Vision and Radar Fusion for Identification of Vehicles in Traffic

Prakriti Banik

Thesis submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Dhruv Batra, Chair
Shane McLaughlin
Devi Parikh

July 1, 2015
Blacksburg, Virginia

Keywords: radar object validation, car detection, lead vehicle detection, radar-vision fusion
Copyright 2015, Prakriti Banik
Vision and Radar Fusion for Identification of Vehicles in Traffic
Prakriti Banik

ABSTRACT

This report presents a method for estimating the presence and duration of preceding and lead vehicle in front of a motorcycle using an object detection algorithm guided by radar data. The video and radar data were collected as part of a large transportation project. The data are recorded by the ego vehicle during a trip while in a naturalistic research study. The goal is to validate objects detected by radar using vision, to identify moving preceding vehicles and the lead vehicle. The proposed approach takes advantage of radar data in locating the vehicles and other targets and then validates the targets as vehicles using Dual-Tree Branch-and-Bound (Kokkinos, 2011) object detection algorithm. Localization, detection and tracking took 0.0385 seconds per frame on average. Precision and recall of lead vehicle detection is 98.61% and 90.53% respectively.

The algorithm presents a comprehensive approach to localize target vehicles in video. The radar object coordinates are mapped on the video frame using perspective projection mapping. Then persistent radar objects are determined by analyzing their trajectory on video frames. When a radar object appears for three consecutive frames, its called a persistent object. A region of interest (ROI) around the persistent radar object is cropped from the frame, and passed to the object detection algorithm to determine if the persistent object is a car. Once a car is detected the validation of the radar object is complete. We track the detected car in the following frames and refresh the detection after every fourteen frames. The car detection algorithm runs whenever a new persistent radar object is introduced. After validating radar objects, at each timestamp, the lead vehicle is determined using radar object’s forward and lateral distance. The time from detecting a lead vehicle to the time when the vehicle disappears or another vehicle becomes lead vehicle, is recorded to get the epochs of following driving mode for that lead vehicle. Finally, the detection result is integrated with MATLAB lane detection system to make a complete system for lead vehicle detection and tracking.

The video of interest has 240×720 resolution and approximately 15 frames per second. The car detection algorithm takes 0.1960 seconds on average to detect one car in a machine with Windows operating system and 4GB RAM. But as the detection algorithm is not run for each frame it saves time. Since no annotated motorcycle video dataset is publicly available, two videos of 52 seconds and 26 seconds were manually annotated to test the performance of the approach. The current approach works almost at real time. The algorithm has been tested and results have been reported on 1 video.
Acknowledgments

Without the guidance and support of my committee members, team members, friends and family, my research would not have been possible. To begin, I would like to express my sincere gratitude to Dr. Shane B. McLaughlin for his support, excellent guidance, encouragement, and providing me with a perfect atmosphere for doing research. I would also like to thank my advisor Dr. Dhruv Batra who guided and inspired me from beginning to end. I thank Dr. Devi Parikh for her valuable feedback and encouragement; she has always been a source of my motivation.

I would like to thank Stanislaw Antol, a close friend, who always provided me technical help. I also want to recognize everyone involved at Virginia Tech Transportation Institute Motorcycle Research Group for sharing their insights on the topic to improve my work. Special thanks also goes to Dr. Avijit Bakshi for his advice, discussions, and feedback for improving my thesis. Without his presence in my life I would not have accomplished this. Finally, I would like to thank my family, Prabhat Chandra Banik, Arati Banik, Priti Dutta, Bikramjit Dutta, and Parthib Dutta for their continuous love and encouragement.
# Contents

1 Introduction ............................................ 1
   1.1 Transportation Application ............................ 1
   1.2 Driving Modes ........................................ 2
   1.3 Vehicle Detection and Tracking ........................ 2

2 Literature Review ...................................... 5

3 Dataset .................................................... 8
   3.1 Naturalistic Motorcycle Dataset Details ................. 8
      3.1.1 Data Acquisition Hardware .......................... 8
      3.1.2 Radar Data Description ............................. 10
      3.1.3 Video Data Description ............................. 10
   3.2 Data Annotation ......................................... 10

