi, \delta\theta) = (100^\circ, 90^\circ)$, which also falls in the typical span of automotive long-range radar specifications [10.4].

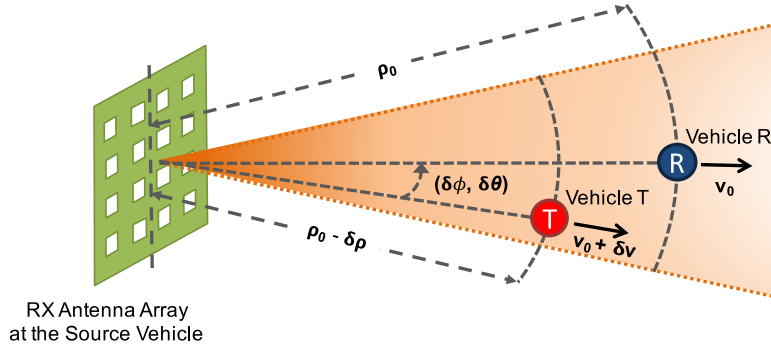


Figure 10.1: A multi-target scenario with two vehicles, namely vehicle R and vehicle T, within the mainlobe of the TX beam at the source vehicle. Both the vehicles are slightly separated in range, relative velocity, and direction with respect to the source vehicle.

To illustrate the high-resolution performance as well as the extension of multi-target processing to the extended target scenario, we consider the scenario with two scattering centers corresponding to the recipient vehicle with 30 m/s velocity and $(90^\circ, 90^\circ)$ AoA/AoD in a 4.2 ms CPI. We assume that the extended target model is 3-range cell wide with the first scattering center at 14.31 cm, i.e., 168th cell and the second at 170th cell. We assume that the energy is distributed uniformly among the two scattering centers. Figure 10.2 shows the 3D mesh plot of the estimated delay-Doppler map for the second scenario with 100 times interpolated discrete-Doppler bins. We infer that the range resolution is 8.52 cm, which is the same as the first scenario, whereas the velocity resolution is 0.59 m/s with a 4.2 ms CPI. This example demonstrates that we can achieve high range resolution of less than 0.1 m and velocity resolution of less than 0.6 m/s in the multi-target scenario for a 4.2 ms CPI, as desired in the automotive long-range radar applications [10.4].

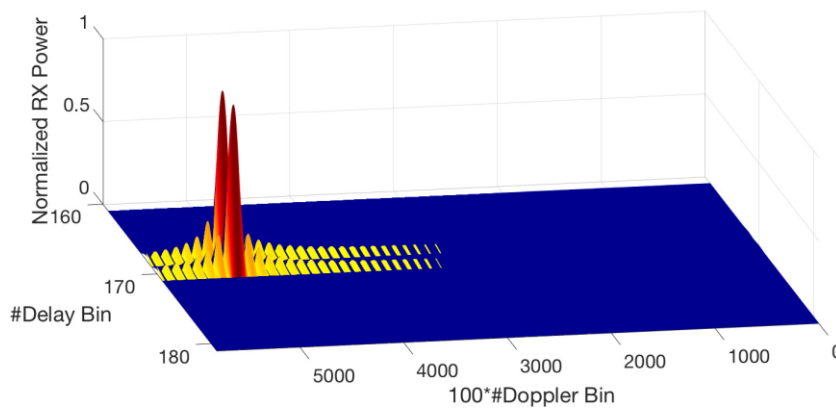


Figure 10.2: Mesh plot of the estimated 3D plot of the normalized-delay Doppler map shows that there are two dominant reflections from the recipient vehicle present at the 50th Doppler bin. Due to the fifty-eight times narrower mainlobe width in the Doppler domain as compared to the first scenario, the velocity resolution is 0.59 m/s in this simulation.

Adaptive Waveform Structure

To enhance range and velocity estimation performance, we investigated an adaptive preamble design that permits a trade-off between target estimation accuracy and communication rate. In particular, we optimized the preamble duration to adapt the waveform, as shown in Figure 10.3. The optimization is based on the weighted average sum of the radar Cramer Rao Bound (CRB) metric for range/velocity and a novel effective communication minimum mean square metric (MMSE) [10.5].

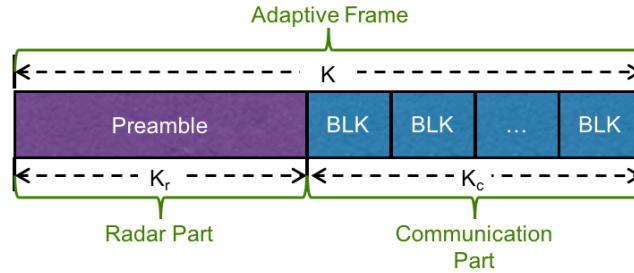


Figure 10.3: Frame structure of an adaptive waveform design with variable preamble duration

The simulation results in Figure 10.4 demonstrate that with an increase in the separation between the target and the source vehicles, the tradeoff between radar and communication gets tightened and the range/velocity CRB bounds per symbol get degraded more severely than the communication MMSE per symbol. The optimum for different weightings were explored, while maintaining Gbps communication data rate and cm-level range accuracy. At a vehicles separation distance of 270 m, the adaptive preamble design resulted in an improvement of the range MMSE by 3.2 cm^2 , while decreasing the spectral efficiency by 1.4 bits/s/Hz as compared to the IEEE 802.11ad preamble. At 5 m distance, the spectral efficiency increased by 1.4 bits/s/Hz, while degrading the range MMSE by a factor of 1.4 dB and the velocity MMSE by 4.2 dB as compared to the IEEE 802.11ad preamble. This work can be extended to a large number of interesting time-domain duplex frameworks for joint radar and communication.

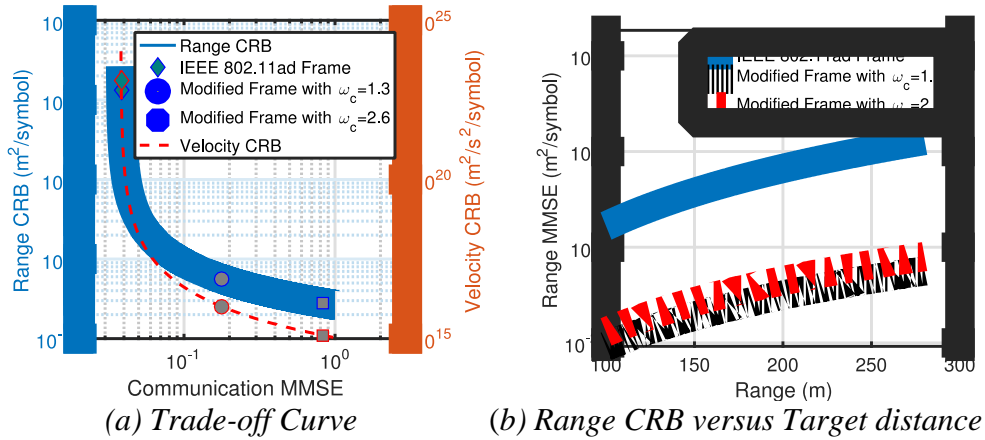


Figure 10.4: Trade-off between radar CRB and communication MMSE with preamble duration

Adaptive Beamforming Algorithm

The IEEE 802.11ad millimeter-wave Wi-Fi waveform can be simultaneously used for high range/velocity resolution radar and Gbps data rate communication in vehicular applications. The angular field of view (FoV) for radar sensing, however, is limited by the employed directional analog beam. We proposed a sparsity-aware beamforming design that permits a trade-off between communication rate and radar detection/estimation performance in the angular domain. The proposed design uses random transmit antenna subsets to form a coherent beam towards a communication receiver while simultaneously perturbing the sidelobes of the resulting beam pattern. Sidelobe perturbation results in random grating lobes, which we exploit for compressed radar sensing, as shown in Figure 10.5. The system performance trade-off involved in the adaptive beamforming design is evaluated using a novel joint communication-radar rate metric [10.6].

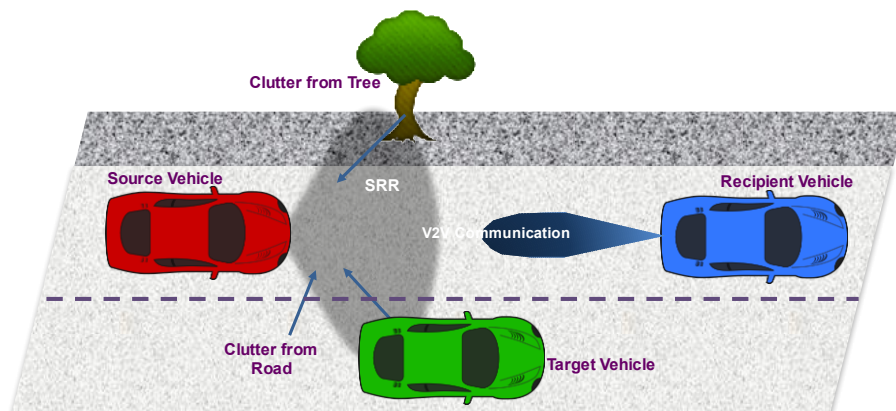


Figure 10.5: An illustration of an IEEE 802.11ad-based joint system of SRR with a wide FoV and V2V communication with a narrow FoV

Numerical results in Figure 10.6 and Figure 10.7 for a multi-target radar channel and a frequency-flat communication channel demonstrate that subset size optimization enables the detection of short-range radar targets with high probability and angular resolution in a wide FoV at the expense of a slight hit in the communication data rate.

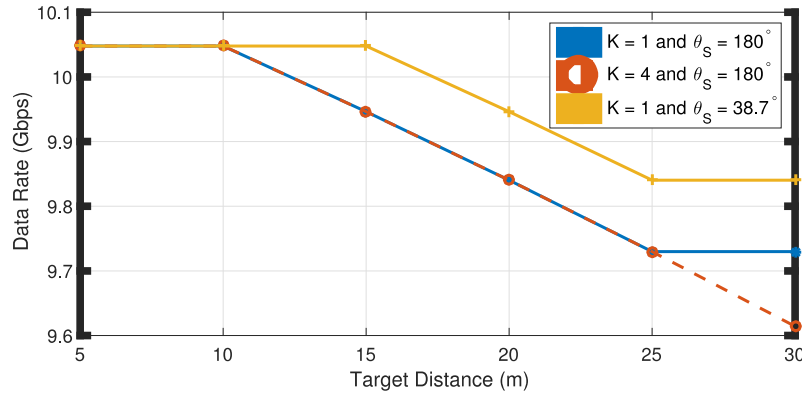


Figure 10.6: Optimized communication rate decreases with target distance

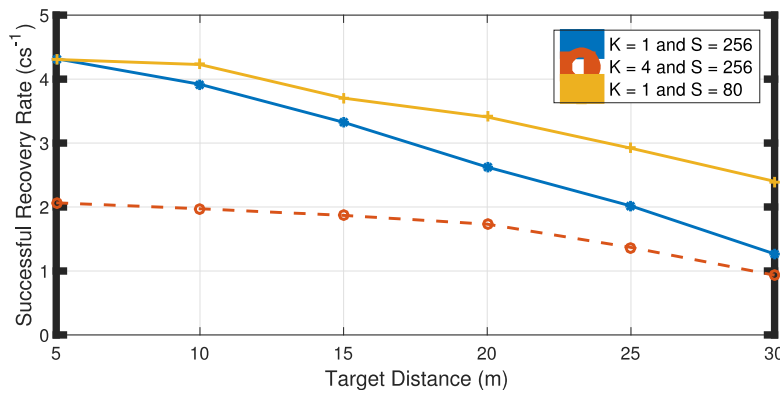


Figure 10.7: Optimized radar recovery rate decreases with target distance

10.1.3 Designing New Joint Waveforms and Signal Processing Algorithms

In this section, we consider the use of new JCR waveforms and multiple-input-multiple-output radar system with low-resolution analog-to-digital converters (ADCs) to enable fully digital baseband processing with high sensing resolution.

Virtual Pulse Design for mmWave WiFi-Based Joint Communication-Radar

Although [10.2] simultaneously achieved a cm-level range resolution and a Gbps data rate by exploiting the preamble of a single frame for radar, the velocity estimation performance was limited. In this section, we explore a virtual pulse design approach for an adaptive joint communication-radar system in a multi-target scenario. In this approach, the frame lengths are

varied such that their preambles, which are exploited as radar pulses, are placed in a coprime fashion. A few non-uniformly placed pulses in a CPI are then used to construct a virtual block with several pulses, leveraging the sparsity inherent in the mmWave channel. This virtually increases the radar pulse integration time and enables an enhanced velocity estimation performance, a more flexible waveform design, and a relaxed trade-off with the communication rate as compared to [10.3].

The frames can be placed either with a constant distance between them, as shown in Figure 10.8 or with varying distance, as shown in Figure 10.9. In either case, the location of the m th frame is assumed to be placed at integer multiple of the Doppler Nyquist sampling interval. Both the pulse approaches use a fixed IEEE 802.11ad preamble with 3328 symbols. For the uniform pulse approach in [10.3], the number of symbols per frame is constant and meets the Nyquist criterion, while for the virtual pulse approach, the frame spacing is varying and chosen in a sub-Nyquist. The virtual pulse approach is conceptually similar to the concepts of staggered pulse repetition intervals (PRI) used in the classical long range radar and sparse sampling/arrays used in the under-sampled frequency/angle/channel estimation. For tractable analysis, we specifically use here the deterministic waveforms that can be represented in closed-form and contain no holes in their corresponding difference co-waveforms, for optimally selecting the locations and the number of frames in a given CPI. Among several redundancy waveforms with no holes [10.7], Wichmann and nested waveforms are especially relevant and interesting to consider.

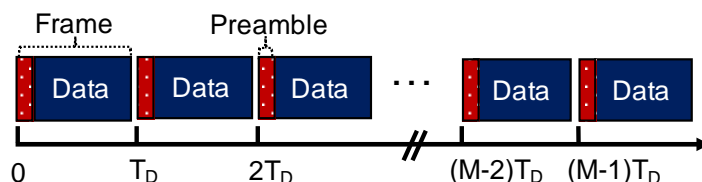


Figure 10.8: Uniform pulse approach, where a CPI consists of M equi-spaced frames placed at Nyquist rate. Here, each frame contains fixed preamble and data lengths.

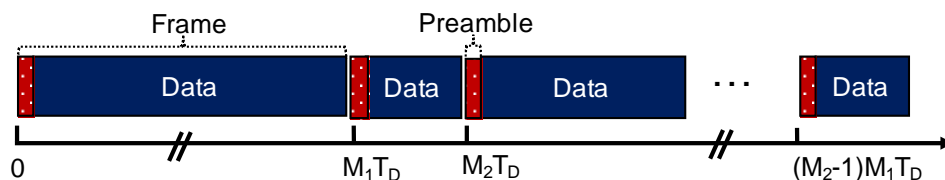


Figure 10.9: Virtual pulse approach, where a CPI consists of non-uniformly placed $M = M_1 + M_2$ frames

The radar performance for the uniform waveform and the non-uniform waveform is evaluated based on the CRB for the target velocity estimation. The communication performance for the joint system is evaluated using a novel distortion MMSE (DMMSE) metric that relates one-to-one with spectral efficiency. To evaluate the achievability of radar CRB, we consider MUSIC-

type algorithms that generally achieves higher estimation accuracy than traditional discrete Fourier transform (DFT)-based radar processing algorithms.

The JCR performance optimization is a multi-objective problem of simultaneously optimizing both the radar performance, and communication performance. Using the scalarization approach known to achieve a Pareto optimal point for multiple objectives, if they are convex, the joint optimization can be formulated as the weighted average of the two metrics. The positive normalizing and weighting factors assign the priorities for radar and communication tasks, which can be adjusted adaptively to meet the requirements imposed by different vehicular scenarios. For example, the weights can be assigned to ensure proportional fairness between two objectives. Alternatively, optimization can be modified as minimization of one of the objectives with second as a constraint that would guarantee an acceptable performance for one of the tasks. In our work, we have optimized waveforms for all three problem formulations.

The simulation results show that the optimal virtual waveforms achieve a significant improvement in the radar CRB with only a small degradation in the communication DMMSE, as shown in Figures 10.10 and 10.11. For a weighted average optimization with a high number of targets, both the optimal radar CRB and the optimal communication DMMSE are minimized. For a radar CRB constrained optimization, the optimal radar range of operation and the optimal communication DMMSE are improved. For a communication DMMSE constrained optimization with a high DMMSE constraint, the optimal radar CRB is enhanced. Comparison of MMSE-based optimization with traditional degrees of freedom (DoF)-based optimization indicated that a DoF-based solution converges to a MMSE-based one only for a small number of targets and a high signal-to-noise ratio.

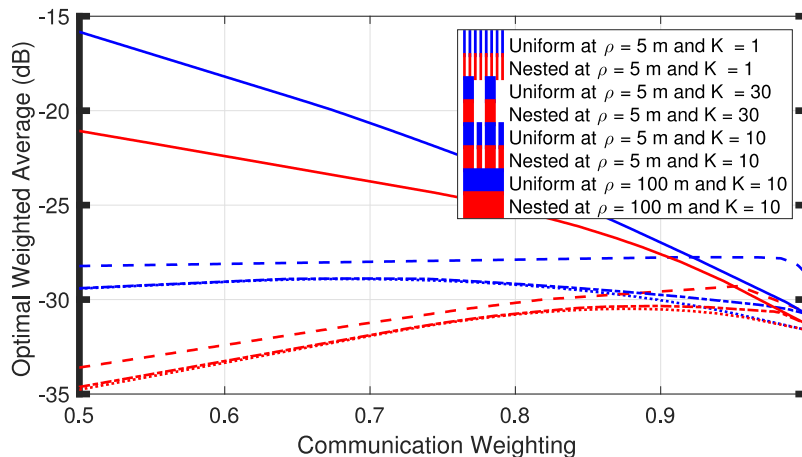


Figure 10.10: Optimized weighted average for different normalized communication weightings

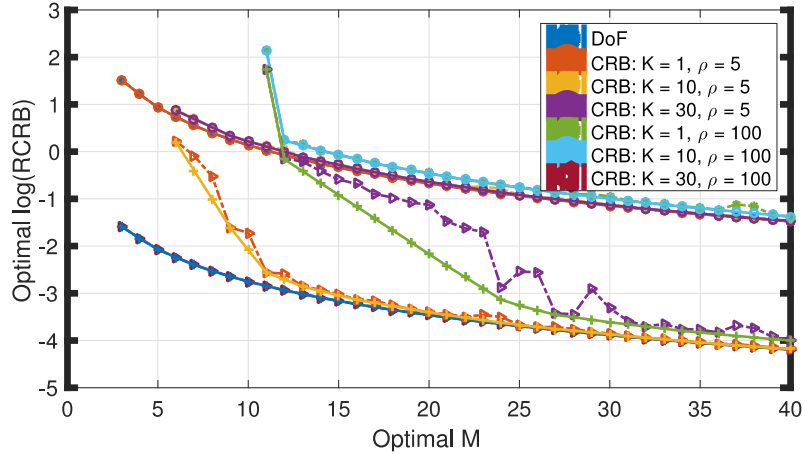


Figure 10.11: Comparison between DoF-based optimization and CRB-based optimization for nested waveform

Low Resolution Sampling for Joint Millimeter-Wave MIMO Communication-Radar

In this work, we proposed the deployment of millimeter-wave MIMO for joint vehicular high speed communication and high resolution radar sensing. To cope with the significant hardware complexity, we considered the use of low resolution ADCs while maintaining a separate radio-frequency chain per antenna, as illustrated in Figure 10.12.

The system performance is analyzed in terms of Cramer Rao lower bound and achievable data rate, and compared to the ideal case with infinite resolution ADCs. Additionally, we study the impact of quantization on the trade-off between these performance metrics.

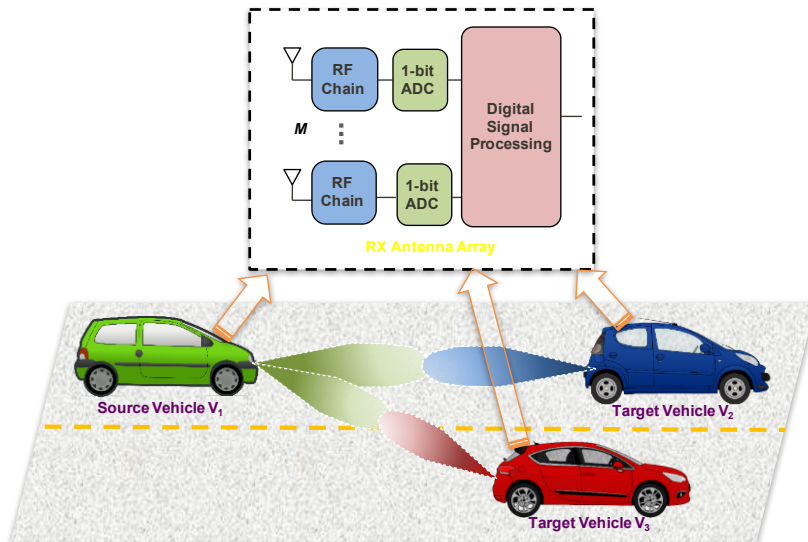


Figure 10.12: The source vehicle sends a waveform to the target communication receiver and use their echoes to estimate their locations. The TX front end (not shown) includes a single directional antenna with a notch in the direction of its receiver.

Numerical results in Figure 10.13 for a single-target scenario demonstrated that proposed mmWave MIMO joint communication-radar system with 1-bit ADC performs closely to the ideal case with infinite resolution. The loss of information due to 1-bit ADC impacted communication performance more than radar for a given vehicle separation distance. This work can be extended to a multiuser MIMO JCR system at the mmWave band.

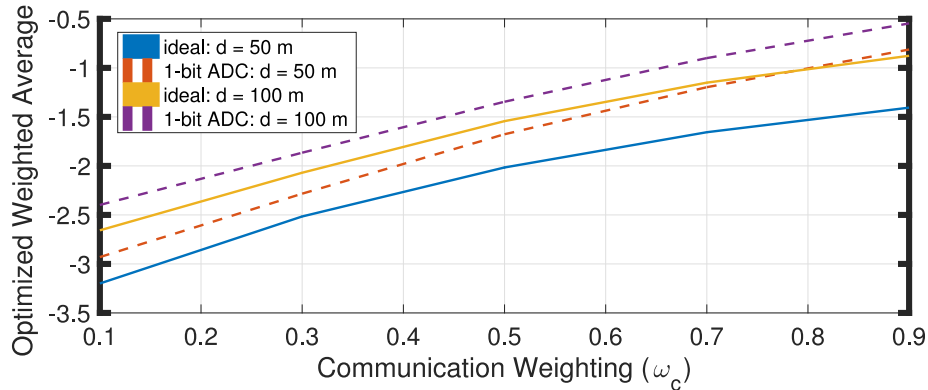


Figure 10.13: Optimized weighted average for different communication weightings

10.1.4 Prototyping a Combined Communication and Radar Car-to-car Link

The research team developed a software-defined mmWave tested to demonstrate and evaluate the performance of our proposed joint communication and radar system at 71-76 GHz band with 2 GHz bandwidth. This band is very close to the automotive radar band at 76-81 GHz. Due to the hardware limitations prototyping at mmWave have been difficult, especially for communications applications. Our proof-of-concept test bed for joint communication and radar leverages recently developed NI mmWave communication platform [10.8]. The NI platform supports fully programmable interface and is based on innovative compiling software on a field-programmable gate array (FPGA) hardware platform that is easy to use as compared to the conventional hardware description language.

Figures 10.14 and 10.15 illustrate our developed mmWave joint communication-radar test bed in a single-input-single-output (SISO) set-up with horn antennas. We consider the use case where a source vehicle transmits signal to a recipient vehicle receiver and uses the echoes from target vehicles and clutter to derive target range and velocity estimates at the source vehicle receiver. For this purpose, we use two chassis, one as the source vehicle that consists of communication transmitter and radar receiver and the other for the recipient vehicle that contains the communication receiver. The mmWave test bed consists of two chassis, housing the NI PXIe 7902 for baseband transmit (TX)/receive (RX) processing, NI PXIe 3630 (IF and LO unit) with 10.5–12 GHz IF frequency, ADC/DAC adapter modules, NI PXIe 3610 (14 bit DAC) and 3630 (12-bit ADC) with sample rate 3.072 GS/s and 2-GHz bandwidth. Each chassis is connected to a

mmWave transceiver RF front-ends and horn antennas. The chassis can be connected using a Rubidium clock for time and frequency synchronization of the joint communication-radar system.

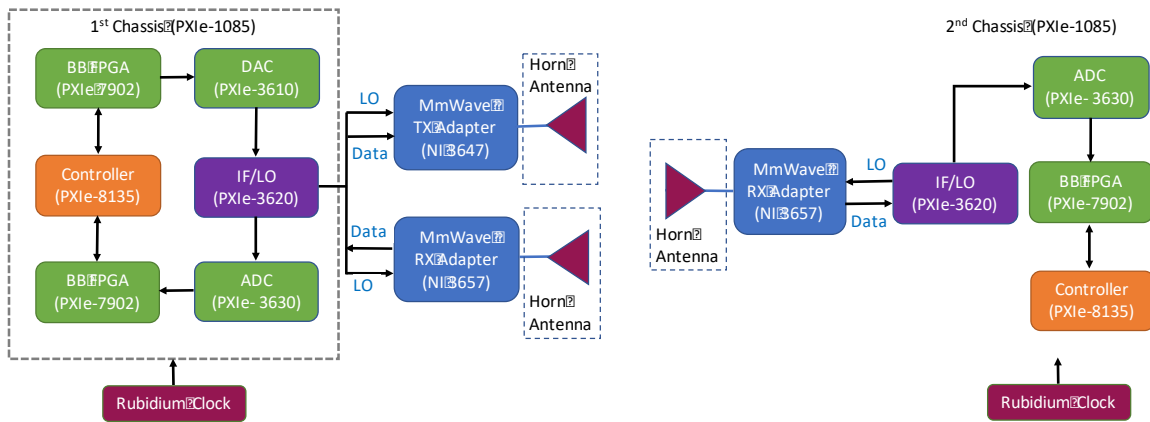


Figure 10.14: Proposed mmWave test bed set-up for joint communication and radar prototyping.

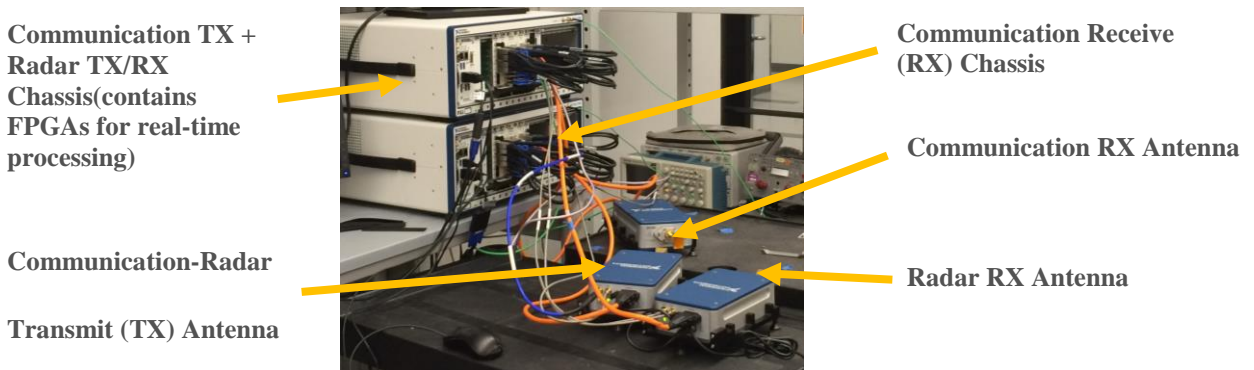


Figure 10.15: Joint radar and communication test bed with horn antennas.

Chapter 11 Data Collection and Analysis Methodology

The performance of ADAS can be influenced by a wide range of variables, including the characteristics of the interacting vehicles, traffic infrastructure, built environment and urban design, weather conditions, human factors (for example, perception and reaction time, in the case of non-autonomous systems), and situational operating conditions (traffic composition, unexpected obstructions, and relative position of vehicles). In the case of intersections, there are different geometric roadway configurations with different approach gradients, lane configurations, lane widths, traffic control types, traffic approach volumes, divided versus undivided approaches, and different turning movements. In the context of passing maneuvers, factors such as vehicle types and associated kinematics, geometric configurations of roadway system and interplaying sight distances, operating speed, and driver behavior come into play. For operation of ADAS in areas with a significant proportion of non-motorized users, the pertinent factors include the number of pedestrians and bicyclists, the volume of motorized traffic flow, signal configurations, geometry of infrastructure, and the extent of interaction between the motorized and non-motorized flow. Because the number of possible combinations of variables in each scenario group can increase very rapidly, identifying the effect of each variable on the performance of the ADAS can be extremely challenging. Therefore, field data collection to test the performance of the proposed ADAS under different traffic scenarios must follow an experimental design approach.

This chapter describes the general data collection methodology developed to (1) analyze and quantify the impact of different road geometry, built-environment, and traffic variables on the performance of each equipment, and (2) understand to what extent different sensors and communication equipment can complement each other under different traffic scenarios. In the remainder of this chapter, we provide details of equipment and vehicles utilized for data collection. Then, we discuss the system performance measures that are adopted as the outcome variables of the experimental design and we present the general procedure used to obtain the primary and secondary factors (explanatory variables) for all experiments. Further details of each specific scenario are provided in the following chapters.

11.1 Equipment

The equipment-suite installed on the vehicles for field data collection comprises sensing equipment, communication equipment, auxiliary equipment, and high-performance laptops. The laptops act as the backbone for controlling the operation of different equipment and, in conjunction with Python-based logging framework, store the feeds from different sensing and communication equipment in data structures conducive to future processing and fusion. The sensing equipment includes on-board camera, radar, and LIDAR (light detection and ranging). The communication equipment comprises a DSRC (dedicated short-range communication) unit capable of transmitting and receiving messages from other DSRC-equipped vehicles (mmWave communication units may be incorporated for future data collection). The vehicles are fitted with two auxiliary instruments

on-board: an inertial measurement unit (IMU) and a high precision GPS (antenna and anchor units). While the IMU provides movement information to help correct readings from different equipment, the high-precision GPS units record the exact locations (ground truth) of the vehicles during the maneuvers that form the benchmark to evaluate the performance of prediction algorithms.

The sedan, SUV, and pickup truck used as prototype vehicles in this project are also used to collect data. The data feeds from different instruments are transmitted to on-board laptops and simultaneously stored in separate log files. For data collection where only two vehicles were needed, such as for intersection crossing, the pickup truck carried a stationary GNSS anchor for ground truth positioning. When all three vehicles were used, the stationary GNSS was kept on an elevated platform nearby.

11.1.1 Equipment Specifications

a) Radar: Delphi ESR

- Detection range: 174 meters from long-range radar, 60 meters from short-range radar
- Field of vision: 10 degrees from long-range radar, 45 degrees from short-range radar
- Frequency of information: once per 50 milliseconds

b) DSRC communication: Cohda Wireless DSRC Unit

- Communication range: Values vary depending on the implementation, but at least 100 meters is accepted as standard
- Communication field of vision: 360 degrees
- Frequency of information: once per 100 milliseconds

c) LIDAR: Quanergy M8

- Detection range: 200 meters
- Field of view: 360 degrees in horizontal domain, 20 degrees in vertical domain
- Frequency of information: once per 50 milliseconds

d) Camera: Logitech c920 USB

- Video resolution: 1920 x 1080
- Field of view: 110 degrees
- Frame rate: 30 frames per second

e) High precision GPS (PPP-GNSS) and IMU: EMLID Reach and MPU9520

f) In-vehicle laptops for data collection and real-time fusion: System76 Oryx Pro 2

The devices are installed as discussed in Section 2.1.

11.2 System Performance Measures

With traffic safety as the prime focus of the project, our research would ideally directly measure the performance of the ADAS in terms of safety improvements (reduction of crash rates, for example). However, we cannot test the operation of the system in actual, quasi-crash events. For this reason, the emphasis was on direct measures of performance that can be easily extracted from the data collected. These results were used later to derive indirect safety performance measures and statistics. Direct performance measures are based on the comparison between the trajectories (or positions) estimated by the system and the true trajectories of the vehicles. Since the experiments are held in real-world traffic environment where we cannot control for the true position of the vehicle, high-precision GPS equipment is used to capture true positions.

For the urban intersection and overtaking-maneuver scenarios, two outcome variables are tested: the difference between the predicted location and the true location of the oncoming vehicle, and whether a trajectory prediction was correct or incorrect. The first variable is continuous and will be modeled through linear models, while the second is a binary variable and will be modeled through discrete choice models. In both cases, non-additive relationships between the explanatory factors are tested by the inclusion of interaction terms. For the scenario with a significant proportion of non-motorized road users (pedestrians and bicyclists), the performance measures adopted are the difference between predicted location and true location of the non-motorized road users (this test should be held in a controlled environment where pedestrians' true positions are known) and the percentage of relevant objects (pedestrians and bicyclists) detected in a given situation (this variable can also be translated as the success/failure of detecting a given object). Again, linear and discrete models are used when appropriate (further details about the modeling framework will be presented separately for each group of scenarios).

11.3 Experimental Design

An experiment is a group of tests in which the input variables (also called factors) are changed based on a given rule in order to identify the reasons for the changes in the output response. Experiments can help determine which variables are most influential on the response and where to set the influential controllable variables so that the response is usually near the desired value [11.1]. In the case of testing ADAS in real-world traffic scenarios, the experimental design approach will serve to identify the situational factors and factor levels (number of different values a factor can assume according to its discretization) that most adversely affect the individual and joint performance of the sensing and communication equipment. Identifying factors that limit the performance of such equipment is crucial for enhancing the algorithms of ADAS to operate more effectively in critical scenarios. Additionally, the experimental design approach allows for estimated coefficients to be used to forecast the overall ADAS performance in untested scenarios,

facilitating a comprehensive analysis of the performance of the ADAS with parsimonious data collection.

Experiments are usually held in very controlled environments in which researchers have the ability to freely manipulate the combination of factors and their levels in order to reach research objectives. In this project, the performance of the ADAS is estimated using data collected in real-world transportation infrastructure, which include factors that cannot be controlled by researchers. In addition, the manner in which different factors are combined (for example, in an intersection, the number of lanes, horizontal alignment of approaches, and vertical gradient) is fixed and depends on the available infrastructure, limiting the flexibility of the research design. Therefore, the experimental design approach implemented in this project is an adaptation of traditional experimental design methods. Specifically, our approach clusters multiple small full factorial designs (with few factors and levels) and adds additional covariates to the analysis.

The full factorial design is probably the most common and intuitive strategy of experimental design: Under this method, samples are taken under every possible combination of the factors' values [11.2]. For example, in a full factorial design of three factors with three levels each, twenty-seven different combinations need to be measured. This design prevents the effect of each factor over the response variable from being confounded with the other factors. However, since the number of combinations grows rapidly as the number of factors and levels increases, full factorial designs may be infeasible when dealing with complex problems. Although literature on the design of experiments offers different solutions to reduce the number of observations required in studies with a large number of factors and levels (for example, fractional factorial designs and efficient designs), such methods are not easily applied to the current research. This is because these methods require the researcher to have the ability to select the exact combination of factor levels for each observation based on fixed rules. Since our data collection depends on existing traffic infrastructure, finding multiple locations that meet exact combinations of factor levels becomes nearly impossible when multiple factors are considered. A solution, therefore, is to identify critical factors and develop multiple full factorial designs based only in small groups of factors. Secondary factors, which cannot be controlled by the researchers, can then be added as additional covariates in the models. In other words, the proposed design contains six steps that should be performed for each of the scenario groups:

- 1) Identification of primary factors and factor levels (factors that are most likely to have a significant impact on the performance of the ADAS).
 - a. Identification of a comprehensive list of factors and factor levels relevant to each of the three groups of scenarios
 - b. Evaluation of the performance sensitivity of each equipment (sensors and communication equipment) to the different scenario factors based on a priori knowledge about the equipment
 - c. Selection of factors and factor levels that most critically affect the equipment performance

- 2) Identification of locations where each of the combinations between factors levels can be measured.
- 3) Development of samples for multiple full factorial designs with a few primary factors each.
- 4) Identification of secondary factors present in the selected locations.
- 5) Development of separate models for each small design.
- 6) Development of a final model that reunites all the small designs with primary and secondary factors.

In the following section, the detailed description of each of the steps above is presented for the urban intersection set of scenarios.

11.4 Overview of Modeling Approach

After data was collected based on the experimental design, models were estimated to quantify the impacts of variables that describe the driving environment (traffic characteristics, weather, and roadway geometry) and the vehicle's relative position to another vehicles and non-motorized road users on the performance of the ADAS. Results from these models were used for multiple purposes. First, they provided guidance for improvement in the prediction algorithms developed earlier in the project. Second, they determined which variable (and under what environmental contexts) most affects the performance of each sensing and communication equipment component. Finally, the models were used to assess the overall gains (in terms of prediction quality) of combining communication and sensing for ADAS purposes.

For the urban intersection and overtaking maneuver sets of scenarios, we adopted two types of measures of system performance: the absolute error of the prediction (a continuous variable measured in meters as the difference between the predicted location of the vehicle and the true location indicated by the high precision GPS system), and whether a prediction was successful (correct) or not (a binary variable). For the scenario with significant proportion of non-motorized road users (pedestrians and bicyclists), the performance measures adopted are the difference between predicted location and true location of the non-motorized road users (this test should be held in a controlled environment where the pedestrians' true positions are known) and the percentage of relevant objects (pedestrians and bicyclists) detected in a given situation (this variable can also be translated as the success/failure of detecting a given object). The reason for adopting two prediction variable variants (both a continuous as well as a binary measure indicating success/failure) is to test different modeling approaches and then to identify which one allows for the extraction of more information. While the linear regression model with a continuous dependent variable offers the advantage of easy interpretability of coefficients, it is highly sensitive to outliers (which are frequent in the current stage of development of the ADAS software). Binary variable models, on the other hand, provide a probabilistic chance of successful prediction based on a pre-

defined threshold, eliminating the model sensitivity to outliers. Although the use of thresholds provides advantages, defining the maximum error that a prediction can have and still be successful (and avoid a crash) can be challenging. Therefore, we tested multiple thresholds. Since the quality of the predictions prevented from the algorithms developed shall improve gradually over the prototyping phase, thresholds to define whether a prediction was successful or failed became progressively more rigorous.

We developed two types of models for each type of dependent variable: pooled and multi-level models. Pooled models ignore possible dependency relationships between observations by assuming that the observations are independent and identically distributed (IID). Multi-level models relax this condition and account for dependency between observations that, in the present study, emerge from different sources depending on the scenario under analysis. Therefore, details of the modeling approaches are provided separately for each scenario type.

Chapter 12 Urban Intersections

This chapter describes the design of the data collection and analysis for the urban intersection scenarios. We applied the six-step procedure developed in the previous chapter and collected data in seven selected intersections using two vehicles fitted with ADAS equipment (an additional vehicle was used to host the GPS antenna anchor).

12.1 Identification of Primary Factors and Factor-levels and Identification of Test Bed Locations

To identify the primary factors that should be considered in the urban intersection experiment, a comprehensive list of variables (and variable levels) that may impact the efficacy of the ADAS was created based on information from available literature ([12.1], [12.2], [12.3], [12.4]), professional judgement, and knowledge gained in the previous phase of this project. The strategy began with a comprehensive list of variables, including those that a priori seemed not to impact the performance of the system, and then narrow the list down to the most relevant ones, based on the analysis of the functionalities of each instrument. The variables were grouped into the functional categories shown below and are presented in Table 12.1:

- Intersection characteristics, geometric design, and signal type
- Driving conditions
- Traffic volume and composition
- Turning movements

Because the data collection would be performed with two vehicles, all possible conflicting and non-conflicting movements between these two vehicles were also listed (considering the most common). Although conflicting movements are the main concern when trying to prevent crashes, ADAS should be able to identify whether two vehicles are in conflicting trajectories or not, therefore, non-conflicting movements should also be considered in the analysis of performance. Additionally, non-conflicting movements may become conflicting if one vehicle weaves out of its lane. Figure 12.1 presents the twelve possible combinations of maneuvers that two vehicles may undertake in an intersection with four approaches (two-way each).