4 Methodology .............................................. 15
   4.1 Mapping Radar Objects on Video Frame .................. 15
      4.1.1 Rigid Transformation ................................ 16
      4.1.2 Perspective Mapping Transformation ................ 17
   4.2 Validating Radar Objects using Trajectory and Car Detection ............... 19
      4.2.1 Determine Persistent Radar Objects .................. 19
      4.2.2 Car Detection using Dual-Tree Branch-and-Bound ........ 21
   4.3 Lead Vehicle Detection .................................. 25
4.4 Tracking the Detected Car ................................................. 28

5 Results ........................................................................ 29
  5.1 Persistent Radar Object Detection Results ...................... 29
    5.1.1 Qualitative Results of Valid Radar Object Detection .... 32
  5.2 Car Detection and Tracking Results ............................... 35
    5.2.1 Qualitative Results of Detection and Tracking using Vision 37
  5.3 Lead Vehicle Detection Results .................................... 41
    5.3.1 Preceding Vehicles Epoch Recording Results ............ 41

6 Conclusion ..................................................................... 43

Bibliography ..................................................................... 45
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Lead vehicle, preceding vehicle, and host vehicle illustration.</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>Driving Regimes and Headways (based on Forbes, 1972) S. B. McLaughlin. Measurement of driver preferences and intervention responses as influenced by adaptive cruise control deceleration characteristics, 1998, Used under fair use, 2015.</td>
<td>3</td>
</tr>
<tr>
<td>3.1</td>
<td>Illustration of main unit, forward camera and radar mounted on a motorcycle, S. B. McLaughlin, S. R. Fritz, S. L. Williams, and T. Buche. Overview of the MSF 100 motorcyclist naturalistic study, 2014, Used under fair use, 2015.</td>
<td>9</td>
</tr>
<tr>
<td>3.2</td>
<td>A video frame from a trip recorded by the forward camera, S. B. McLaughlin, S. R. Fritz, S. L. Williams, and T. Buche. Overview of the MSF 100 motorcyclist naturalistic study, 2014, Used under fair use, 2015.</td>
<td>11</td>
</tr>
<tr>
<td>4.1</td>
<td>Block diagram of the vehicle detection and tracking system.</td>
<td>15</td>
</tr>
<tr>
<td>4.2</td>
<td>(a) Radar object data representation (b) Camera coordinates (black) and radar coordinates (blue). $\phi$ is the angle between radar’s heading axis $W$ and camera’s lens axis $Z$.</td>
<td>16</td>
</tr>
<tr>
<td>4.3</td>
<td>The coordinate systems involved in mapping radar object on video frame.</td>
<td>18</td>
</tr>
<tr>
<td>4.4</td>
<td>One-dimensional profile of mapped radar object on frames for 2000 timestamps of a video from Train Dataset.</td>
<td>20</td>
</tr>
</tbody>
</table>
4.5 Trajectories of vehicles in motion and background from stacked horizontal-profiles along the time axis. The zig zag (red) is more likely the motorcycle moving back and forth or wiggling the radar at lower speed, Y. Tan, F. Han, J. Eledath, R. Kumar, and F. A.-K. Ibrahim. Radar guided vision system for vehicle validation and vehicle motion characterization, Jan. 15 2013. US Patent 8,355,539, Used under fair use, 2015. .......................... 21

4.6 (a) Height of cars vs. distance of the car from motorcycle (b) Width of cars vs. distance of the car from motorcycle .......................... 23

4.7 Four frames with cars from each of the four distance groups. (from top to bottom) a) car distance 4 meter, b) car distance 8 meter, c) car distance 14 meter, and d) car distance 24 meter, S. B. McLaughlin, S. R. Fritz, S. L. Williams, and T. Buche. Overview of the msf 100 motorcyclist naturalistic study, 2014, Used under fair use, 2015. ............................................ 24

4.8 Vertical profile of radar objects on video frame. This determines the road horizon .......................... 26

4.9 Example of final detection selection, when multiple detections are returned by the DTBB algorithm. A video frame with one persistent radar object (top), the ROI with multiple detections returned by DTBB (middle), and the final detection (bottom), S. B. McLaughlin, S. R. Fritz, S. L. Williams, and T. Buche. Overview of the MSF 100 motorcyclist naturalistic study, 2014, Used under fair use, 2015. .......................... 27