Table 12.1: Exhaustive list of variables for urban intersection scenario

Intersection Characteristics and Geometric Design	
Variable	Variable level
Pedestrian crossing types	<ol style="list-style-type: none"> 1. None 2. Uncontrolled crosswalks 3. Fixed-phase signal crosswalks 4. Pedestrian actuated crosswalks (e.g., HAWK, Puffin)
Pedestrian crossing design elements	<ol style="list-style-type: none"> 1. No curb 2. Curb ramp 3. Extended curb ramp 4. Raised median 5. Marked crosswalk 6. Unmarked crosswalk
Bike lane	<ol style="list-style-type: none"> 1. None 2. Conventional bike lane (exclusive lane adjacent to motor vehicle lane) 3. Buffered bike lane (buffer space between motor vehicle lane and bike lane) 4. Contra-flow bike lane (opposite direction flow—one-way street to two-way flow) 5. Shared right-of-way
Presence of construction activities in vicinity	<ol style="list-style-type: none"> 1. Interaction of flow from site with regular traffic (disrupted flow) 2. No interaction
Location of intersection	<ol style="list-style-type: none"> 1. CBD / high-traffic density zones 2. Low-traffic density zones
Type of intersection – based on intersecting road types	<ol style="list-style-type: none"> 1. Major intersection (one or more approach roads are “major”) 2. Minor intersection
Type of intersection – based on traffic control	<ol style="list-style-type: none"> 1. Uncontrolled 2. Signal-controlled 3. Yield-sign controlled 4. Stop-sign controlled <ol style="list-style-type: none"> a. four-way stop control b. two-way stop control

Intersection Characteristics and Geometric Design	
Variable	Variable level
Type of intersection – based on geometric configuration (grade-level)	<ol style="list-style-type: none"> 1. Three-approach legs <ol style="list-style-type: none"> a. T intersection b. Y intersection 2. Four-approach legs 3. Multi-leg (five or more) 4. Roundabout 5. Skewed intersection <ol style="list-style-type: none"> a. Greater than seventy-five b. Less than seventy-five (needs special consideration) 6. J-Turns
Turn and conflict controls	<ol style="list-style-type: none"> 1. Channelizing islands 2. Median islands 3. Corner islands (assist in turning)
Number of ways in each approach	<ol style="list-style-type: none"> 1. One-way 2. Two-way
Number of lanes in each approach	<ol style="list-style-type: none"> 1. One 2. Two 3. Three 4. Four or more
Horizontal angle between approaches (horizontal alignment)	<ol style="list-style-type: none"> 1. Rectangular 2. Acute 3. Obtuse
Lane size variations at intersection	<ol style="list-style-type: none"> 1. Narrowing lanes 2. Widening lanes
Grade level of approach roads (vertical alignment)	<ol style="list-style-type: none"> 1. Less than 3 percent 2. Greater than 3 percent but less than 6 percent 3. Greater than 6 percent
Bus-stop locations with respect to intersection	<ol style="list-style-type: none"> 1. Near side of intersection 2. Far side of intersection 3. Midblock

Intersection Characteristics and Geometric Design	
Variable	Variable level
Roadside parking	<ol style="list-style-type: none"> 1. Parking on shoulder 2. Parallel parking lane 3. Angular parking lane (45 degree, 90 degree) 4. Curb offset
Rail crossing	<ol style="list-style-type: none"> 1. Gated / barrier crossings 2. Open crossings 3. Open crossings with traffic signals
Driving Conditions	
Variable	Variable level
Light conditions	<ol style="list-style-type: none"> 1. Dawn 2. Daylight 3. Dusk 4. Dark 5. Dark artificial light
Weather conditions	<ol style="list-style-type: none"> 1. Clear 2. Overcast 3. Fog 4. Light rain/snow 5. Heavy rain 6. Heavy snow
Sight distance at intersection (vision obstructed by land use or vehicle)	<ol style="list-style-type: none"> 1. Clear 2. Blind turn

Driving Conditions	
Variable	Variable level
Types of urban canyons	<p><i>On basis of aspect ratio (canyon height H to width W)</i></p> <ol style="list-style-type: none"> 1. No canyon 2. Regular canyon (=1) 3. Avenue canyon (<0.5) 4. Deep canyon (=2) <p><i>On basis of symmetry</i></p> <ol style="list-style-type: none"> 1. No canyon 2. Symmetric canyon 3. Asymmetric canyon
Traffic Volume and Composition	
Variable	Variable level
Traffic composition	<ol style="list-style-type: none"> 1. Predominantly cars 2. Car + transit + pedestrians/bicyclists 3. Pedestrians/bicyclists + few cars
Traffic volume (qualitative)	<ol style="list-style-type: none"> 1. Low (none) 2. Medium 3. High
Vehicle Movements	
Variable	Variable level
Vehicle movements	<ol style="list-style-type: none"> 1. Straight 2. Right-turn 3. Left-turn

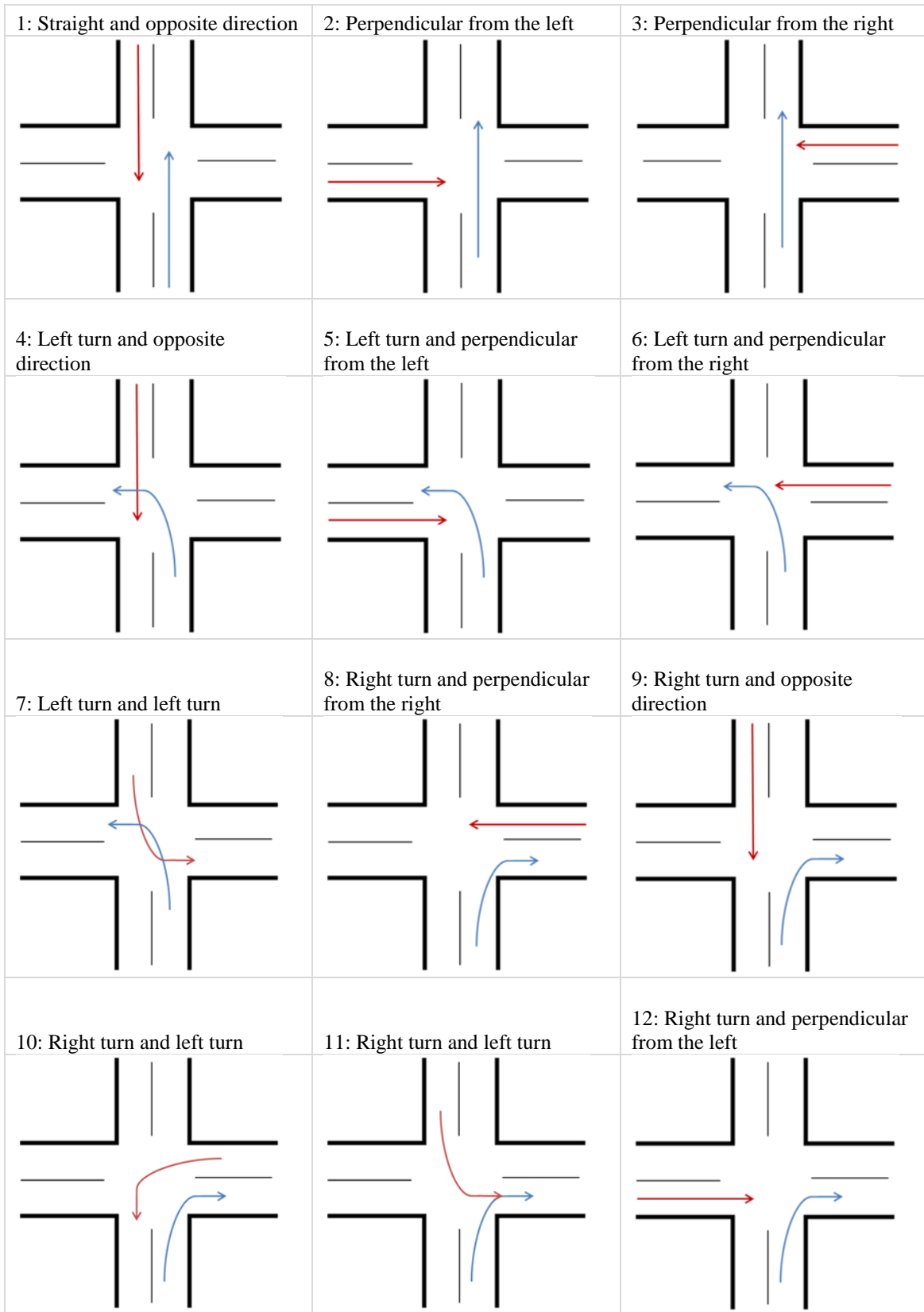


Figure 12.1: Conflicting and non-conflicting maneuvers in four-way intersection

The next step in the process of narrowing down to the most relevant factors for the experiment involved the identification of variables and levels that can directly affect the performance of each equipment in ADAS (Table 12.2). The main attributes of the instrument that impact the sensitivity include range, field-of-view (line-of-sight), number of simultaneous detections, and operating conditions. The expected performance of equipment subjected to each variable was classified as:

- F - Fail: Instrument fails to provide data
- P – Partial: Instrument can provide data but is not reliable or does not accommodate all traffic users
- R – Reliable: Instrument data is reliable for accurate predictions and accommodates all traffic users

The expected performance was obtained based on (1) the equipment layout in our test vehicles; (2) the equipment specifications provided by the manufacturer; (3) an analysis of equipment performance reported by the USDOT [12.5]; and (4) technical knowledge and experience of the research group. Note that since the radar and camera are positioned inside and toward the front of the vehicle, these units are considered to have limited field-of view. On the other hand, the LIDAR is placed on the top of the vehicle and rotates on a fixed axis, providing a 360-degree field-of-view. Although the GPS data is necessarily transmitted by the DSRC (e.g., V2V-DSRC and GPS could be considered as part of the same equipment), the variables that affect the former instrument are not the same that affect the later. For that reason, the limitations of these instruments are analyzed separately, and variables that do not affect the GPS antenna are considered not applicable for this equipment. Finally, while Table 12.2 provides only the expected individual performance of each instrument, models were used after the data collection to measure the joint performance of the equipment.

Table 12.2: Performance of instruments under different situations and conditions

Variable	Sensor			Communication		Absolute position
	RADAR	LIDAR	Camera	DSRC (V2V)	DSRC (V2I)	GPS
Road Geometric Design						
Vertical alignment of approach: low upward	P	P	P	R	R	**
Vertical alignment of approach: high upward	F	P	F	R	R	**
Vertical alignment of approach: low downward	P	P	P	R	R	**
Vertical alignment of approach: high downward	F	P	F	R	R	**
Blind Intersection/merge limiting field of view	F	F	F	P	R	**
Horizontal alignment between oncoming vehicles: rectangular angle	P	R	P	R	R	**
Horizontal alignment between oncoming vehicles: obtuse angle	R	R	R	R	R	**
Horizontal alignment between oncoming vehicles: acute angle	F	R	F	R	R	**
Driving Conditions and Obstructions						
Low light conditions	R	R	F	R	R	R
Snow, hail ,and heavy rain	R	R	F	P	P	P
Smoke, fog	R	R	F	R	R	R
Direct glare, shadows	P	R	P	R	R	R
Urban canyons	R	R	R	P	P	F
Heavy foliage	R	R	R	P	P	F
Obstructed view due to larger vehicle	F	F	F	R	R	R
Road obstructions out of line-of-sight (e.g., crash or debris far ahead on vehicle route)	F	F	F	P	R	**
Traffic Speed, Volume and Composition						
High speed traffic	P	P	P	P	P	**
High traffic volumes	P	P	P	R	R	**
Presence of pedestrians	P	R	P	F	F	**
Presence of bicycles	P	R	P	F	F	**
Presence of vehicles without communication technology in the vehicle line-of-sight	R	R	R	F	F	**
Parked vehicles	R	R	R	F	F	**

Note: ** entries = not applicable

In Table 12.2, we identify that the variables that may be most likely associated with detection/communication failure or partial reliability are vertical alignment, horizontal alignment (specifically acute angles), blind intersections, or situations with obstructed view, urban canyons, or heavy foliage, situations that involve high speed (less likely in urban intersections) or high density of traffic, and mixed traffic composition. Among these, researchers can control vertical and horizontal alignments, blind intersection, urban canyons, or heavy foliage by their choice of data collection locations. We identify that vertical and horizontal alignment should be primary factors in the experimental design, while the remaining variables should be secondary factors.

Additionally, we consider that road width largely influences the relative position of vehicles in intersections. Therefore, although road width does not seem to interfere directly with equipment performance, we select this variable as a third primary factor. Finally, the type of maneuver (relative movement between the two vehicles) is also added as a primary factor. In summary, the overall criteria for the selection of environment-related factors for the experimental design were:

- Availability of roadway infrastructure with desired levels of variables
- Line-of-sight/field-of-view affected
- Compromised range-of-operation /signal blockage (for communication equipment)
- Varying relative positions of vehicles

The final list of primary factors selected for the data collection is:

- Horizontal alignment of approach roads (three levels)
 - Rectangular
 - Acute
 - Obtuse
- Vertical alignment (gradient) of approach roads (three levels)
 - At-grade
 - Upslope
 - Downslope
- Road width: number of lanes on major approach road (three levels)
 - Two-lanes
 - Four-lanes
 - Six-lanes
- Maneuver: relative movement between the two cars
 - Conflicting: both cars going straight
 - Conflicting: one car going straight and the other turning left
 - Conflicting: one car going straight and the other turning right
 - Conflicting: both cars turning, one left and the other right
 - Non-conflicting: both cars turning, one left and the other right (traffic control: stop sign)
 - Non-conflicting: one car going straight and the other turning right

The secondary factors are:

- Blind approach due to roadway geometry (example in Figure 12.2a)
- Blind approach due to situation environment (example in Figure 12.2b)
- Traffic control type (traffic signal, all-way stop sign, stop sign on minor road)
- Presence of non-test vehicles influencing maneuvers of test vehicles

- Presence of non-test vehicles not influencing maneuvers of test vehicles
- Presence of bike lane
- Presence of two-way left turn lane
- Presence of pedestrians
- Presence of bicyclists
- Protected left turn signal vs unprotected left turn signal
- Curbside parking
- Number of approaches (three versus four)
- Weather conditions
- Heavy foliage

Seven locations (Figure 12.3, Figure 12.4, and Figure 12.5) were chosen to test the impacts of the primary and secondary factors on ADAS performance. The selection of locations was also an iterative process because, although some locations had the desired road geometric design and built environment, they presented unexpected challenges for the data collection (such as heavy traffic, high-speed vehicles, difficulty in resetting maneuvers).

- Oltorf Street/South Congress Avenue and East Riverside Drive/South Congress Avenue were chosen to test the effects of horizontal alignment (Figure 12.3). The two intersections provide approaches with rectangular, acute, and obtuse angles and have a similar number of lanes, traffic control devices, and traffic composition. Additionally, different times of the day and days of the week offer different traffic volumes and compositions.

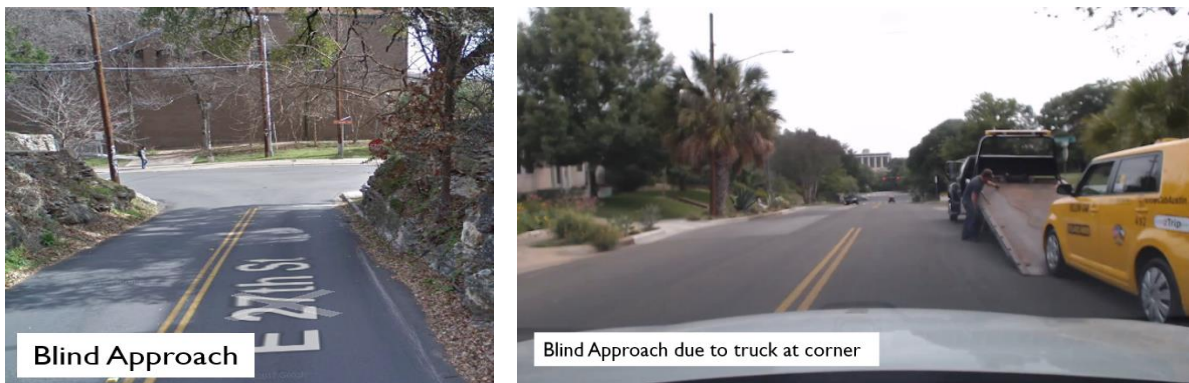


Figure 12.2: Blind approaches: (a) due to roadway geometry, and (b) situational condition.

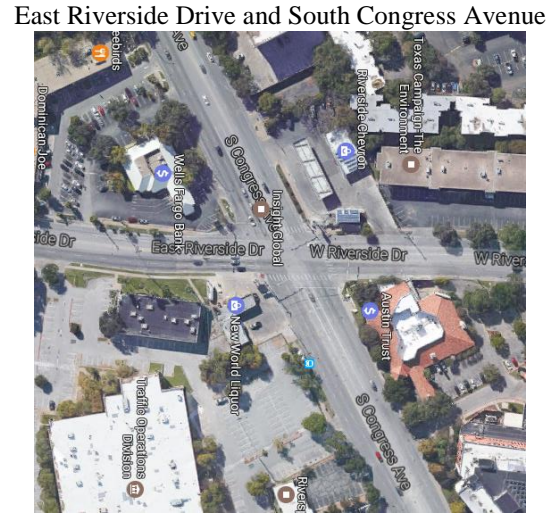
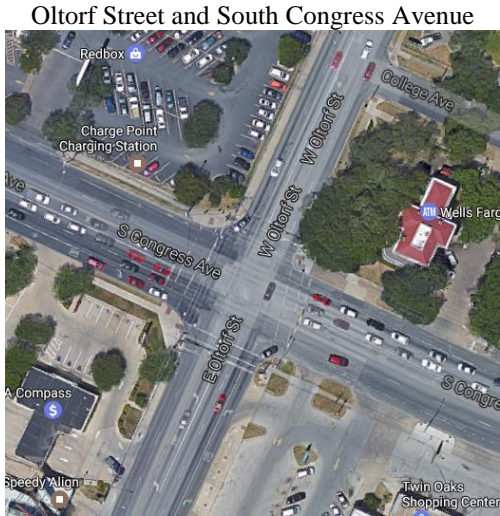


Figure 12.3: Horizontal alignment of approach roads (orthogonal versus angled)

- Cedar Street/West 34th Street, West 11th Street/Baylor Street, and East 27th Street/Speedway (Figure 12.4) were selected to test the effects of vertical alignment. All three intersections have a similar width, three approaches, and similar traffic conditions during the time of data collection. The Cedar Street intersection has approaches that are at grade; the West 11th Street intersection has an approach with upslope; and the East 27th Street intersection has downslope, in which the slopes are accentuated and cause reduced visibility. East 27th Street and Speedway features a large building one corner that can be used to test signal blockage. Cedar Street and West 34th Street has heavy foliage. The Cedar Street and West 34th Street intersection does not present stop signs in all three approaches, while the latter two intersections do. Curbside parking is allowed on the Cedar Street and West 34th Street and West 11th Street and Baylor Street.

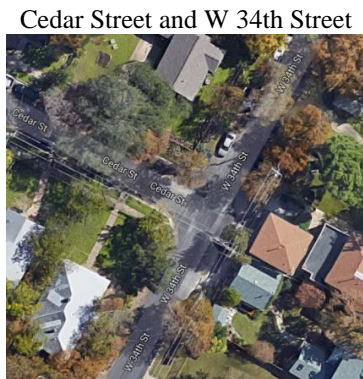


Figure 12.4: Vertical alignment (gradient) of approach roads (at-grade, upslope, downslope)

- Cedar Street and West 34th Street, Harwood Place and Travis Heights Boulevard, and West Johanna Street and South Congress Avenue (Figure 12.5) were chosen to test differences in road width. In all three cases the approaching road has a similar width but the major approach road has varying widths, measured by the number of lanes in each direction of traffic flow. All three intersections allow for curbside parking in their proximities, but the Johanna and South Congress intersection has parking lines. South Congress Avenue presents a central two-way turn lane. All three intersections have a stop sign at the minor approach road, but situations in which the vehicle on the main approach road has to yield to the vehicle emerging from minor approach road were also tested.

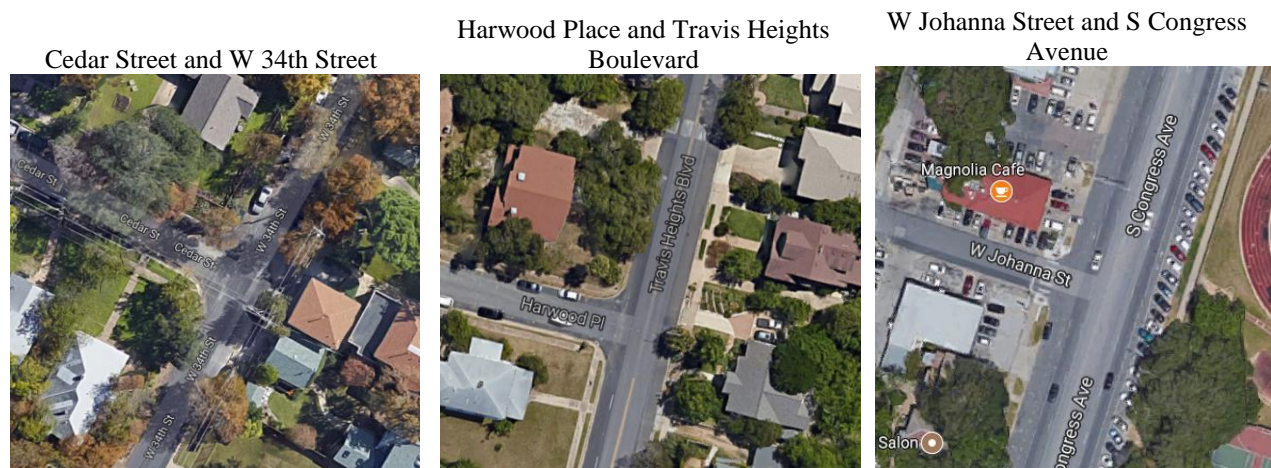


Figure 12.5: Number of lanes on major approach road (two versus four versus six lanes)

12.2 Planning and Execution of Data Collection

Data was collected between February and June 2017, most often on Sunday mornings. The choice for Sunday mornings was based on traffic volumes. Sunday morning fieldwork allowed the researchers to capture scenarios with varied traffic volumes with almost no vehicles circulating at 7am to a significant number of vehicles at 11am. Each data collection effort required five to six team members: three drivers and two to three equipment supervisors. Detailed plans, indicating start, end, and re-setting points for each vehicle were defined for each maneuver (see Figure 12.6 for an example). The exact point in time for the start of each maneuver was coordinated by phone, but a real-time visualization of the location of the vehicles based on the position data exchanged through DSRC was created to help this coordination (Figure 12.7). A third vehicle was parked close to the intersections as an anchor for the high-precision GPS unit.

Even though the final prototype (joint system of sensing and communication equipment) supports real-time data fusion, processing, and predictions, the current research exercise involves post-processing of data collected by the equipment that form the ADAS. Post-processing is advantageous for the experiment because it allows for data to be collected simultaneously by all

different communication and sensing equipment while predications can be made a posteriori using any desired combination of instruments. This allows the comparison of prediction results of different equipment configurations with parsimonious data collection.



Figure 12.6: Example of maneuver plan indicating starting, ending, and waiting areas

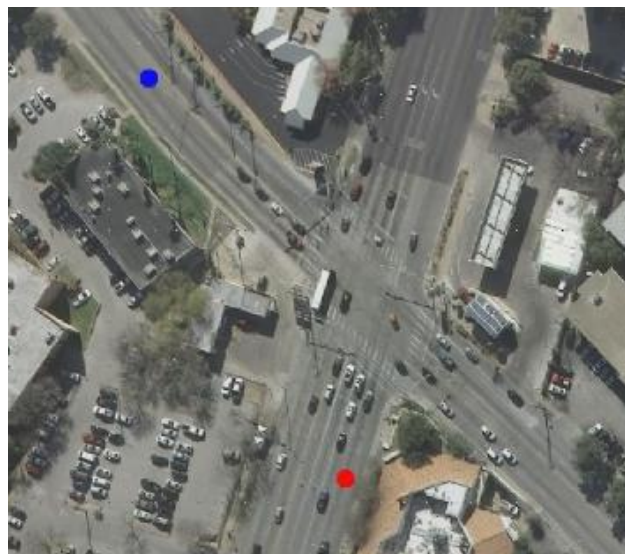


Figure 12.7: Real-time visualization of the position of the experimental vehicles based on data exchanged through DSRC

The first three months of data collection served as learning period for the team, since some important pieces of equipment were still arriving (such as the LIDAR) and installation adjustments

had to be constantly made.¹ The data collected during these months was not considered for analysis in the experimental design. The experimental design uses data collected in May and June. The total number of maneuvers and events measured is described below. A maneuver is considered as a combination of movements between the two vehicles (see Figure 12.8 for an example), while an event is a single measurement of a maneuver. While the same maneuver was repeated multiple times, each instance is considered an event.

- Horizontal alignment
 - twelve different combinations between maneuvers and locations were measured (Figure 12.8 shows all twelve combinations which are formed by three types of approaches, creating four types of maneuvers)
 - seventy-six valid events
- Vertical alignment
 - thirty-three different combinations of maneuvers and locations were measured
 - 162 valid events
- Road width
 - thirty-three different combinations of maneuvers and locations were measured
 - 170 valid events

¹ Data was also collected in an eighth intersection that is not included in the analysis (and was substituted by the intersection at Cedar Street and West 34th Street) because of problems with the high precision GPS system in that location.

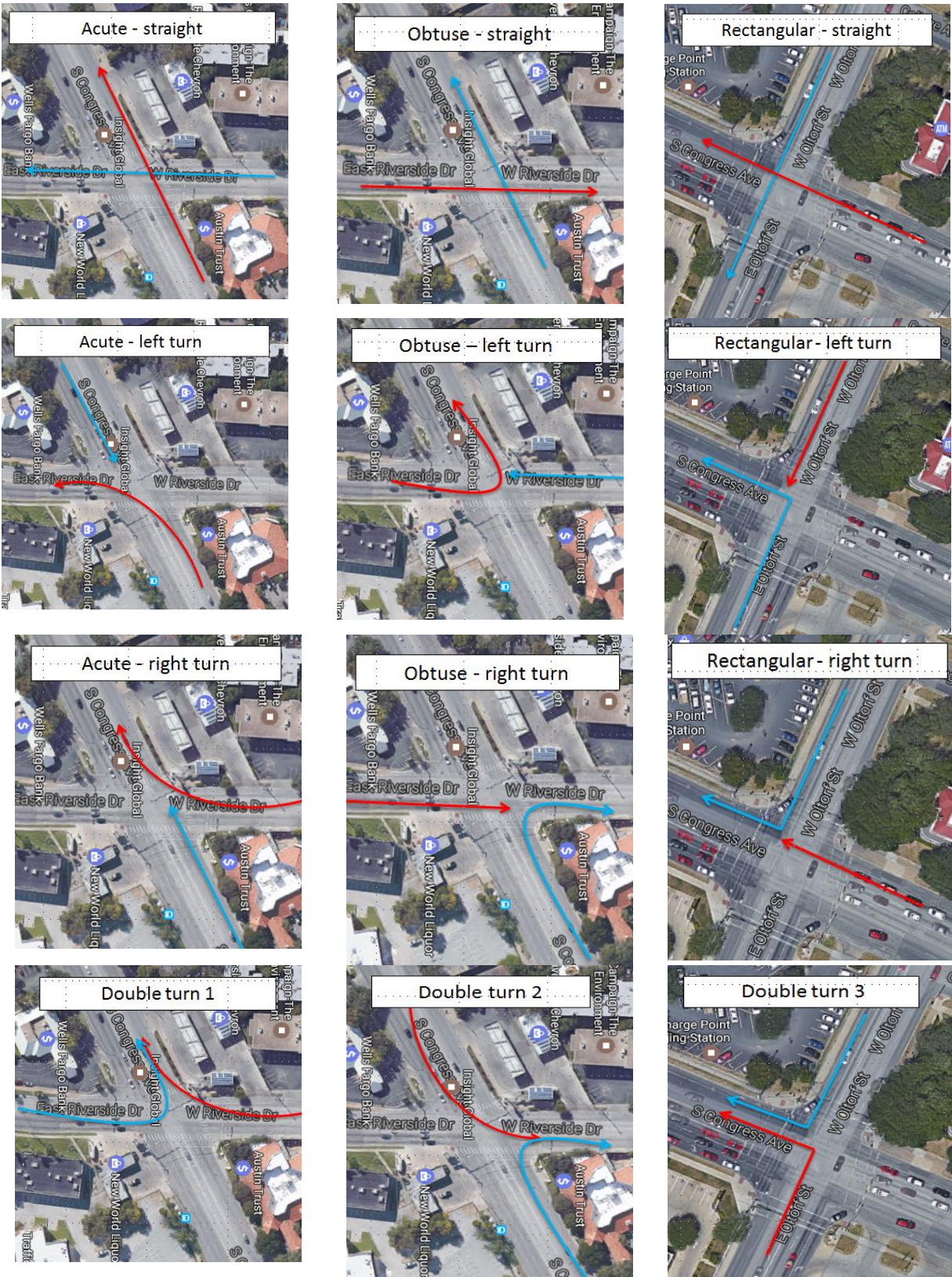


Figure 12.8: Maneuvers considered when testing the effect of horizontal alignment

12.3 Modeling Framework

For this experiment, the individual observations are locations of the target vehicle as predicted by the ADAS on the predictor vehicle. For example, in Figure 12.8, the vehicle represented by the red arrow is predicting the trajectory of the vehicle represented by the blue arrow and vice versa. Each event results in ten predictions, five for each vehicle (predictions are made for five discrete points in time, relative to time at which the vehicles cross each other's trajectories: -4s, -2s, 0s, 2s, and 4s). Each of these predictions were considered one observation.

We develop two models for each type (continuous and binary) of dependent variable: a univariate pooled model and a univariate multi-level model. Pooled models ignore possible dependency relationships between observations. In other words, these models assume that the observations are independent and identically distributed (IID). Multi-level models relax this condition and account for dependency between observations. In the analysis presented in Section 3.5, three sources of dependency are considered: dependency within a same event, dependency within a same maneuver type, and dependency within a same day. The last dependency is necessary because drivers (and their specific driving behavior) and specificities of the equipment assembling or environmental circumstances are the same for data collected in the same day but vary between different days of data collection. In the next sections, we provide the formulation of the pooled and multi-level models used to analyze the impact of the primary and secondary factors on the performance of the ADAS. For the initial analyses, all four types of models were tested for all scenarios. As the project advances, modeling approaches may be changed, replaced, or removed.

12.3.1 Pooled Linear Model

A pooled linear model treats individual observations (predicted locations) associated with each event, maneuver type, and day as independent and identically distributed random variables. The dependent variable for this model is the continuous measure for absolute error of prediction for a particular component or set of components in the ADAS. A linear regression framework captures the impact of primary and secondary factors on the performance and coefficients are estimated using ordinary least squares method. The nature and magnitude of influence of different primary and secondary factors (for example, equipment configuration, prevalent traffic conditions, etc.) are considered to be the same over all maneuver events and days of collection. The model for can thus be written as,

$$y_k = \beta + \boldsymbol{\gamma}'\mathbf{x}_k + \varepsilon_k$$

where y_k is the prediction error for observation k , β is intercept coefficient, $\boldsymbol{\gamma}$ is a $(N \times 1)$ vector of coefficients for \mathbf{x}_k ($N \times 1$) vector of exogenous covariates. Also, ε_k is the random error associated with observation k and ε_k follows a normal distribution with mean 0 and variance σ^2 . The errors are independently and identically distributed across observations.

12.3.2 Multi-level Linear Model

In these models, we proceed with the same dependent variable as the pooled linear model. Additionally, we account for the interdependencies between observations (predicted locations) due to event, maneuver type, and day of data collection. Consider a four-level hierarchical framework with the continuous dependent variable (absolute error) and individual observations (predicted locations) at level 1, event at level 2, maneuver type at level 3, and day of collection at level 4. The primary and secondary factors (for example, equipment configuration, prevalent traffic conditions, etc.) at an observation level are included as level 1 variables and we assume these variables have fixed effects (same coefficients) across observations extracted from all maneuvers and all days. Factors capturing characteristics of day of collection (for example, weather and driver-related variables) are included in the level 4 equations. The effects of level 4 variables percolate to the level 1 dependent variable of interest through the hierarchical structure. The model can be thus formulated as,

$$\text{Level 1:} \quad y_{hijk} = \beta_{hij} + \boldsymbol{\gamma}'\mathbf{x}_k + \varepsilon_{hijk}$$

$$\text{Level 2:} \quad \beta_{hij} = \sigma_{hi} + \mu_{hij}$$

$$\text{Level 3:} \quad \sigma_{hi} = \delta_h + \vartheta_{hi}$$

$$\text{Level 4:} \quad \delta_h = \alpha + \mathbf{g}'\mathbf{z}_h + \xi_h$$

$$\text{Reduced Form Model:} \quad y_{hijk} = (\alpha + \mathbf{g}'\mathbf{z}_h + \xi_h + \vartheta_{hi} + \mu_{hij}) + \boldsymbol{\gamma}'\mathbf{x}_k + \varepsilon_{hijk}$$

where y_{hijk} is the prediction error for observation k within event j from maneuver type i on day h , β_{hij} is intercept coefficient of event j , $\boldsymbol{\gamma}$ is a $(N \times 1)$ vector of observation-specific slope coefficients for \mathbf{x}_k ($N \times 1$) vector of exogenous covariates which explain variations at the observation level, and ε_{hijk} is the random error associated with observation k nested within event j , maneuver type i and day h . The effects of these exogenous variables are fixed across different events, maneuvers and days. ε_{hijk} follows a normal distribution with mean zero and variance σ^2 . In level 2, the level 1 intercept coefficient is used as dependent variable and is expressed as a function of random intercept σ_{hi} associated with maneuver type i on day h and level 2 error term μ_{hij} - random effects term for the j th event of type i on h th day. μ_{hij} follows a normal distribution with mean 0 and variance τ^2 . In level 3, the level 2 intercept coefficient is used as dependent variable and is expressed as a function of random intercept δ_h associated with day h and level 3 error term ϑ_{hi} that follows a normal distribution with mean zero and variance ρ^2 . Finally, the variations in the intercept term at level 3 are explained by level 4 equations. Intercept for day h represented by fixed effect α , exogenous variable vector \mathbf{z}_h ($T \times 1$) and associated coefficient vector \mathbf{g} ($T \times 1$), and random error term ξ_h . ξ_h follows a normal distribution with mean 0 and variance v^2 . The error terms between different levels are uncorrelated. The model parameters are estimated using the maximum likelihood approach in STATA software.

- Ignoring the presence of the hierarchical structure in the data can affect the interpretation of the results (neglecting shared variance between observations within the same group) and underestimation of standard errors of regression coefficients.

12.3.3 Pooled Discrete Model (Probit)

The dependent variable in this model takes the form of a binary variable indicating success or failure in prediction by particular component or set of components of ADAS. Akin to a pooled linear model, the pooled probit model also ignores the error correlation between observations from same event, maneuver type, and day. A simple probit model models the probability of obtaining a binary outcome as a function of variable exogenous covariates and estimates the coefficient values using the maximum likelihood method, as follows:

$$\pi_k = \Pr(y_k = 1)$$

$$\Phi^{-1}(\pi_k) = \beta + \boldsymbol{\gamma}'\mathbf{x}_k$$

where, y_k is the binary variable which takes the value 1 if the system makes a successful prediction of the test vehicle in observation k , Φ is the standard normal cumulative distribution function, whose inverse acts as the link function, β is the intercept and $\boldsymbol{\gamma}$ is a $(N \times 1)$ vector of slope coefficients for \mathbf{x}_k ($N \times 1$) vector of exogenous covariates which incorporates fixed effects of primary and secondary variables.

12.3.4 Multi-level Discrete Model (Probit)

With the same dependent variable as the pooled probit model, consider a four-level hierarchical model with the binary variable of interest (successful detection of test vehicle) at level 1, event at level 2, maneuver type at level 3, and day of data collection at level 4. The primary variables and secondary variables (equipment configuration, ambient situational environment and prevalent traffic conditions), are included as level 1 variables with fixed effects (same coefficients across maneuvers and days of collection). Weather related variables are included in the level 4 equations, which model the level 3 intercept as a random variable. The probit model is formulated as follows:

$$\begin{aligned} \text{Level 1: } & \pi_{hijk} = \Pr(y_{hijk} = 1) \\ & \Phi^{-1}(\pi_{hijk}) = \beta_{hij} + \boldsymbol{\gamma}'\mathbf{x}_k \\ \text{Level 2: } & \beta_{hij} = \sigma_{hi} + \mu_{hij} \\ \text{Level 3: } & \sigma_{hi} = \delta_h + \vartheta_{hi} \\ \text{Level 4: } & \delta_h = \alpha + \mathbf{g}'\mathbf{z}_h + \xi_h \end{aligned}$$

$$\text{Reduced Form Model: } \Phi^{-1}(\pi_{hijk}) = (\alpha + \mathbf{g}'\mathbf{z}_h + \xi_h + \vartheta_{hi} + \mu_{hij}) + \boldsymbol{\gamma}'\mathbf{x}_k$$

where, y_{hijk} is the binary variable which takes the value 1 if the system makes a successful prediction of the test vehicle in observation k , event j , maneuver type i and day of collection h . Φ

is the standard normal cumulative distribution function, whose inverse acts as the link function. β_{hij} is intercept coefficient of event j , $\boldsymbol{\gamma}$ is a $(N \times 1)$ vector of observation-specific slope coefficients for \boldsymbol{x}_k ($N \times 1$) vector of exogenous covariates which explains fixed effects of the exogenous covariates. In level 2, the level 1 intercept coefficient is used as dependent variable and is expressed as a function of random intercept σ_{hi} associated with maneuver type i on day h and level 2 error term μ_{hij} - random effects term for the j th event of type i on h th day. μ_{hij} follows a normal distribution with mean 0 and variance τ^2 . In level 3, the level 2 intercept coefficient is used as dependent variable and is expressed as a function of random intercept δ_h associated with day h and level 3 error term ϑ_{hi} that follows a normal distribution with mean zero and variance ρ^2 . Finally, the variations in the intercept term at level 3 are explained by level 4 equations. Intercept for day h represented by fixed effect α , exogenous variable vector \boldsymbol{z}_h ($T \times 1$) and associated coefficient vector \boldsymbol{g} ($T \times 1$), and random error term ξ_h . ξ_h follows a normal distribution with mean 0 and variance v^2 . The error terms between different levels are uncorrelated. The model parameters are estimated using the maximum likelihood approach in STATA software.

12.3.5 Results

The preliminary results suggested that both DSRC and radar performance deteriorates when vehicles approach each other in intersections with acute horizontal angles. The built environment in such approaches may also block communication signals. Based on DSRC data, left turns seemed to be the most difficult maneuvers to predict successfully. However, radar data seemed to serve this type of prediction well, indicating that the technologies may have an important complementarity effect.

The expansion from preliminary to full results proved impossible due to the difficulty of obtaining accurate ground truth location information, which was likely a result of the large number of trees in the testing area. The usage of high-precision GPS mitigated this issue, but the errors were still too large and rendered the data unusable: after “correction” through the high-precision GPS, vehicles were still displayed “floating” on sidewalks or in other impossible locations, indicating the low quality of the corrections made.

Upon presenting these results to TxDOT, instead of repeating the data collection effort with even higher GPS accuracy and precision, the research team was asked to instead focus their efforts on the three work-zone-related demonstrations (i.e., the “Construction Zone,” “Pedestrian Crossing,” and “Highway Merging” demonstrations), described in Chapter 15.

Chapter 13 Overtaking Maneuvers on Rural Roads

This chapter describes the design of the data collection and analysis for investigating overtaking maneuvers on rural roads scenarios. Overtaking is the act of passing the vehicle in front of you by temporarily changing lanes. We specifically study overtaking maneuvers on two-lane two-way rural roads, where the lane used for passing may have traffic in the opposite direction. As discussed earlier, ADAS can use sensing and communication equipment to help drivers identify whether it is safe to execute an overtaking maneuver by detecting oncoming vehicles, assessing their speeds, and determining if the available gap between vehicles is sufficient for the overtaking to take place or not. In Phase 1 of this project, a V2V-based overtaking assistant was studied. This assistant determined whether an overtaking maneuver would be unsafe before the driver changed lanes. The performance of the overtaking ADAS was tested using simulated maneuvers that were defined based on information from existing literature that characterized typical overtaking variables and the capabilities of V2V communication. In this phase of the project, real tests were performed to assess whether the communication units in our prototype vehicles had the capability to provide timely information to assist on the choice to overtake. In addition to the communication capabilities, radar, LIDAR, and camera were also tested as potential oncoming-vehicle detectors for an overtaking ADAS. An experimental design was developed to create scenarios for the testing of the communication and sensing equipment. Data was collected in a FM road in Austin, Texas, using the three prototype vehicles, which emulated typical overtaking situations in two-lane two-way rural roads. As depicted in Figure 13.1, Vehicle 1 was the overtaker (vehicle that wishes to perform the overtaking maneuver), Vehicle 2 was the vehicle to be overtaken, and Vehicle 3 was the oncoming vehicle (vehicle coming in the opposite direction) that is in the lane use for overtaking. To ensure a safe experiment in a real world rural road environment, no actual overtaking maneuvers were executed. Such restriction was not a limitation to the current study because the role of the ADAS is simply to identify safe overtaking gaps and not to perform the actual maneuver.