4.10 No lead vehicle even when preceding vehicle exists on the same lane .......................... 28

5.1 Precision-Recall curve for persistent radar object detection on Train dataset 31

5.2 ROC curve for persistent radar object detection on Train dataset 31

5.3 Confusion matrix for persistent radar object detection for different combinations of \( V \) and \( M \) on Train dataset. Class 1: correct radar objects; Class 2: incorrect radar objects; Class 3: duplicate radar objects with respect to frame number .......................... 32

5.4 Raw and persistent radar objects mapped on video frames \((V = 3, M = 3)\). Incorrect radar objects stays non-persistent (red) even after appearing in three consecutive frames. While correct radar object becomes persistent (green) after appearing in three consecutive frames (frame 820 to 822). When a persistent radar object is missed by radar in one frame, it still remains persistent and shown in circle (green) in frame 825 and 826, S. B. McLaughlin, S. R. Fritz, S. L. Williams, and T. Buche. Overview of the MSF 100 motorcyclist naturalistic study, 2014, Used under fair use, 2015. .......................... 34
5.5 Radar data trajectory on 30 video frames. Trajectory of correct and incorrect data is clearly distinct. ................................................................. 35

5.6 Qualitative results for 16 consecutive frames of video for $V = 3$, $M = 3$, $N = 14$. We show the raw radar data at each frame, valid radar data, bounding box detected by DTBB-T algorithm, lead vehicle. We also show the bounding box returned by DTBB-T algorithm in thick magenta box when the car was not even detected by radar. S. B. McLaughlin, S. R. Fritz, S. L. Williams, and T. Buche. Overview of the MSF 100 motorcyclist naturalistic study, 2014, Used under fair use, 2015. ................................................................. 40

5.7 Amount of time taken to process each frame .................................................. 41
List of Tables

3.1 Train dataset statistics ....................................................... 13
3.2 Test dataset statistics .................................................... 14

4.1 Average height and width of cars at different distance groups ......... 22

5.1 Persistent radar object detection on Train and Test dataset ............... 30
5.2 Car detection and tracking results on Train and Test dataset for the DTBB-T algorithm ........................................................ 36
5.3 Car detection test results comparison with baseline methods .......... 36
5.4 Lead vehicle detection results on train and test dataset ................... 41
Chapter 1

Introduction

The objective of this thesis is to develop a radar-machine vision fusion system that will be able to analyze an incoming video stream in real-time and detect lead vehicles which the host motorcycle is following. The machine vision is to validate radar data and identify lead vehicles, so that rider behavior while following a lead vehicle can be further analyzed. The work is performed on motorcycle trip data in naturalistic studies conducted by the Virginia Tech Transportation Institute Motorcycle Research Group. A trip is defined by the time when the motorcycle is turned on to the time when it is turned off. The proposed approach uses both video and radar data that were recorded during a trip. This chapter gives a short introduction to the lead vehicle detection and tracking problem, its application on transportation and driving modes, and an introduction to object detection problem.

1.1 Transportation Application

Motorcycle riding has become more popular in recent years; there were about 8.5 million motorcycles on the road in 2012, according to the U.S. Department of Transportation. Motorcycles are far less crashworthy than cars or any other closed four-wheel vehicles. In United States, thousands of people die on the roads each year in motorcycle crashes. The National Highway Traffic Safety Administration (NHTSA, 2013) estimated that in 2013, there were 4,668 motorcyclist killed in motor vehicle traffic crashes. Also 88,000 motorcyclists were injured in 2013 (NHTSA, 2013). Among two-vehicle crashes, 74 percent of the motorcycle-car accidents were frontal impacts. Just 6 percent were struck in the back. This indicates that intelligent driver assistance system, such as forward collision warning system or lead vehicle detection system for motorcycle could enhance the safety of motorcyclist by monitoring the on-road environment and providing feedback or rider support. Detecting and tracking lead vehicles ahead is not only important for the motorcyclist to ride safe on road, it will also help researchers to analyze the rider’s behavior/performance in different driving modes.
1.2 Driving Modes

The investigation focuses on detecting and tracking the lead vehicle in order to identify when the host vehicle is in a state of "following" behind another vehicle. The system is intended identify epochs when the host vehicle is following a lead vehicle, and also other vehicles which are ahead of the host vehicle and within the visual range. For further understanding of the application and design requirements of the system, it is helpful to understand the driving scenario and terms.