13.1 Experimental Design

The six main variables that may influence overtaking maneuver scenarios were:

- 1) Speed of the overtaken vehicle (Vehicle 2)
- 2) Speed of the oncoming vehicle (Vehicle 3)
- 3) Acceleration of the overtaken vehicle (Vehicle 2)
- 4) Acceleration of the oncoming vehicle (Vehicle 3)
- 5) Gap between the overtaker (Vehicle 1) and overtaken vehicle (Vehicle 2).
- 6) Acceleration of the overtaker vehicle (Vehicle 1) during the actual overtaking maneuver.

Since no overtaking actually took place, the experimental design only took the first five variables mentioned above into consideration. The levels tested were as follows:

- Speed of the overtaken vehicle (Vehicle 2)
 - 40 mph
 - 50 mph
 - 60 mph
- Speed of the oncoming vehicle (Vehicle 3)
 - 40 mph
 - 50 mph
 - 60 mph
- Acceleration of the overtaken vehicle (Vehicle 2)
 - Decelerating
 - Constant speed
 - Accelerating
- Acceleration of the oncoming vehicle (Vehicle 3)
 - Decelerating
 - Constant speed
 - Accelerating
- Gap between the overtaker (Vehicle 1) and overtaken vehicle (Vehicle 2)
 - 3 seconds
 - 6 seconds

The speeds used took into consideration the speed limits of the roadway used for testing. The drivers could not control for precise accelerations, therefore a less precise approach was used in which vehicles maintained a constant gap of either 3 or 6 seconds by means of synchronously accelerating or decelerating. The 3- and 6-second gaps were used because they are commonly used as guidelines and rules-of-thumb for safe roadway driving.

The experiment consisted of a full factorial design with 162 runs.

We did not consider areas where overtaking would have been illegal, such as curves or areas with significant gradient change.

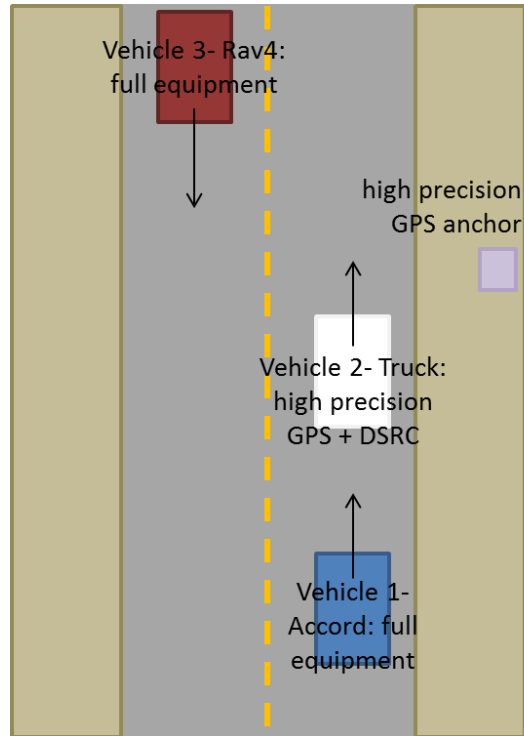


Figure 13.1: Diagram for overtaking maneuver

13.2 Data Collection

The final prototype (joint system of sensing and communication equipment) supports real-time data fusion, processing, and predictions. This analysis instead post-processed data collected by the equipment that form the ADAS. Post-processing is advantageous for the experiment because it allows for data to be collected simultaneously by all different communication and sensing equipment while predictions can be made a posteriori using any desired combination of instruments. This allows the comparison of prediction results of different equipment configurations with parsimonious data collection.

The main location used for data collection location was FM 973, close to Shumaker Enterprises Inc. The precise address is 2900 N FM 973, Austin, TX 78725 and can be seen in Figure 13.2.



Figure 13.2: Location of data collection for overtaking maneuvers

Each of the three vehicles had an associated number and function:

- Vehicle 1: Honda Accord – hypothetical overtaker
- Vehicle 2: Truck F-150 – hypothetical overtaken
- Vehicle 3: Toyota RAV4 – oncoming vehicle

The maneuver was synchronized by following the script below:

- Vehicle 2 left its starting point, then 5 seconds later vehicles 1 and 3 left their starting points (vehicles 1 and 2 had the same starting point). Note: the starting and ending points had area for safely maneuvering the vehicle.

- The vehicles reached the designated speed (defined in the experimental design) when passing the North or South Cone (placed approximately 300 m from each starting point)(Figures 13.3 and 13.4).
- After passing the Cone, vehicles 2 and 3 followed the “acceleration instruction” of the experimental design.
- Vehicle 1 kept a gap from vehicle 2 of either 3 or 6 seconds according to the experimental design.
- Each vehicle stopped at the designated stopping point.
- After finalizing each run, the vehicles were reset to start the next run in the opposite direction (e.g., if vehicle 1 was going south, it would go north in the following run).



Figure 13.3: Location of South Cone for overtaking maneuver

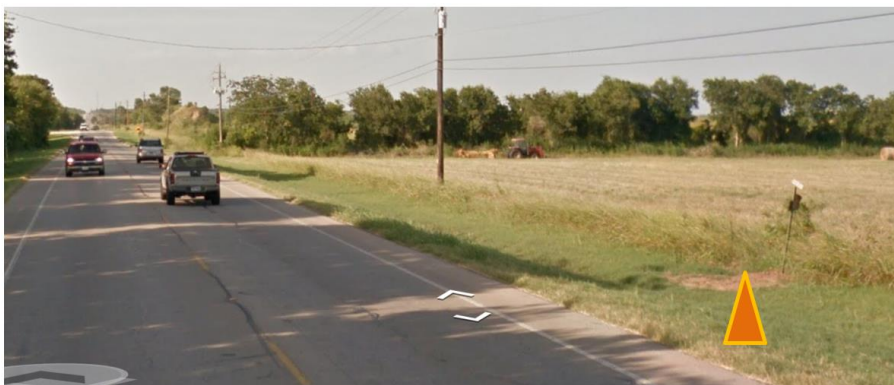


Figure 13.4: Location of North Cone for overtaking maneuver

13.3 Data Processing

Advance prediction of the details of an overtaking maneuver, as studied in Phase 1 of this project, was not the focus of this task. Instead, we identified the hypothetical last safe maneuver—or equivalently, first unsafe maneuver—each time the lead and oncoming vehicles drove past each other. First, we identified the ending time of the last safe maneuver by finding the last point in time where the gap between oncoming and lead vehicles was large enough to allow the overtaking vehicle to switch back to the right lane and still be 2 meters ahead of the overtaken vehicle. Then the hypothetical time at which the overtaking vehicle would have started overtaking was reversed engineered, using the vehicle’s positions and an assumed overtaking acceleration. Since this acceleration is unknown, we present results for four different assumed accelerations: 1.0, 1.5, 2.0, and 2.5 meters per second squared, which, translate to approximately 2.2, 3.6, 4.5, and 5.6 miles per hour per second, respectively. These values follow the findings of [13.1]. Table 13.1 shows the summary statistics of the hypothetical maneuvers. “Critical Gap” represents the physical distance between the locations where the last safe maneuver should start and end. “Detection Distance” represents the distance between the overtaking vehicle and the oncoming vehicle. Finally, “Maneuver Time” represents how much time the last safe overtaking maneuver would take considering the acceleration times on the left hand side of the table. The maximum distance at which detection is required is roughly 800 meters or 2500 feet, which is similar to the distances simulated in Phase 1.

Table 13.1: Summary results for overtaking maneuvers

Acceleration (m/s ²)	Critical Gap (m)			Detection Distance (m)			Maneuver Time (s)		
	Max	Mean	Min	Max	Mean	Min	Max	Mean	Min
1.0	821.5	392.2	131.5	1803.3	692.8	227.7	40.7	15.4	6.7
1.5	797.1	333.5	117.1	1695.8	545.2	196.7	35.2	11.8	5.8
2.0	779.5	298.3	105.9	1621.0	466.5	173.0	33.0	10.0	5.1
2.5	749.2	275.3	98.0	1534.9	417.5	156.3	30.8	8.9	4.6

13.4 Performance Evaluation

To reliably warn against unsafe maneuvers, a sensor must detect the oncoming vehicle at or before the last safe maneuver’s starting time. Table 13.2 shows the number of hypothetical test maneuvers for which each sensor detected the oncoming vehicle in time. V2V communication from the oncoming vehicle was almost always received in time. Most of the maneuvers for which V2V communication failed fell in the lowest assumed acceleration. This is mostly because the lower acceleration rates require much larger maneuver times and therefore larger distances between the vehicles, making V2V communication less reliable. Of the runs where V2V failed, the Detection Distances were all above 700 meters. This finding supports the conclusion from Phase 1 that typical V2V devices may not function across a high enough distance to reliably assist

for overtaking maneuvers. However, the oncoming vehicle was detected more often in these tests than in Phase 1’s simulations, seeming to be very reliable for distances less than 700 meters.

Table 13.2: Overall performance of camera, radar and V2V communication

Acceleration (m/s ²)	Successful detections of oncoming vehicle			Number of Maneuvers
	Camera	Radar	V2V Comm.	
1.0	2	0	118	153
1.5	3	0	136	153
2.0	5	0	145	153
2.5	7	0	148	153

The non-communicative sensors were entirely unsuccessful as overtaking assistants, as was expected. Two factors inhibit typical automotive sensors, first being the distance at which an oncoming vehicle must be detected. The radar and LIDAR devices used have advertised ranges of 200 and 100 meters respectively. Secondly, all sensors were positioned at the center of the vehicle, as is common for Level 2 and 3 automated vehicles. A large lead vehicle, such as the pickup truck used in these tests, will block the view of the road ahead from sensors. Figure 13.5 shows an image taken from the camera at the starting time of last safe maneuver, where the oncoming vehicle is invisible due to either the high distance or the truck’s occlusion. Note that human drivers are much less affected by the second factor as they are positioned to the left of the car, nearer to the opposite side of the road. Sensors positioned to the side of the vehicle would be similarly capable of seeing past the lead vehicle. Figure 13.6 shows the percentage of overtaking maneuvers that could hypothetically be warned against by a line-of-sight sensor depending on its position. “Lateral Distance” represents the distance between the hypothetical sensor and the center of the vehicle. The hypothetical sensor in Figure 13.6 can detect objects at any distance, which is untrue for our current sensors.



Figure 13.5: Lead vehicle blocking view for overtaking maneuver

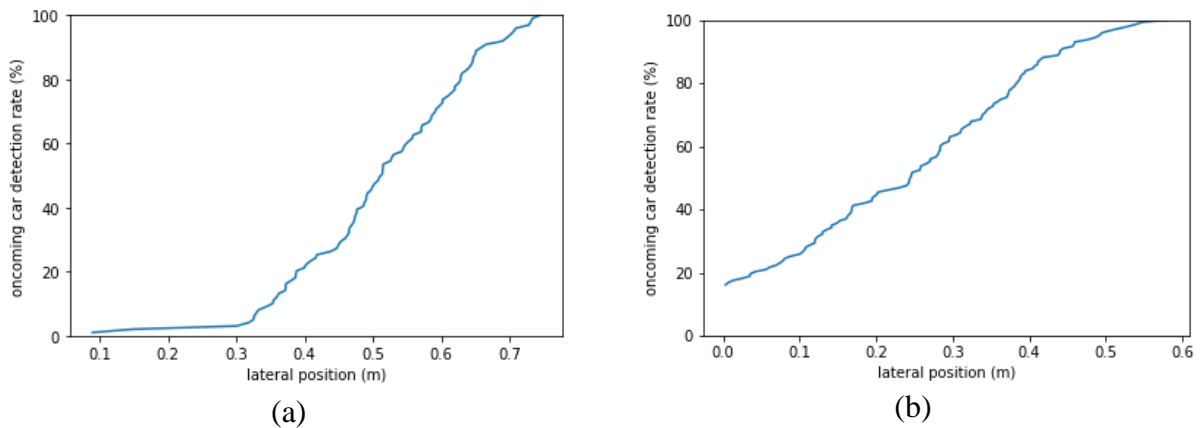


Figure 13.6: Percentage of hypothetical detection of oncoming vehicle as a function of sensor position—(a) Low acceleration: 1.0 m/s², (b) High acceleration: 2.5 m/s²

It can be assumed that vehicular sensors are usually laterally positioned near the center of vehicles, excepting highly autonomous vehicles with a large suite of sensors, because this position is the most useful for most applications. Thus, an overtaking warning assistant without V2V would require long-distance sensors equipped on the side of the vehicle solely for this application, at which point its benefits may not match its cost. On the other hand, an overtaking assistant with V2V is entirely achievable but may require a higher communication range.

13.5 Conclusions

The research team carried out a controlled experiment to evaluate the performance of ADAS during overtaking maneuvers in rural roads. In these tests, there no actual overtaking truly

took place. Such restriction was not a limitation to the current study because the role of the ADAS is simply to identify safe overtaking gaps and not to perform the actual maneuver.

The experiment showed the research team that non-communication-based technology, such as camera and radar, performed very poorly and rarely (if ever) successfully detected the presence of the oncoming vehicle. The poor performance of the non-communication technology is due to two main factors. Firstly, the short detection range of each sensor compared to the true distance between vehicles. Secondly, the positioning of the sensors in the middle of the vehicle also negatively affected their ability to detect the oncoming vehicle. V2V technology, on the other hand, performed very well and was able to detect the oncoming vehicle in the vast majority of the runs. The situations in which V2V communications did not work were mainly low acceleration scenarios which generated large distances between the overtaking vehicle and the oncoming vehicle.

These results indicate that an overtaking warning assistant is achievable in two ways:

- with a V2V communication system with slightly higher communication range than what was used here and without sensing capabilities;
- without a V2V communication system and with long-distance sensors equipped on the side of the vehicle solely for this application

The safest alternative is likely to have long-distance sensors on the side of the vehicle as well as long-distance V2V communication system in place. This might, however, prove too costly to implement in real-scale applications.

Chapter 14 Pedestrian Detection

This chapter describes the design of the data collection and analysis for the pedestrian detection scenario. In this scenario, we investigate the ADAS' capability of detecting and accurately measuring the distance between pedestrians that are crossing a road and an approaching vehicle. For this particular scenario, since there is no other vehicle to communicate with, the ADAS can only use sensing equipment to help drivers identify if there are pedestrians, cyclists or any other entities crossing the road. For the sake of simplicity, instead of referring to "pedestrians, cyclists or any other entities crossing the road" multiple times, we will simply use the term "pedestrians" in this section as a catchall term.

We used four different algorithms to determine pedestrian's positions: camera only, LIDAR only, shallow fusion, and deep fusion. As the names suggest, "camera only" and "LIDAR only" relied solely on those respective sensors. The "shallow fusion" and "deep fusion" approaches relied on a combination of the data obtained by these two sensors.

Besides looking at the overall performance, the research team evaluated how certain factors influenced the successful detection of pedestrians, such as the total number of pedestrians in the frame and the distance between the vehicle and the pedestrians. The research team also studied how these factors impacted the magnitude of the detection error as well.

14.1 Experimental Design

The two main variables that influence the pedestrian detection scenarios were:

- 1) Number and type of elements crossing the road
- 2) Speed of the vehicle

The levels tested in the first data collection effort were as follows:

- Number and type of elements crossing the road
 - Single pedestrian
 - Single cyclist
 - Two pedestrians
 - Two cyclists
 - Multiple pedestrians
 - Mix of pedestrians and cyclists
- Speed of the vehicle
 - Stopped
 - 20 mph
 - 35 mph
- Location
 - Location 1
 - Location 2
 - Location 3

The levels tested in the second data collection effort were as follows:

- Number and type of elements crossing the road
 - Single pedestrian
 - Two pedestrians
 - Multiple pedestrians
- Speed of the vehicle
 - 20 mph
 - 35 mph
- Location
 - Location 1
 - Location 2
 - Location 3

The speeds were kept low so that the research team could ensure the safety of the pedestrians crossing the road. The first two locations were on different sides of Innovation Boulevard on the J.J. Pickle Research Campus. One side was wide open and exposed to the sun. The other had trees on one side and a fence on the other, and thus was shaded and near a large number of non-pedestrian objects. The third location was the west end of Read Granberry Trail on the J.J. Pickle Research Campus. This third location had a single tree, as well as some nearby parked vehicles and objects such as a fire hydrant. These three locations were selected to test the pedestrian detection algorithms in visually different environments.

14.2 Data Collection

The original scope of project only considered one day of data collection. However, the research team encountered several difficulties to define ground truth for the data from the first date, rendering that effort's data mostly unusable. This prompted the need for a second data collection effort. For this second data collection effort, a new and more precise LIDAR sensor was also used, further improving the quality of the data.

The general setup for the data collection can be seen in Figure 14.1.

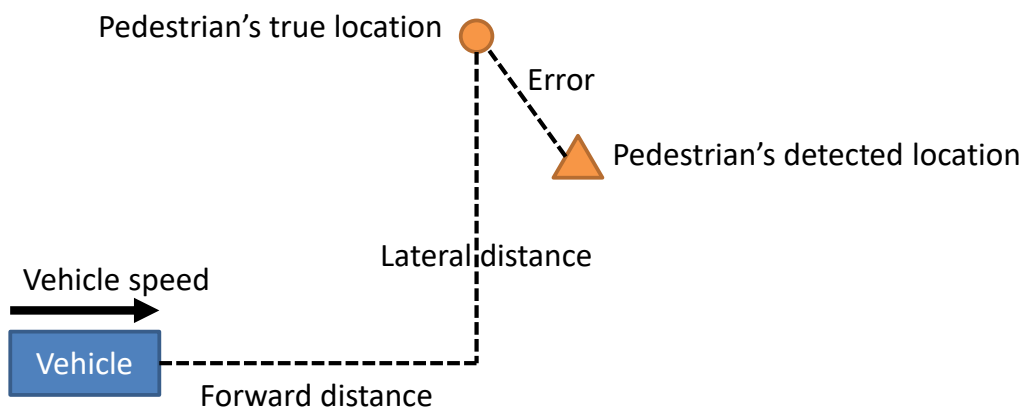


Figure 14.1: Diagram of pedestrian detection scenario

14.3 Performance Evaluation

The overall performance of the pedestrian detection was evaluated using three metrics: the average successful detection rate, which indicates how often the sensors and algorithms correctly detected the presence of a pedestrian; average error, which represents the average Euclidian distance between the perceived and true positions of the pedestrian (as illustrated in Figure 14.1); and false positive count, which illustrates how many instances the sensors detected a pedestrian crossing the road when, in reality, there were none. The main results, summarized in Table 14.1, show us that the LIDAR only alternative performed rather poorly: it had the lowest successful detection rate and the highest false positive count. The shallow fusion algorithm showed surprisingly positive results with a high successful detection rate and considerably low false positive count. The average error was also quite small. Curiously, although the deep fusion algorithm is more complex, it displayed a lower successful detection rate than the shallow fusion algorithm. Conversely, though, it had fewer false positives and a smaller average error.

Table 14.1: Overall performance measures for pedestrian detection

Technology	Avg. Successful Detection Rate	Average Error (m)	False positive count
Camera Only	71%	2.0	2,045
LIDAR Only	65%	1.7	10,265
Shallow Fusion	96%	1.4	1,161
Deep Fusion	75%	1.3	755

14.3.1 Performance Models

Besides the overall performance results presented above, the research team investigated what factors affected the successful detection and the magnitude of the error of each of the algorithms.

For the investigation of the successful detection, we used a simple probit regression. In this case, consider q as an index that uniquely identifies a pedestrian in time. Therefore, if there are 3 pedestrians crossing the road and there are 100 frames (images), q would range from 1 to 300. In this case, we have

$$Y_q^* = \mathbf{X}_q' \boldsymbol{\beta} + \epsilon_q, \quad Y_q = I(Y_q^* > 0) \quad (1)$$

where Y_q^* is a continuous unobserved (latent) variable which indicates the propensity of being correctly detected by the algorithm, \mathbf{X}_q is a vector of covariates influencing the successful detection, $\boldsymbol{\beta}$ is a vector of parameters to be estimated, ϵ_q is the error term which is assumed to be standard normally distributed (considered independent and identically distributed across all q), Y_q is a binary variable that indicates whether or not the pedestrian was successfully detected in a particular frame, and $I(\cdot)$ is the indicator function.

For the investigation of the magnitude of the errors, we used a log-regression. Here too consider q as an index that uniquely identifies a pedestrian in time. For the log-regression, we have:

$$Z_q = \exp(\mathbf{W}'_q \boldsymbol{\gamma} + \xi_q) \Rightarrow \ln(Z_q) = \mathbf{W}'_q \boldsymbol{\gamma} + \xi_q \quad (2)$$

where Z_q represents the Euclidian distance between the pedestrian's true and perceived positions (i.e., the error), \mathbf{W}_q represents a vector of covariates influencing the magnitude of the error, $\boldsymbol{\gamma}$ is a vector of parameters to be estimated and ξ_q is error term, assumed to be normally distributed (considered independent and identically distributed across all q).

14.3.2 Results

The two models described above were estimated for all four algorithms resulting in eight models. The results of the detection (probit) and the error (log-regression) models for the camera only and LIDAR only algorithms can be found in Table 14.2, while the results for the shallow fusion and deep fusion can be found in Table 14.3.

For the camera only algorithm, most variables behaved as expected: the propensity of successful detection is reduced when there are multiple pedestrians on screen, when the pedestrians bunch together and when the vehicle is travelling at higher speeds. Curiously, though, the camera only algorithm displayed better performance at larger distances. The error model also shows mostly expected results: errors are smaller at slower speeds and when there were fewer pedestrians on screen. But here too distance plays an unintuitive role: larger distances between the vehicles and the pedestrians are correlated with smaller errors.

The LIDAR only algorithm is less effective at correctly detecting pedestrians when they are bunched together, especially in groups of five or more. Slower speeds and smaller distances between the vehicle and pedestrians improve correct detection rates. The distance between pedestrians did not have a significant effect on the average the error of the algorithm. Larger vehicle speeds were associated with larger errors while larger distances to the vehicle were correlated with smaller errors. This might be because very few "true" positives were observed. In addition, having multiple pedestrians in the frame increased the error when compared to the case with a single pedestrian, as already expected.

The shallow fusion algorithm's correct predictions were associated with smaller distances between pedestrians, lower vehicle speeds, fewer pedestrians in the frame and smaller distances between the vehicle and the pedestrians. The errors for the shallow fusion algorithm followed similar patterns: smaller distances between pedestrians, greater distances between the vehicle and the pedestrians, higher vehicle speeds, and the presence of multiple pedestrians in the frame increased the magnitude of the algorithm's error.

The deep fusion algorithm's correct predictions did not seem to be statistically correlated with the distance to the vehicle. Greater distances between pedestrians and faster vehicle speeds, however, were associated with the algorithm's lower accuracy. The presence of four or more pedestrians in the frame also reduced the probability of the deep fusion algorithm's correct

prediction. Interestingly, the deep fusion algorithm’s error model shows that the errors increase as pedestrians are further from each other and when there. As expected, the errors decrease with lower speeds and smaller distances between the vehicle and the pedestrian. The errors also decrease when there is only one pedestrian in frame. The locations (i.e., visual surroundings) of the runs was controlled for and seemed to impact several of the models. These differences, however, were just taken into consideration for control purposes and have not yet been investigated in more detail.

Table 14.2: Performance modeling for “camera only” and “LIDAR only” alternatives

	Camera Only				LIDAR Only			
	Detect (Probit)		Error (Log-Reg)		Detect (Probit)		Error (Log-Reg)	
	Est.	t-stat	Est.	t-stat	Est.	t-stat	Est.	t-stat
Distance to nearest pedestrian	-0.091	-3.615	-0.040	-1.970	-0.085	-3.327	---	---
Distance between pedestrian and vehicle	0.290	5.287	-0.141	-2.861	-0.266	-5.041	-0.217	-4.911
Vehicle speed	-3.533	-16.630	0.806	4.846	-1.967	-9.853	1.238	9.494
Total pedestrians in frame (base: 1 pedestrian)								
2 pedestrians	---	---	0.317	2.752	---	---	---	---
3 pedestrians	-0.363	-4.022	0.317	2.752	---	---	0.331	4.317
4 pedestrians	-0.363	-4.022	0.317	2.752	---	---	0.331	4.317
5 pedestrians	-0.599	-5.711	0.317	2.752	-0.367	-5.479	0.331	4.317
6 pedestrians	-0.599	-5.711	0.317	2.752	-0.367	-5.479	0.331	4.317
Location (base: Location 1)								
Location 2	---	---	---	---	---	---	---	---
Location 3	-0.624	-9.842	---	---	---	---	0.253	6.507

Table 14.3: Performance modeling for “shallow fusion” and “deep fusion” alternatives

	Shallow Fusion				Deep Fusion			
	Detect (Probit)		Error (Log-Reg)		Detect (Probit)		Error (Log-Reg)	
	Est.	t-stat	Est.	t-stat	Est.	t-stat	Est.	t-stat
Distance to nearest pedestrian	-0.075	-2.820	-0.076	-4.438	-0.140	-5.032	0.029	2.209
Distance between pedestrian and vehicle	-0.246	-4.357	0.277	7.355	---	---	-0.136	-4.137
Vehicle speed	-2.652	-11.569	0.331	2.340	-1.130	-7.292	0.676	6.041
Total pedestrians in frame (base: 1 pedestrian)								
2 pedestrians	---	---	0.502	3.873	---	---	-0.363	-5.236
3 pedestrians	---	---	0.502	3.873	---	---	-0.363	-5.236
4 pedestrians	-0.350	-5.270	0.502	3.873	-0.426	-5.216	-0.363	-5.236
5 pedestrians	-0.350	-5.270	0.502	3.873	-0.816	-8.407	-0.363	-5.236
6 pedestrians	-0.350	-5.270	0.502	3.873	-0.816	-8.407	-0.363	-5.236
Location (base: Location 1)								
Location 2	0.228	2.859	-0.288	-6.018	-0.439	-4.317	---	---
Location 3	0.262	3.187	-0.255	-5.261	-1.609	-16.044	1.058	28.131

14.4 Conclusions

The results from the pedestrian detection scenarios show that the sensors (and algorithms) are usually more prone to correctly detect the presence of pedestrians and to reduce the detection errors when the vehicle is at slower speeds and closer to the pedestrians. Having fewer pedestrians simultaneously in the frame also usually improves prediction, as do larger distances between the multiple pedestrians in the frame.

Chapter 15 ADAS Application Tests

The intersection and overtaking applications were analyzed with simulations in Phase 1 of this project and with collected data in Phase 2, because real-time tests of a warning system would be difficult to safely perform. Three other applications were selected that allowed for safer demonstrations and tests of the full ADAS system. A demonstration of each of these applications was shown to TxDOT officials on September 5, 2018. They are each relevant to traffic construction safety, though this is not the only field that would potentially benefit.

15.1 Application 1: Construction Zone

15.1.1 Goals of the Demo

Road construction areas are considered one of the more dangerous transportation environments. A driver or a passenger in a vehicle died in 85% of fatal work zone crashes (25,713 crashes in Texas in 2016, with 172 fatalities). In this demo we tried to show the value of a bird's eye view app based on sensing at the infrastructure to increase safety for both, construction workers and drivers or pedestrians, who accidentally enter the construction area.

Bird's eye view displays are one of the use cases considered for 5G supporting vehicular communication. This type of displays show a top view of a given area, expanding the sensing range that a pedestrian or a driver without additional resources might have. The bird's eye view includes a map of the area and images that represent potential hazards. An example of a bird's eye view display is shown in Figure 15.1 (taken from [15.1]).



Figure 15.1: Example of an experimental bird's eye view display

The final goal of the demo is to show how sensing and communication at the infrastructure can provide the information necessary to warn pedestrians and drivers about possible hazardous situations.

15.1.2 Modules Developed for the Road Construction Demo

Sensing Base Station

We designed and developed the prototype sensing BS shown in Figure 15.2. It includes a pole, communication module, a radar, a camera, and a laptop in charge of processing the information obtained by the sensors.



Figure 15.2: Prototype sensing base station developed for the demo

Bird's Eye View App

We developed an app that can run on a smart phone or a tablet that receives the processed sensor data from the prototype sensing base station. The detected objects are sent to the smart phone or tablet in real time, and shown as images overlaid over the map of the area. The app also generates warning images and sounds when a pedestrian and a vehicle get close. A picture of the app is shown in Figure 15.3.

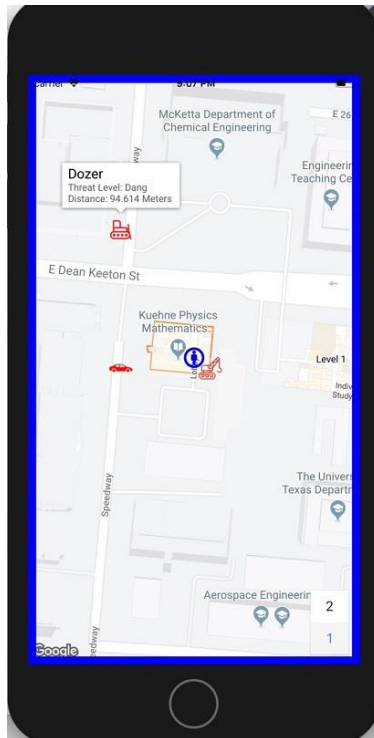


Figure 15.3: Image of the developed app showing a bird's eye view of the demo area

Road construction area

We designed and developed a simulated road construction area, including vehicles, workers, barricades, barriers, cones, a big container acting as an obstacle limiting the FOV of both drivers and pedestrians, and other objects usually present in this type of environments. An illustration of the design and a picture of the final deployment can be seen in Figure 15.4 and Figure 15.5.

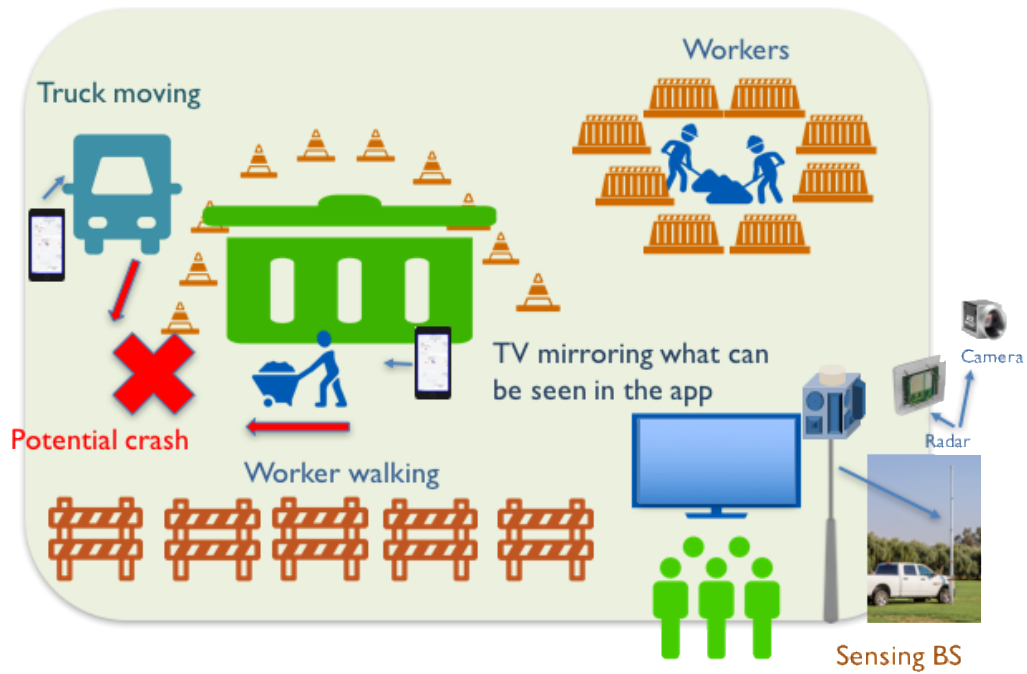


Figure 15.4: Designed simulated road construction area



Figure 15.5: Deployed road construction area

15.1.3 Presentation of the Demo

The demo starts with two actors working between the barricades. A car ready to drive around the fake road work area and behind the container that blocks the view of any potential pedestrian or driver. One of the workers has an iPhone with the bird's eye view app working. The driver has an iPad with the bird's eye view app working.

The images and sounds displayed in the bird's eye view app are mirrored in a TV screen so that the people attending the demo can see and listen the warning images as they appear.

1. The car goes for a first round and nothing happens. Workers keep working between the barricades. TV is mirroring the iPhone screen. Camera and radar data are shown. The app shows the car moving but no warning sign or sound.
2. Second round for the car. One of the workers gets out of the secure area between the barricades and walks with the wheelbarrow towards the large storage box, which blocks the field of view for him and for the car driver. When the car and the worker are getting too close, the app starts beeping and showing in red the icons for the car and the worker. The worker stops, and the car slows down a lot. The worker crosses and then goes back quickly to the secure area.

A video of the demo is available upon request.

15.2 Application 2: Pedestrian Crossing

The in-car pedestrian warning system (Figure 15.6) directly uses the detection methods tested in Chapter 14. Pedestrian collision warning is a simpler decision to make than many other safety warnings because pedestrians do not move at high speeds and are very unlikely to have injury-causing collisions with the side or rear of the host vehicle. Only pedestrians towards the front of the vehicle, and within 20 meters to either side, are detected tracked at any time. We specifically detect and track pedestrians and cyclists, and not other nearby objects. This warning system will be more valuable when the vehicle is expected to come near many stationary objects (for example, vehicles in a tight work zone). The current detection method is shallow camera/LIDAR fusion. A visual display is used to show the pedestrian detections while an audio display gives warnings. Warnings are given when moving at the current speed may result in a collision within 2 seconds. The vehicle's GPS is used to determine current speed, though for simpler live demonstrations the vehicle may be kept stationary with an imaginary speed.

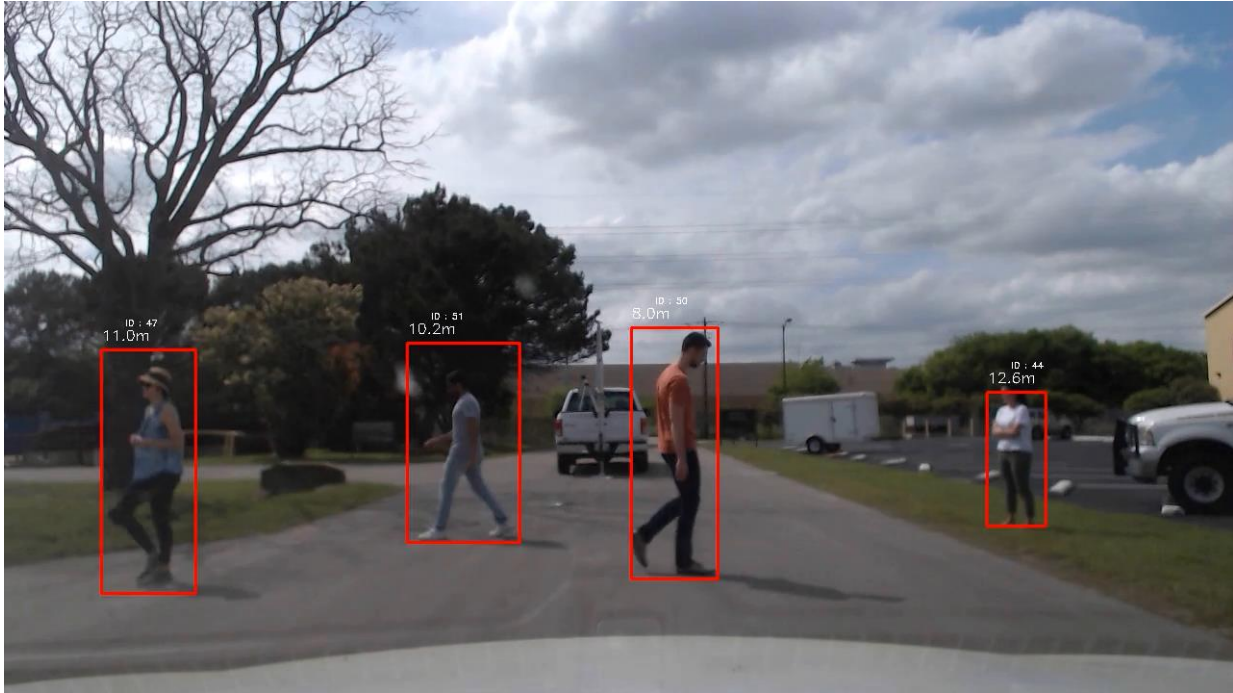


Figure 15.6: Image seen on computer screen during live demonstration of pedestrian crossing

15.3 Application 3: Highway Merging

In this application, a stopped vehicle is waiting to enter a high-speed road (for example, while exiting a construction zone). It uses V2V communication to detect oncoming vehicles and warns the driver when it not safe to merge.

Merging onto a highway from a construction zone is dangerous because of the time required for a slow, large vehicle to reach highway speed from a standstill. A fast-moving car from over one hundred meters away at the start of the merge could collide with the large vehicle if it is inattentive. We designed a merge warning system for the merging vehicle based on V2V. All vehicles determine their own positions via GPS and broadcast them using the standard Basic Safety Message. This message can easily be received from high distances, and GPS's possible positioning error of several meters is not critical when such large distances and speeds are considered.

Warnings could be generated based on the approaching vehicle's speed and the merging vehicle's assumed merging acceleration, but this is dependent on said acceleration, which may vary per vehicle and doesn't capture odder situations, such as where the approaching vehicle is slow but accelerating. Additionally, this warning criterion was found too lax by test users. Instead, the warning is based on the human driver's limitations—it is assumed that humans have difficulty judging the distance, speed, and perhaps even presence of vehicles beyond one hundred meters. A vehicle that will be within 100 meters of the merging vehicle within 2 seconds generates a warning. The warning continues until the vehicle passes the driver—if the other vehicle stops or is somehow not a threat, the driver is assumed to be capable of judging this and ignoring the warning. It is also

possible that more than one vehicle is approaching. There is a unique sound every time a new vehicle reaches the warning zone, so that each approaching vehicle will be noticed by the driver.

Two visual displays were created, one that simply alternates between several images and one that shows a bird's eye view of the road with positions of the detected vehicles. The former is less distracting and ensures that the driver will check to see the approaching vehicle on his own, while the latter provides more information and so may be trusted more. Figure 15.7 shows the first visual display, while Figure 15.8 shows the second.

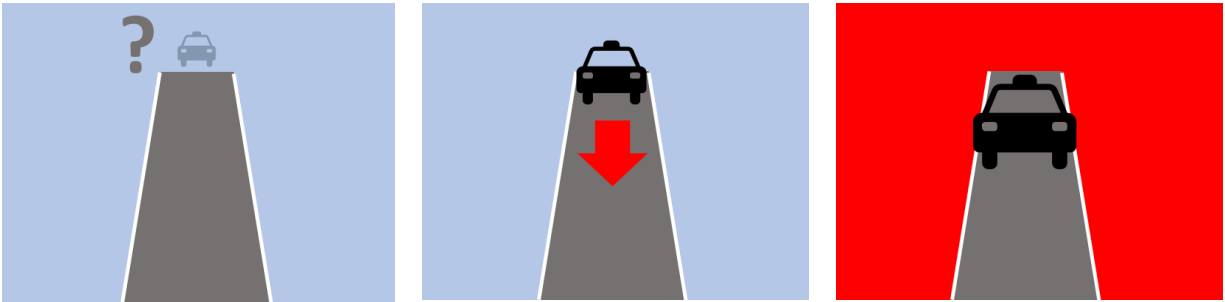


Figure 15.7: Simplified images for highway merge demo



Figure 15.8: More informative visual display that overlays approaching vehicles on the road

The images shown in Figure 15.7 represent the simple visual display for a safe situation (no vehicle in danger zone yet), a warning situation (at least one vehicle is approaching), and a critical situation (the vehicle has started merging but another vehicle is close behind). The latter will not be shown in the demonstration because purposefully reaching such a situation is difficult without jeopardizing pedestrians' safety.