A Closest In-Path Moving Vehicle (CIPV) is called a “lead vehicle” (illustrated in Figure 1.1). At any point of time, there can be almost one lead vehicle. Forbes (1972) demonstrated that vehicles having the way clear ahead up to 10 or 12 seconds can be thought to be working in an open-road driving mode. In this state, the driver’s pace is not influenced by the lead vehicle ahead. The driver of the following vehicle will changeover to overtaking mode as a slower moving lead vehicle comes nearer. In overtaking mode the driver of the overtaking vehicle closes to the vehicle being overtaken within approximately 9 seconds (Forbes, 1972). In this driving mode the overtaking vehicle follows the lead vehicle for 4 seconds. According to Forbes (1972), after the overtaking vehicle has approached to closer than 4 seconds from the lead vehicle, the driver can be considered to drive in a following mode (McLaughlin, 1998). In this mode, driver is adjusting speed and distance. Following driving mode is considered to extend to about 0.5 seconds headway. Identifying this region is the application of this work. The driving modes described above is demonstrated in Figure 1.2. The illustration also provides the associated headways in time and distance at various speeds.

1.3 Vehicle Detection and Tracking

Over the past decade, on-road vehicle detection system has been an active research area (Sun et al., 2006). Particularly vision-based detection system have become more popular (Sivaraman and Trivedi, 2013). Along with many other objects, car detection continues to be investigated extensively by the computer vision community. The definition of detection task as per Everingham and Winn (2011) is - “for each of the classes predict the bounding boxes of each object of that class in a test image (if any)”. Availability of large scale annotated datasets and deep convolutional neural networks have accelerated the research even more. The basic framework for any image based object detection algorithm is: given an image, a set of candidate proposals which can potentially contain an instance of an object are extracted and then each of them is classified for one or more classes. In this work we have considered only one class: Car. Detection of other vehicle types such as Truck or Bus are not included in this work and left for future work. The video from this transportation research project is somewhat different from generic object detection problem in dataset like PASCAL VOC (Everingham and Winn, 2011) or ImageNet (Deng et al., 2009). The problem in hand is to detect and track lead vehicles in low resolution, low frame-rate video, with processing feasible
Figure 1.1: Lead vehicle, preceding vehicle, and host vehicle illustration.

<table>
<thead>
<tr>
<th>Time Headway</th>
<th>0.5s</th>
<th>4s</th>
<th>9s</th>
<th>10s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance @ 55 mph</td>
<td>40ft</td>
<td>323ft</td>
<td>726ft</td>
<td>807ft</td>
</tr>
<tr>
<td>Distance @ 65 mph</td>
<td>48ft</td>
<td>381ft</td>
<td>858ft</td>
<td>953ft</td>
</tr>
<tr>
<td>Distance @ 75 mph</td>
<td>55ft</td>
<td>440ft</td>
<td>990ft</td>
<td>1100ft</td>
</tr>
</tbody>
</table>

Figure 1.2: Driving Regimes and Headways (based on Forbes, 1972) S. B. McLaughlin. Measurement of driver preferences and intervention responses as influenced by adaptive cruise control deceleration characteristics, 1998, Used under fair use, 2015.
in real time, but with certain assumptions that can be permitted in order to be able to do real-time image processing. For example, manual inspection of video and radar data revealed that lead vehicles beyond approximately 25 meters from the host vehicle are too small to detect using object detection algorithms which evaluate shape and size or unsupervised learned features for detection. Another assumption is that, in a video, an object’s overall appearance and location do not change significantly in few consecutive frames. Hence not every video frame needs to be analyzed.

Tracking cars in video means identifying objects over time as they move within a dynamic visual scene. Due to the movement of the object of interest, they vary in size, shape, color, and orientation. Additionally, because both the host vehicle and other vehicles are traveling along a roadway, lighting and background elements change in every frame. of cars in on-road environment. Not only appearance of cars changes, also background and illumination changes at every frame as the video is recorded from a camera installed in a moving vehicle. Moreover, on-road objects move independent of the moving camera. Changing environmental conditions include shadows due to buildings and trees along the path, cloudiness, and daylight and nighttime conditions. Application to video from motorcycles introduces additional variability due to roll (rotation along the axis pointing forward- W axis in Figure 4.2b), especially in curves. As the rider leans, the orientation of the video frame rotates, requiring tracking of vehicles in the presence of rolling movements. There have been numerous publications on general object recognition and tracking using vision, or combination of radar and vision for cars but few are focused on Motorcycles. We will review the related literature in Chapter 2.

In this work, first we map the radar data as points onto the video frame using perspective projection mapping. From all the mapped data persistent radar objects are determined. This is achieved by creating a horizontal profile of mapped radar points on frame. The persistent point locations is used as a first guess at a car’s location on the frame. A pre-determined rectangular box around such a point is used as an input image for the car detection algorithm. The detection algorithm might return multiple boxes per image, hence further filtering is done to select the final box. As there can be multiple cars in one frame, the detection task is parallelized over the number of persistent radar objects to check in a frame to speedup the process. After detecting a car, next step is to track it in the future frames. To improve speed, this is done after fourteen frames. The detection algorithm is run again to update car’s bounding box location if the radar point still exists on frame. Figure 4.1 provides a block diagram of the system. During the detection and tracking processes, we record data timestamps for each car and code which one is lead vehicle based on distance and lateral offset at that timestamp. We present the details of the algorithm in Chapter 4. Finally, we present the experiments, results, and performance evaluation of our methods in Chapter 5.
Chapter 2

Literature Review

In this chapter we discuss some of the related works in on-road vehicle detection and tracking based on vision or radar-vision fusion. With the advancement of imaging technology and sensing modalities, variety of sensors such as radar, lidar, and computer vision have become available for on-road vehicle detection task. There exists work using monocular, stereo vision, and active sensor vision fusion. Computing power has also increased with the emergence of parallelization and GPUs. Such hardware advances speeded up video processing using computer vision approaches, and thus has become popular for vehicle detection and tracking tasks.

Vehicle detection can be done either by computer vision only, by radar data only or by combining both. Vehicle detection approaches can be divided into two main categories - 1) appearance based, and 2) motion based (Sivaraman and Trivedi, 2013). In monocular vehicle detection, appearance based methods are more popular. These methods compute a variety of appearance features and recognize vehicles directly from images. Some earlier work (Hoffmann, 2006; Hilario et al., 2005; Arróspide et al., 2008) used very low level features such as symmetry along computed edges on image patches to recognize the vertical sides of the back surfaces of a preceding vehicle. Some other efforts used more advanced features like Histogram of Oriented Gradient (HOG) (Dalal and Triggs, 2005) and Haar-like features (Viola and Jones, 2001) for vehicle detection. Sivaraman and Trivedi (2014) did a comparative study using both of these features in an active learning setup: Support Vector Machine (SVM) classification using HOG features and Adaboost classification using Haar-like features. In spite of HOG feature's success in detection tasks, they cannot be used for real time vehicle detection as they are slow to compute. Haar-like features, on the other hand are efficient to compute. Sivaraman and Trivedi (2010) also used Haar-like features for detecting back preceding vehicles. In Zhang et al. (2011), they not only detected rear face of the vehicle but could also handle partial occlusions based on Scale Invariant Feature Transform (SIFT) features (Lowe, 1999). Edge and Speeded-Up Robust Feature (SURF) (Bay et al., 2008) were integrated to detect vehicles in the blind spot areas by Lin et al.
With the advent of deformable part-based model (DPMs) (Felzenszwalb et al., 2010b), object detection task became much more efficient. DPMs uses HOG features and latent SVM and represent objects using mixture of deformable part models. The shape, size and orientation of on-road vehicles change and so are considered a deformable part-based object. DPM has been used in many vehicle detection system (Takeuchi et al., 2010; Niknejad et al., 2011, 2012). Another effort which takes the classifier of the DPM and improves it’s objective function to gain speed using Branch and Bound approach is Kokkinos (2011). This approach produces same detection results as DPM but runs 10-20 times faster on average. Convolutional Neural Networks (CNN) (Jia et al., 2014) features are one of the most successful approaches for object detection in static images. One of the state-of-the-art object detectors, region proposals with CNN (R-CNN), by Girshick et al. (2014) first extracts about 2,000 candidate windows from each image via selective search (Uijlings et al., 2013). Then the image region in each window is warped to a fixed size (227 × 227). A pre-trained deep network is used to extract the feature of each window. A binary SVM classifier is then trained on these features for detection. R-CNN generates results of compelling quality and substantially outperforms previous object detection methods. However, computing CNN features is time consuming. Moreover, the computational requirements for applying CNN for all candidate proposals for each frame would be unrealistic in case of vehicle detection from video. Another comparatively fast approach, Spatial Pyramid Pooling net (SPP-net) by He et al. (2014), uses deep learning features to detect objects very accurately. But this approach also is not suitable when it comes to detect vehicles in video due to its computation time. Using GPU processing one frame takes an average of 14 secs.