V2V functioned reliably for this demonstration during most tests, but failed in one demonstration when many audience members gathered around the merging vehicle. This coincides with a conclusion of the intersection data analysis: vehicular communication is more robust to occlusion than sensors are, but not immune to it. A designated merging zone on a highway should be designed so that no large objects are placed between a merging vehicle and the oncoming traffic.

Chapter 16 Conclusion

The last twenty years have introduced many technologies for advanced driver assistant systems. This project used sensing technology from current ADAS, sensing technology common in self-driving vehicles, existing wireless communication, and next-generation communication. The relative strengths of these technologies were examined, in terms of their reliability in different applications and the cost or difficulty of implementing them. As was expected, each technology was found to have unique value in certain applications. Additionally, often a combination of several devices or techniques provided the best solution.

In this project the CTR research team developed and installed ADAS software, completing proof-of-concept tests for each of the five ADAS applications covered. Actual deployment of these applications will require minor design considerations and more rigorous testing, but should start with the devices and software given. In addition to describing the ADAS systems, this report contains qualitative and quantitative analysis of technologies or techniques that were not considered as useful for specific applications. Both the current applications and the comparative analyses should be valuable to the development of new ADAS technology.

Notable discoveries are briefly noted here. Accurate calibration of each sensor's position is difficult when devices may be placed differently or imprecisely in each car. A satisfactory automated calibrator has not been found, so the recommended approach is to design software that enables quick manual calibration for each device installation. Finally, cameras are useful but rely on powerful computers. It is recommended that cameras are used primarily when different types of objects must be treated differently, as in the pedestrian application, or as downward-facing sensors on tall infrastructure where simpler visual detection techniques are reliable. Tests of the intersection application suggest that forward-facing sensors can't detect vehicles from the side in enough time to prevent collisions. All-directional LIDAR can detect vehicles approaching from the side when the view is unobstructed. V2V communication is also affected by large obstructions between two vehicles, but not enough to fail to warn in the test locations. Tests of the overtaking application indicate that vehicular communication is the only appropriate tool for this purpose. Lastly, the results from the pedestrian detection application show that merging information from camera and LIDAR provides the most reliable solution. The algorithm is still imperfect but performs well on relatively slow-moving vehicles.

This project explored mmWave communication as an emerging option for traffic safety applications, as it enables high data rate communication between vehicles and/or infrastructure. Most of the research focused on overcoming practical difficulties of implementing mmWave communication on vehicles and proving its viability with a real prototype, which leveraged equipment from a variety of sources. It was found that using sensor information, the overhead associated with setting up a millimeter-wave communication link could be reduced substantially making millimeter-wave more viable for transportation applications. The potential of a joint communicating and sensing device has also been demonstrated through simulations and physical tests. Measurements were made to establish how a millimeter wave communication waveform can

also be used for radar sensing, reducing the potential hardware required on vehicles. The prototypes created are not ready for public use but significantly advance the research of high-bandwidth communication technologies for transportation applications.

Following is a summary of the results for Phase 2 of this project:

- *Task: Real-time fusion of V2V communications and radar*
 - Subtask – Creating a fusion prototype
 - Much of the material in this report falls under this subtask.
 - Subtask – Exploring other types of information fusion
 - Both high-level and low-level fusion of sensors was performed in several ways. One application, the pedestrian-only collision warning, relies fundamentally on the combination of camera and radar/LIDAR information. Sensor calibration was an unanticipated but unavoidable subproblem of sensor fusion.
 - Subtask – Incorporating the presence/absence of measurements - imputation
 - This is performed whenever necessary, sometimes inherently by the perception framework and sometimes explicitly. Examples in our current V2V-based tracking include the multiple categories of tracking when communication has just been established, the long-term storage of vehicle heading when stopped at an intersection, the setup of occlusion for objects.
- *Task: Distributed decision-making in complex, realistic scenarios*
 - Several modifications of the original distributed decision-making technique for cooperative driving are designed and tested. Further improvements are needed before kind of technology is user-safe, but in concept it has been shown to be robust to occasional input error and capable of making decisions that seem reasonable to a human driver.
- *Task: Exploring additional sources of information*
 - Subtask – Storing and making use of driver history
 - Driver history was examined as part of the more complex prediction methods in Section 6.1.1. It was not considered necessary for any of the examined safety applications.
 - Subtask – Acquiring and using location context
 - Location context was ultimately considered an application-specific tool, given that different applications have very different requirements for location context. The degree of context for each application is discussed.
 - Subtask – Utilizing Non-BSM messaging in DSRC
 - We have implemented a version of V2V tracking in which each vehicle imputes its speed, acceleration, heading, and turning rate using self-tracking algorithms and then communicates this information, along with

the standard position reading, to all other vehicles. The construction zone application uses infrastructure to communicate nearby object positions and a warning message. Details of non-BSM messaging such as standardization and security are not a focus of this project.

- *Task: DSRC-aided mmWave vehicular communication*
 - Subtask – DSRC-aided mmWave communication between cars
 - Leveraging DSRC information, an efficient beam alignment method for mmWave vehicular communication was proposed. The developed algorithms use machine learning to help predict preferred mmWave communication beams.
 - An efficient method to estimate mmWave channels by using the similar property between sub-6GHz (DSRC) and mmWave channels is described.
 - Subtask – RSU-aided mmWave communication between cars
 - Using sensors on infrastructure, an algorithm for detecting and localizing the vehicle was tested. This information can be broadcast from an RSU to vehicles to configure mmWave communication links quickly.
 - Leveraging position information (extracted by the RSU), our proposed adaptive-beamforming algorithm outperforms other conventional methods in mmWave vehicular systems.
 - Subtask – DSRC-aided mmWave communication proof-of-concept prototype
 - A proof of concept prototype has been created and tested in simple environments.
- *Task: Radar-aided mmWave communication*
 - Subtask – Using radar at the infrastructure to assist mmWave communication between cars
 - Exploiting the similarity of covariance between mmWave radar and mmWave communication channels, our proposed method can significantly reduce the overhead required for mmWave link configuration without a non-trivial loss compared to full beam search.
 - Subtask – Radar-aided mmWave proof-of-concept prototype with car-mounted radar & infrastructure-mounted radar
 - The radar returns of the mmWave prototype were examined and used to enable forward vehicle detection. In this way, the radar is able to provide important side information that can help the communication channel.

- *Task: Advanced technique to further enhance joint mmWave radar and communication system*
 - Subtask – Modify IEEE 802.11ad for better radar and communication performance
 - Several efficient mmWave channel acquisition methods have been described: 1) exploiting channel sparsity to reduce training overhead in frequency selective channels; 2) leveraging information extracted from sub-6 GHz communication systems to aid beam selection; and 3) using subspace estimation techniques to acquire information about channel main directions. These methods enhance the conventional approach of IEEE 802.11ad to advance radar and communication systems.
 - Subtask – Designing new joint waveforms and signal processing algorithms
 - We developed new waveforms that enhance the waveforms used in IEEE 802.11ad with improvements to allow more accurate velocity estimation. The key idea was to insert additional training signals into the communication waveform to better measure doppler shifts and thus velocity. Subtask – Prototyping a combined communication and radar car-to-car link
 - An mmWave automotive radar based on the mmWave WLAN standard to enable a joint vehicular communication-radar system at 60 GHz has been developed. This provides a suitable baseline for designing future vehicular communication standard that can be used for automotive radar applications.
 - A preliminary mmWave test bed to prototype mmWave joint communication and radar system was completed. Based on that platform, it was confirmed that the theoretical predictions of high data rate and accurate radar sensing match what is observed in practice, at least in simple laboratory settings.
 - Subtask – Field tests
 - Tests were performed for the mmWave joint communication and radar system outdoors in static settings, and in the lab in more dynamic configurations using a remotely controlled model car to move a target. The results confirm that the core theoretical principle is valid, and that this line of research is promising for future applications of mmWave to vehicles.
- *Task: Experimental design, prototype equipment installation and data collection, and optimizations of algorithms*
 - Subtask – Prototype equipment installation and data collection
 - Two vehicles have been equipped with sensing and communication (DSRC) equipment, as well as computers for data processing.

- Data was collected and processed for the intersection, overtaking, and pedestrian applications. Each involved multiple locations and scenarios according to a predetermined experimental design.
 - Live tests were performed for the pedestrian, highway merge, and work zone applications. This involved designed test areas and test direct or indirect users.
 - Subtask – Optimization of ADAS algorithms for different scenarios
 - Results based on the data collected in urban intersections, overtaking maneuvers and pedestrian detection scenarios have been used to guide improvements in the ADAS. Live tests performed on mock work zones and highway merge points were also
- *Task: Performance evaluation of the ADAS and expansion of CONOPS*
 - Subtask – Performance evaluation of the ADAS
 - Modeling of the system’s performance on specific applications or substeps was used to choose appropriate technologies or pinpoint improvements required by the system. The overall state of the system is described here and in the CONOPS document.

References

- [2.1] <https://axleaddict.com/safety/Deaf-People-Drive>
- [2.2] S. Hasirlioglu, A. Riener, W. Huber, and P. Wintersberger. “Effects of Exhaust Gases on Laser Scanner Data Quality at Low Ambient Temperatures”. *IEEE Intelligent Vehicles Symposium (IV)*, 2017.
- [2.3] Global Positioning System (GPS) Standard Positioning Service (SPS) Performance Analysis Report (2014). Federal Aviation Administration, Washington, DC 410.
- [2.4] D. Douchamps. “A small list of IMU/INS/INU.” <https://damien.douxchamps.net/research/imu/>
- [2.5] F. Lambert. “First look at Tesla’s latest Autopilot (2.5) computer in Model 3, S, and X vehicles.” Web article, *Eletrek*, 2018. <https://electrek.co/2018/03/28/tesla-autopilot-2-5-computer-model-3-s-x-first-look>
- [2.6] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado et al. “Tensorflow: Large-scale machine learning on heterogeneous distributed systems.” <http://tensorflow.org/>, Software available from tensorflow.org.
- [2.7] <https://motchallenge.net/data/MOT17>
- [2.8] <https://www.udacity.com/self-driving-car>
- [2.9] comma.ai <https://github.com/commaai/openpilot>
- [2.10] <http://apollo.auto/>
- [2.11] Nvidia “Driveworks.” <https://developer.nvidia.com/driveworks>
- [3.1] Särkkä, Simo. “Bayesian filtering and smoothing.” Vol. 3. Cambridge University Press, 2013.
- [3.2] Martin Liggins, et al. “Handbook of Multisensor Data Fusion: Theory and Practice, Second Edition.” CRC Press, 2017.
- [3.3] Reuter, Stephan. 2014. “Multi-Object Tracking Using Random Finite Sets.” Universität Ulm. <https://doi.org/10.18725/OPARU-3204>.
- [3.4] Williams, Jason L. "Hybrid Poisson and multi-Bernoulli filters." *Information Fusion (FUSION)*, 2012 15th International Conference on. IEEE, 2012.
- [3.5] Björklund, Andreas, et al. “On Evaluation of Permanents.” arXiv [cs.DS], 21 Apr. 2009.

- [3.6] Williams, J., and R. Lau. “Approximate Evaluation of Marginal Association Probabilities with Belief Propagation.” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 4, Oct. 2014, pp. 2942–59.
- [3.7] Crouse, David F., et al. “The JPDAF in Practical Systems: Computation and Snake Oil.” *Signal and Data Processing of Small Targets*, 2010, vol. 7698, International Society for Optics and Photonics, 2010, p. 769813.
- [3.8] García-Fernández, Ángel F., et al. “Poisson Multi-Bernoulli Mixture Filter: Direct Derivation and Implementation.” arXiv [cs.CV], 13 Mar. 2017.
- [3.9] Scheel, A., et al. “Vehicle Tracking Using Extended Object Methods: An Approach for Fusing Radar and Laser.” *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 231–38.
- [3.10] F. Kunz, D. Nuss, J. Wiest, H. Deusch, S. Reuter, F. Gritschender, A. Scheel, M. Stubler, M. Bach, P. Hatzelmann, and others. “Autonomous driving at Ulm University: A modular, robust, and sensor-independent fusion approach.” *IEEE IV*, 2015.
- [3.11] A. K. Gostar, R. Hoseinnezhad, and A. Bab-Hadiashar. “Sensor control for multi-object tracking using labeled multi-Bernoulli filter.” *IEEE FUSION*, 2014.
- [3.12] F. Papi and D.Y. Kim. “A particle multi-target tracker for superpositional measurements using labeled random finite sets.” *IEEE Transactions on Signal Processing*, 2015.
- [3.13] Petrov, N., et al. “Rectangular Extended Object Tracking with Box Particle Filter Using Dynamic Constraints.” *IET Conference on Data Fusion Target Tracking 2014: Algorithms and Applications (DF TT 2014)*, 2014, pp. 1–7.
- [3.14] Broßbeit, P., et al. “Probabilistic Rectangular-Shape Estimation for Extended Object Tracking.” *2016 IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 279–85.
- [3.15] Cho, H., et al. “A Multi-Sensor Fusion System for Moving Object Detection and Tracking in Urban Driving Environments.” *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1836–43. L-shape lidar
- [3.16] Wyffels, K., and M. Campbell. “Negative Information for Occlusion Reasoning in Dynamic Extended Multiobject Tracking.” *IEEE Transactions on Robotics*, vol. 31, no. 2, Apr. 2015, pp. 425–42.
- [3.17] Granström, K., et al. “A Multiple Model PHD Approach to Tracking of Cars under an Assumed Rectangular Shape.” *17th International Conference on Information Fusion (FUSION)*, 2014, pp. 1–8.
- [3.18] Lamard, L., et al. “Dealing with Occlusions with Multi Targets Tracking Algorithms for the Real Road Context.” *2012 IEEE Intelligent Vehicles Symposium*, 2012, pp. 371–76.
- [4.1] <https://nationalmap.gov/>

- [4.2] <https://developers.google.com/maps/documentation/>
- [4.3] Moosmann, F., and C. Stiller. "Joint Self-Localization and Tracking of Generic Objects in 3D Range Data." *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 1146–52.
- [4.4] Cadena, C., et al. "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age." *IEEE Transactions on Robotics*, vol. 32, no. 6, Dec. 2016, pp. 1309–32.
- [4.5] Levinson, J., and S. Thrun. "Robust Vehicle Localization in Urban Environments Using Probabilistic Maps." *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 4372–78.
- [4.6] Mur-Artal, Raul, Montiel, J. M. M. and Tardos, Juan D. "ORB-SLAM: a Versatile and Accurate Monocular SLAM System." *IEEE Transactions on Robotics*, vol. 31, no. 5, 2015.
- [4.7] B. Mishra, S.K. Panigrahy, T.C. Tripathy, D. Jena, and S.K. Jena. "A secure and efficient message authentication protocol for VANETs with privacy preservation." *IEEE WICT*, 2011.
- [4.8] M. Ester, H. P. Kriegel, J. Sander, and X. Xu. "A density-based algorithm for discovering clusters in large spatial databases with noise." In *KDD*, vol. 96, no. 34, pp. 226-231. 1996.
- [4.9] Scheel, Alexander, and Klaus Dietmayer. "Tracking Multiple Vehicles Using a Variational Radar Model." arXiv preprint arXiv:1711.03799 (2017).
- [4.10] J. Redmon, and A. Farhadi. "YOLO9000: better, faster, stronger." arXiv preprint arXiv:1612.08242 (2016).
- [4.11] <http://mscoco.org/>
- [4.12] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018).
- [4.13] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado et al. "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." <http://tensorflow.org/>, Software available from tensorflow.org.
- [4.14] OpenCV Version 3.10. "Camera Calibration." Feb 2015. https://docs.opencv.org/3.1.0/dc/dbb/tutorial_py_calibration.html
- [4.15] Magnier, Valentin, Dominique Gruyer, and Jerome Godelle. "Automotive LIDAR objects detection and classification algorithm using the belief theory." *Intelligent Vehicles Symposium (IV)*, 2017 IEEE. IEEE, 2017.
- [4.16] Mioulet, Luc, et al. "Efficient combination of Lidar intensity and 3D information by DNN for pedestrian recognition with high and low density 3D sensor." *Intelligent Vehicles Symposium (IV)*, 2017 IEEE. IEEE, 2017.

- [4.17] Scheel, A., et al. "Vehicle Tracking Using Extended Object Methods: An Approach for Fusing Radar and Laser." *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 231–38.
- [4.18] Cho, H., et al. "A Multi-Sensor Fusion System for Moving Object Detection and Tracking in Urban Driving Environments." *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1836–43. L-shape lidar
- [4.19] Kampker, Achim, et al. "Towards Multi-Object Detection and Tracking in Urban Scenario under Uncertainties." arXiv, 8 Jan. 2018, <http://arxiv.org/abs/1801.02686>. arXiv.
- [4.20] Hasirlioglu, Sinan, et al. "Effects of exhaust gases on laser scanner data quality at low ambient temperatures." *Intelligent Vehicles Symposium (IV)*, 2017 IEEE. IEEE, 2017.
- [4.21] Texas Hill Country, "Lovebugs Are Everywhere, but They're Only 'Harmful' to Your Car." August 25, 2017. <http://texashillcountry.com/lovebugs-everywhere-car>
- [5.1] Scheel, A., et al. "Vehicle Tracking Using Extended Object Methods: An Approach for Fusing Radar and Laser." *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 231–38.
- [5.2] Cho, H., et al. "A Multi-Sensor Fusion System for Moving Object Detection and Tracking in Urban Driving Environments." *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1836–43. L-shape lidar
- [5.3] Stiller, C., et al. "Multisensor Obstacle Detection and Tracking." *Image and Vision Computing*, vol. 18, no. 5, Apr. 2000, pp. 389–96.
- [5.4] Vo, Ba Ngu, and Ba Tuong Vo. "Multi-sensor multi-object tracking with the generalized labeled multi-Bernoulli filter." arXiv preprint arXiv:1702.08849 (2017).
- [5.5] Kim, Taewan, and Joydeep Ghosh. "Robust detection of non-motorized road users using deep learning on optical and LIDAR data." *Intelligent Transportation Systems (ITSC)*, 2016 IEEE 19th International Conference on. IEEE, 2016.
- [5.6] Liu, Wei, et al. "Ssd: Single shot multibox detector." *European conference on computer vision*. Springer, Cham, 2016.
- [5.7] Scheidegger, Samuel, et al. "Mono-Camera 3D Multi-Object Tracking Using Deep Learning Detections and PMBM Filtering." arXiv [cs.CV], 27 Feb. 2018
- [5.8] Tensorflow, "Tensorflow detection model zoo." https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md
- [5.9] R. Unnikrishnan, and M. Hebert, "Fast extrinsic calibration of a laser rangefinder to a camera," Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-05-09, July 2005. <http://www.cs.cmu.edu/~ranjith/lcct.html>

- [5.10] A. Geiger, L. Philip, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite." *Computer Vision and Pattern Recognition (CVPR)*, 2012 IEEE Conference on. IEEE, 2012
- [5.11] J. Castorena, U. S. Kamilov, and P. T. Boufounos. "Autocalibration of LIDAR and optical cameras via edge alignment." *In Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pp. 2862-2866. IEEE, 2016.
- [5.12] H. J. Chien, R. Klette, N. Schneider, and U. Franke. "Visual odometry driven online calibration for monocular LiDAR-camera systems." *In Pattern Recognition (ICPR), 2016 23rd International Conference on*, pp. 2848-2853. IEEE, 2016.
- [5.13] N. Schneider, F. Piewak, C. Stiller, & U. Franke, "RegNet: Multimodal Sensor Registration Using Deep Neural Networks." *Intelligent Vehicles Symposium (IV)*, 2017, IEEE
- [7.1] Forbes. "The Uber Self-Driving Car Crash: What Really Happened." <https://www.forbes.com/sites/meriameriboucha/2018/05/28/uber-self-driving-car-crash-what-really-happened/>. May 2018
- [7.2] <https://motchallenge.net/data/MOT17>
- [7.3] M. Motro, T. Kim, R. Kalantari, J. Ghosh, J. Choi, A. Ali, P. Kumari et al. "Communications and Radar-Supported Transportation Operations and Planning." No. FHWA/TX-16/0-6877-1. 2017.
- [7.4] M. Motro, J. Ghosh, and C. Bhat. "Optimal alarms for vehicular collision avoidance." IEEE IV, 2017. <https://arxiv.org/pdf/1708.04922.pdf>
- [7.1] Wolpert, David H., and Kagan Tumer. "An introduction to collective intelligence." arXiv preprint cs/9908014 (1999).
- [7.2] Kalantari, Rahi, et al. "A distributed, collective intelligence framework for collision-free navigation through busy intersections." *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016.
- [7.3] S. Acharyya and J. Ghosh. "MEMR: A Margin Equipped Monotone Retargeting Framework for Ranking." UAI, 2014.
- [7.4] C. Painter-Wakefield and R. Parr. "Greedy algorithms for sparse reinforcement learning." arXiv preprint arXiv:1206.6485 (2012).
- [7.5] A. Hashemi and H. Vikalo. "Sparse linear regression via generalized orthogonal least-squares." *Signal and Information Processing (GlobalSIP), 2016 IEEE Global Conference on*. IEEE, 2016.
- [7.6] Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." *Nature* 518.7540 (2015): 529.