Motion based monocular vehicular detection methods, on the other hand, detects cars from a sequence of images. Optical flow (Lucas et al., 1981) has been extensively used for motion based vehicle detection (Sun et al., 2004). A number of efforts have used this optical flow approach for real time vehicle detection and tracking Arróspide et al. (2008); Yamaguchi et al. (2006); Alonso et al. (2008); Song and Chen (2007). One of the motion based vehicle detection and tracking efforts Jazayeri et al. (2011) used geometry features to trace vehicles in motion from the video. They used Viterbi Algorithm (Forney Jr, 1973) in the hidden Markov model (HMM) to estimate the probability of a trace point to be either car or background.

So far we have discussed works that used vision as the main component for vehicle detection and tracking. Fusion of vision with other sensors has received considerable attention among researchers in recent years. Radar can provide good longitudinal and lateral range measurement to a preceding vehicle. In most vehicle based sensor-fusion work, either radar data are used to locate other vehicles and vision is used to validate the radar, or vice versa. Richter et al. (2008) used a combination of radar measurements and a contour based image processing algorithm to detect objects in the path of a moving vehicle. Another work by Liu et al. (2011), first detects vehicles using shadow segmentation. Next a matching process between the video and radar data is used to validate the detection. In Alessandretti et al. (2007) radar data is mapped on video frames, then in that image area is searched for vehicles
using vertical symmetry.

The present work closely matches with Tan et al. (2013), where radar data is mapped on frames and then a fixed area around each radar point is passed through the HOG-SVM classifier for vehicle detection. This approach tracks a vehicle that is in the same lane as the host vehicle. However, our work differs from this work in the following ways: 1) we validate each radar object using the trajectory of radar data itself before validating the vehicle using an object detection algorithm; 2) we use more advanced Dual-Tree Branch-and-Bound (Kokkinos, 2011) algorithm to detect cars; 3) we detect all the cars that are in the video and detect the lead vehicle at each timestamp.
Chapter 3

Dataset

3.1 Naturalistic Motorcycle Dataset Details

The datasets for which this work is designed are collected by the Virginia Tech Transportation Institute in riding studies conducted for the Motorcycle Safety Foundation and the National Highway Traffic Safety Administration. These studies involve participants riding their personal motorcycles with sensors and cameras installed for up to two years. The objectives of these research efforts are to understand natural riding and the causes of crashes in today’s transportation system, and subsequently identifying strategies for avoiding crashes. Though data are still accumulating, approximately 54,000 trips have been recorded, with a trip being the time when the motorcycle was started to when it was turned off. these trips total to approximately 1,000,000 minutes of video and sensor data capturing approximately 487,000 miles of riding by 260 participants. The majority of the data are collected in California, Florida, Virginia, and Phoenix, but because data were collected from every place the participants rode, riding in nearly all of the U.S. is included.

3.1.1 Data Acquisition Hardware

The data acquisition hardware consisted of the main unit (that includes the processor, firmware, hard drive, cellular antenna, technician access panels, and interfaces to the sensors distributed around the motorcycle), as well as a sensor suite (McLaughlin et al., 2014). The sensor suite included:

- Five color cameras (forward, rear, left, right, rider)
- Accelerometers (3 axes)
- Gyro (3 axes)
Figure 3.1: Illustration of main unit, forward camera and radar mounted on a motorcycle, S. B. McLaughlin, S. R. Fritz, S. L. Williams, and T. Buche. Overview of the MSF 100 motorcyclist naturalistic study, 2014, Used under fair use, 2015.