- [7.7] Tampuu, Ardi, et al. "Multiagent cooperation and competition with deep reinforcement learning." *PloS one* 12.4 (2017): e0172395.
- [7.8] Foerster, Jakob, et al. "Learning to communicate with deep multi-agent reinforcement learning." *Advances in Neural Information Processing Systems*. 2016.
- [7.9] Karkus, Peter, David Hsu, and Wee Sun Lee. "Qmdp-net: Deep learning for planning under partial observability." *Advances in Neural Information Processing Systems*. 2017.
- [7.10] Pendrith, Mark D., and Claude Sammut. "On reinforcement learning of control actions in noisy and non-Markovian domains." University of New South Wales, School of Computer Science and Engineering, 1994.
- [7.11] Littman, Michael L., Anthony R. Cassandra, and Leslie Pack Kaelbling. "Learning policies for partially observable environments: Scaling up." *Machine Learning Proceedings 1995*. 1995. 362-370
- [7.12] Fox, Roy, Ari Pakman, and Naftali Tishby. "Taming the noise in reinforcement learning via soft updates." *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2016.
- [8.1] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta and P. Popovski, "Five disruptive technology directions for 5G," in *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74-80, February 2014.
- [8.2] 5G PPP Pre-standardisation Working Group, "5G-PPP Pre-standards-WG Issues Paper", https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-Pre-standards-WG-Issues-Paper-for-20-Oct-15-WS_final-edited.pdf, October 2015.
- [8.3] 3GPP, "3rd Generation Partnership Project; Technical Specification Group Radio Access Network; NR; NR and NG-RAN Overall Description; Stage 2 (Release 15)", 3GPP TS 38.300 V15.3.1, October 2018.
- [8.4] J. B. Kenney, "Dedicated Short-Range Communications (DSRC) Standards in the United States," *Proceedings of the IEEE*, pp. 1162-1182, 2011.
- [8.5] Remcom, "Wireless InSite," [Online]. Available: <https://www.remcom.com/wireless-insite>. [Accessed August 2018].
- [8.6] NIST, "5G mmWave Channel Model Alliance," [Online]. Available: <https://www.nist.gov/ctl/5g-mmwave-channel-model-alliance>. [Accessed August 2018].
- [8.7] V. Va, T. Shimizu, G. Bansal and R. W. Heath Jr., "Millimeter Wave Vehicular Communications: A Survey," *Foundations and Trends® in Networking*, vol. 10, no. 1, pp. 1-113, 2016.

- [9.1] N. Garcia, H. Wymeersch, E. G. Strom and D. Slock, "Location-aided mm-wave channel estimation for vehicular communication," in *SPAWC*, 2016.
- [9.2] F. Athley, "Threshold region performance of maximum likelihood direction of arrival estimators," *IEEE Tran. Signal Process.*, vol. 53, pp. 1359-1373, 2005.
- [9.3] *J2735 Dedicated Short Range Communications (DSRC) Message Set Dictionary*, 2009.
- [9.4] Remcom, "Wireless InSite," [Online]. Available: <https://www.remcom.com/wireless-insite>. [Accessed August 2018].
- [9.5] M. Giordani, M. Polese, A. Roy, D. Castor and M. Zorzi, "A Tutorial on Beam Management for 3GPP NR at mmWave Frequencies," *arXiv preprint arXiv:1804.01908*, 2018.
- [9.6] S. Park and R. Heath Jr., "Spatial channel covariance estimation for mmWave hybrid MIMO architecture," in Proc. Asilomar Conf. Signals, Syst. Comput. (ASILOMAR), Asilomar, 2016.
- [9.7] S. Jaeckel, L. Raschkowski, K. Börner and L. Thiele, "QuaDRiGa: A 3-D Multi-Cell Channel Model With Time Evolution for Enabling Virtual Field Trials," in *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 6, pp. 3242-3256, June 2014.
- [9.8] *INRAS Radarbook*. <http://www.inras.at/en/products/radarbook.html>
- [9.9] *INRAS Radarlog*. <http://www.inras.at/en/products/new-radarlog.html>
- [9.10] T. Trump and B. Ottersten, "Estimation of nominal direction of arrival and angular spread using an array of sensors," *IEEE Tran. Signal Process.*, vol. 50, pp. 57-69, 4 1996.
- [10.1] S. M. Patole, M. Torlak, D. Wang, and M. Ali, "Automotive radars: A review of signal processing techniques," *IEEE Signal Processing Magazine*, vol. 34, no. 2, pp. 22–35, March 2017.
- [10.2] P. Kumari, N. Gonzalez-Prelcic, and R. W. Heath Jr, "Investigating the IEEE 802.11ad Standard for Millimeter Wave Automotive Radar," in *Proceedings of the IEEE Vehicular Technology Conference*, September 2015, pp. 3587–3591.
- [10.3] P. Kumari, J. Choi, N. Gonzalez-Prelcic, and R. W. Heath, "IEEE 802.11ad-based radar: An approach to joint vehicular communication-radar system," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 3012–3027, April 2018.
- [10.4] J. Hasch, E. Topak, R. Schnabel, T. Zwick, R. Weigel, and C. Waldschmidt, "Millimeter-wave technology for automotive radar sensors in the 77 GHz frequency band," *IEEE Transactions on Microwave Theory and Techniques*, vol. 60, no. 3, pp. 845–860, 2012.
- [10.5] P. Kumari, D. H. N. Nguyen, and R. W. Heath, "Performance trade-off in an adaptive IEEE 802.11ad waveform design for a joint automotive radar and communication

system,” in Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Mar 2017, pp. 4281–4285.

[10.6] P. Kumari, M. E. Eltayeb, and R. W. Heath, “Sparsity-aware adaptive beamforming design for IEEE 802.11ad based joint communication-radar,” in Proceedings of the IEEE Radar Conference (RadarConf), March 2018, pp. 4281–4285.

[10.7] R. Rajamaki and V. Koivunen, “Comparison of sparse sensor array configurations with constrained aperture for passive sensing,” in Proc. IEEE Radar Conference (RadarConf), May 2017, pp. 0797–0802.

[10.8] Cudak, M., Kovarik, T., Thomas, T., Ghosh, A., Kishiyama, Y., & Nakamura, “Experimental mm wave 5G cellular system”, in Proceedings of IEEE Globecom Workshops, pp. 377-381, Dec. 2014.

[11.1] Montgomery, D.C. 2013. *Design and analysis of experiments*. Eighth ed., John Wiley & Sons, Inc., Hoboken, NJ

[11.2] Cavazzuti, M., 2012. Optimization methods: from theory to design scientific and technological aspects in mechanics. Springer Science & Business Media.

[12.1] NACTO, 2013. Urban Street Design Guide. National Association of City Transportation Officials. Island Press, 2013.

[12.2] Institute of Transportation Engineers (ITE), 2017, "Unsignalized Intersection Improvement Guide." Link: <http://www.ite.org/uiig/types.asp>

[12.3] Fitzpatrick K., Wooldridge M.D., and Blaschke J.D., 2005a, Urban Intersection Design Guide. Volume 1 - Guidelines, Texas Transportation Institute Report.

[12.4] Fitzpatrick, K., Wooldridge, M.D., and Blaschke, J.D., 2005b. Urban Intersection Design Guide: Volume 2-Applications (No. FHWA/TX-05/0-4365-P2 Vol. 2).

[12.5] Harding, J., Powell, G., R., Yoon, R., Fikentscher, J., Doyle, C., Sade, D., Lukuc, M., Simons, J., Wang, J. (2014, August). Vehicle-to-vehicle communications: Readiness of V2V technology for application. (Report No. DOT HS 812 014). Washington, DC: National Highway Traffic Safety Administration.

[13.1] Brooks, R.M., 2012. Acceleration characteristics of vehicles in rural Pennsylvania. Analysis 1 (5), 5–22.

[15.1] Vaaraniemi, Mikael. “Usability Oriented Visualization Techniques for 3D Navigation Map Display.” Diss. Technische Universität München, 2014.

Appendix A – Vehicle Specifications

All sensors are connected to a deep-cycle battery for power, and all sensors except for the Reach are connected to a computer to store data. The Reach can store GNSS and IMU readings on its own hard drive and can be controlled wirelessly. The battery and the majority of cables and adapters reside in the back left seat of the car, while the computer is held by a person in the front passenger seat. This design avoids any permanent alteration to the vehicle while still having a (mostly) dedicated space for the electronics.

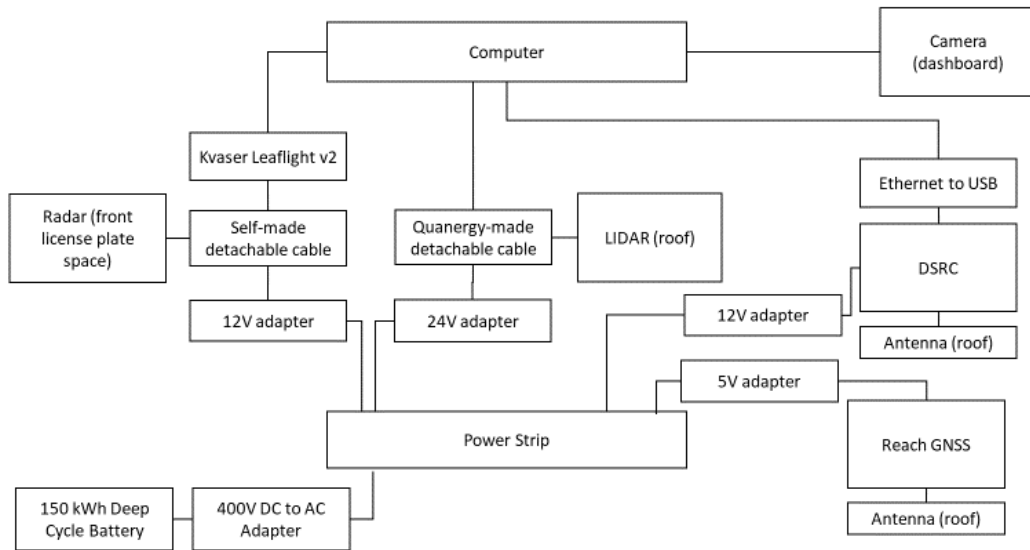


Figure A.1: Diagram of wire connections



Figure A.2: Picture of car with sensors annotated



Figure A.3: Closer picture of camera attachment to front window

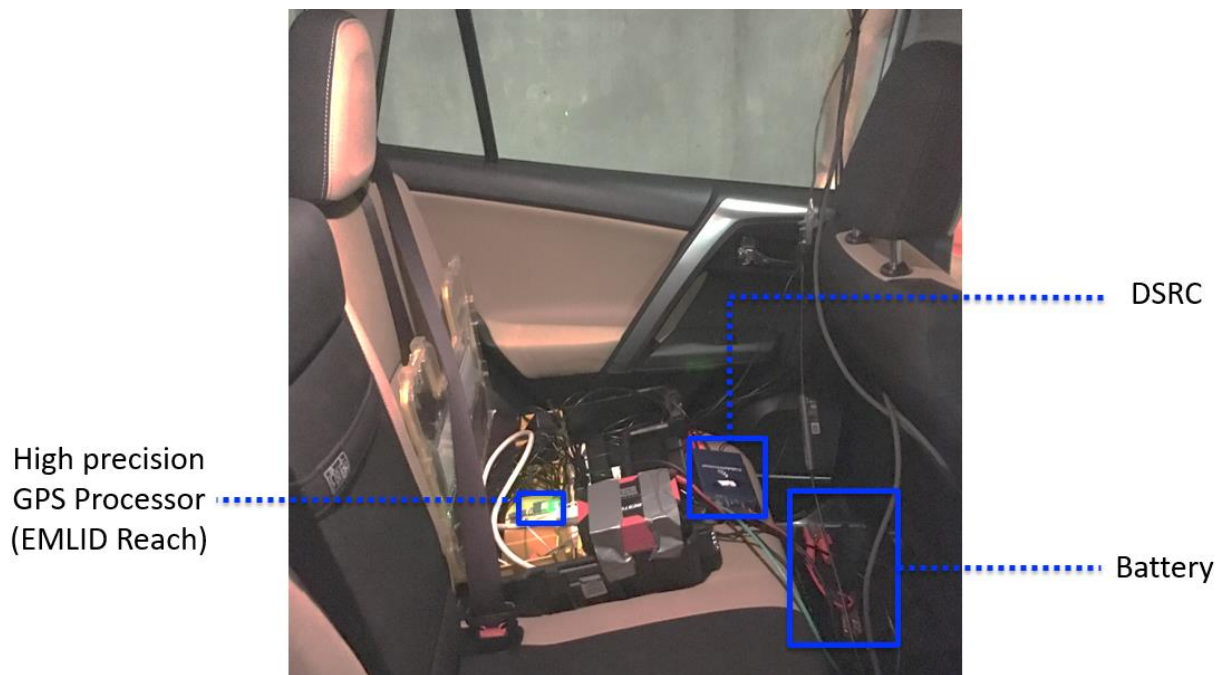


Figure A.4: Picture of adapters and connections in the back seat

Appendix B – Pedestrian Ground Truth

Certain values had to be determined to properly create ground truth:

- The time lag for each computer
- A rough elevation map of the ground
- The position and orientation of the tower camera

Inaccuracies in times reported by a computer can be significant when the computer is operating at high capacity, such as when it is storing large amounts of data, and when it does not have access to correcting information from internet servers. However, the Velodyne LIDAR has a GPS unit that reports precise time. First, the LIDAR and the car's video time is matched manually using video with LIDAR detections overlaid. The car's and tower's video times are matched manually by comparing snapshots of pedestrians from each. Snapshots taken 0.1 seconds apart from each camera are shown in Figures B.1 and B.2.

The elevation along each road is determined by averaging the car's elevation, as reported by GPS, at various positions on the road. The highest elevation was 0.02 radians or 1.1 degrees. Additionally, roads are generally slightly curved to ensure water runoff. This curve was approximated with a planar slope of 0.05 radians. The grass to the side of the road was sometimes heavily sloped. This slope was approximated but not accurately determined—thus, pedestrian positions off the road were determined less accurately.

The position and orientation of the tower camera were also determined using the car's GPS. As the car approached and retreated from the pedestrian zone at least six times at each location, many car positions in the tower's video and corresponding GPS readings were available. The object detection applied to the tower video also gave bounding boxes for the car, and the bottom corners of these boxes were matched to the GPS-determined corners of the car.

The solvePnP function of OpenCV solves for the camera pose that creates the best match. Given the camera's pose, the method for determining where a pedestrian is on the ground, using their position in an image, is described in Section 4.4.2. In that section it is used for on-vehicle detection, but the approach is the same—and in fact more accurate for a higher, downward-angled camera.



Figure B.1: Vehicle camera snapshots 0.1 seconds apart

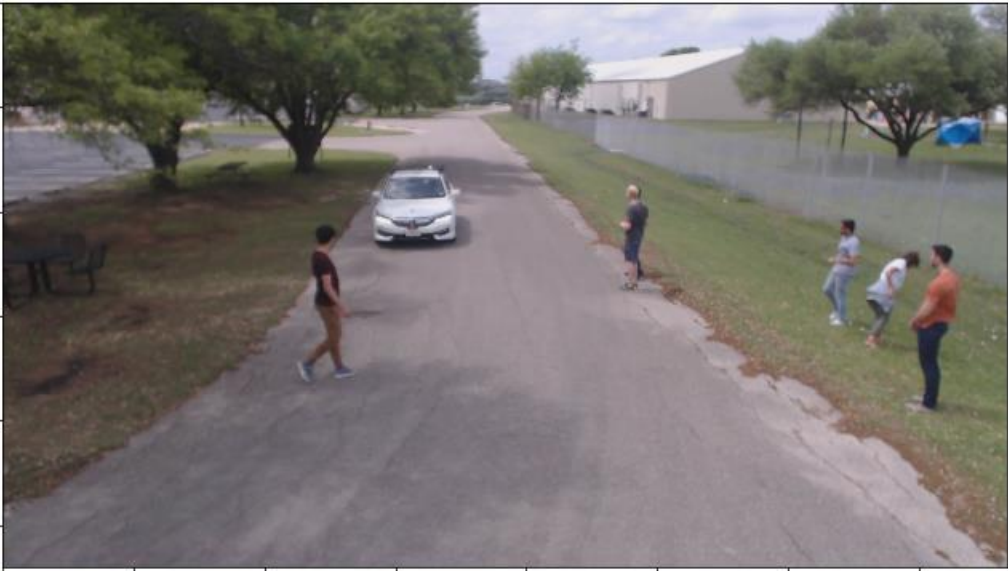
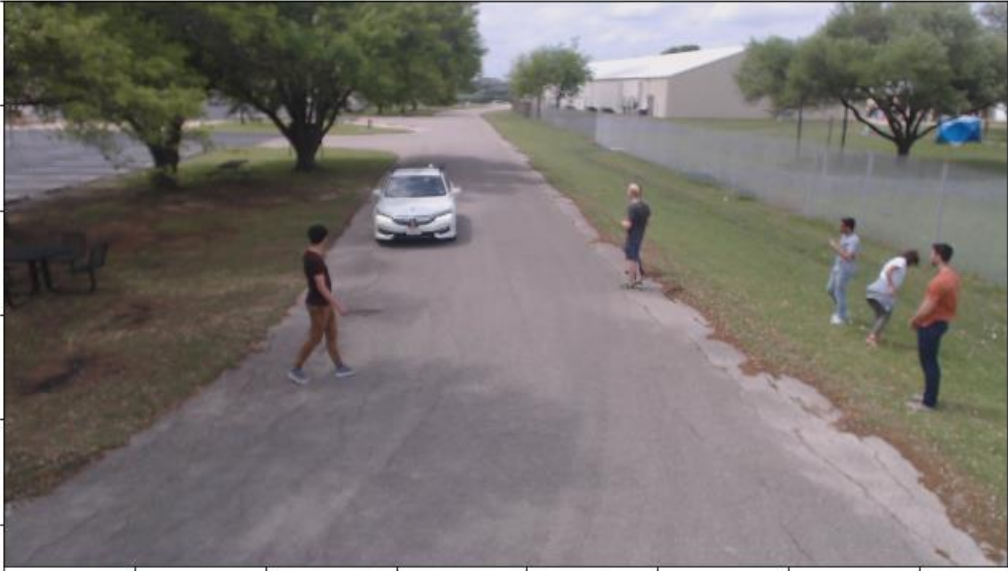


Figure B.2: Tower camera snapshots 0.1 seconds apart

Appendix C – Demonstration Videos

Video playlist 1: Radar tracking examples:

1A: <https://utexas.box.com/v/CarstopTM1vid1A>

1B: <https://utexas.box.com/v/CarstopTM1Vid1B>

1C: <https://utexas.box.com/v/CarstopTM1Vid1C>

Video playlist 2: Radar tracking examples, focusing on alternate test vehicle:

2A: <https://utexas.box.com/v/CarstopTM1Vid2A>

2B: <https://utexas.box.com/v/CarstopTM1Vid2B>

2C: <https://utexas.box.com/v/CarstopTM1Vid2C>

Video playlist 3: YOLO example:

3A: <https://utexas.box.com/v/CarstopTM1Vid3>

Video playlist 4: LIDAR tracking:

4A: <https://utexas.box.com/v/CarstopTM1Vid4A>

4B: <https://utexas.box.com/v/CarstopTM1Vid4B>

Video playlist 5: Radar-vision fusion:

5A: <https://www.youtube.com/watch?v=KZxb20Gupx0>