- Forward radar (up to eight targets tracked)
- GPS
- Speed (from GPS)
- Turn signal state
- Brake lamp state
- Brake lever inputs
- Engine RPM
- Machine vision lane tracker

Approximately 35-75 seconds after the motorcycle was started, the data acquisition system began recording and continued to record until the motorcycle ignition was turned off. Data recording rate varied from sensor to sensor. In Figure 3.1 the main unit and radar are shown on one of the motorcycle types included in the study.

For the current work only video data recorded by the forward facing camera and the radar data was used. The forward video camera was included in the radar housing. The camera’s optical lens specification is as follows:

- Focusing Range: 60cm to $\infty$
• Focal Length: Specified as 3.0mm. Calibration computation for field-of-view (FOV) indicated focal length of 3.8mm.

• Field-of-view: 60°

• TV Distortion: < 1%

The forward camera sensor is 1/4” CCD, with a film plane of 3.2mm × 2.4mm size for forward video recording.

3.1.2 Radar Data Description

The radar has a range of 200 meters, which means the radar can detect vehicles up to 200 meters. In the present work, readings of up to 100 meters were used. At every time stamp of the trip the data acquisition system retained at most 8 objects reported by the radar. For each timestamp $t$, the proposed algorithm uses three of the recorded measurements: (1) Total number of objects detected at $t$, (2) Distance from host vehicle to the detected objects, and (3) Lateral distance (left or right) of the detected objects from the radar’s heading direction.

3.1.3 Video Data Description

The video of the forward scene collected by the system has a resolution of 240 × 720 and a frame rate of 14.985 fps. This frame rate and resolution creates motion blurring within video frames. When a vehicle is beyond 20 meters a lead vehicle is considered too small for detection using traditional object detection algorithm and meaningful feature extraction. An example of a video frame recorded by the forward camera is shown in Figure 3.2

3.2 Data Annotation

Though there are large amounts of this research video data, the video data is not annotated. To build, tune parameters, test, and evaluate performance of every component of the approach - valid radar object detection, car detection, and tracking, this work first required annotated video. Two datasets were annotated, a train and test set. The Train dataset contains 705 annotated frames from one trip. For testing, 350 frames were annotated from another trip. Each object of car class which appears in the frame and can be a potential lead vehicle or preceding vehicle was annotated. Any vehicle that could be a lead vehicle candidate but missed by radar was labeled too. The annotation of each car includes:

• Data timestamp
Figure 3.2: A video frame from a trip recorded by the forward camera, S. B. McLaughlin, S. R. Fritz, S. L. Williams, and T. Buche. Overview of the MSF 100 motorcyclist naturalistic study, 2014, Used under fair use, 2015.

- Video frame number
- Radar Target ID
- Radar range
- Radar lateral range
- The state of car: depending on whether the data is correct or incorrect, and radar range the state of the car can fall into one of the following categories:
  - Radar data correct and forward distance $\leq$ 25 meters - this data will be used to compute car detection performance.
  - Radar data correct and forward distance $\leq$ 25 meters, but car is too small and significantly occluded - this data will not be used for car detection performance computation.
  - Radar data correct and 25 meters $<$ radar range $<$ 100 meters - vision algorithm will not run for such data.
  - Radar data incorrect - radar objects are static background object.
  - Duplicate reading of same object with respect to frame - radar objects were recorded multiple times during the duration of one frame.
  - Missed by radar - data timestamps to frame number mapping is not always continuous
- A bounding box that contains the car - the top left corner and bottom right corner coordinates are saved. This data will be used to compute car detection performance
- Frame name in $<$ file_id $>$-$<$ frame_no $>$ format. Eg. 613197.5734
Lead vehicle or not - will be used to compute accuracy of lead vehicle detection

Annotating the data this way helps us to measure the performance of each component of the algorithm individually. Our Train dataset statistics are shown in Table 3.1. The position of most of the cars in the scene were either oblique view or rear view. Statistics of the Test dataset is presented in Table 3.2. Similar to Train dataset, cars in Test dataset too mostly have oblique or rear view. Figure 3.3 shows an e.g. of two cars annotated on a video frame. In our work, we have ignored any radar data which are more than 100 meters away from the host vehicle. In training dataset there were 2011 total radar readings and 297 readings were ignored. Among 297 readings, 11 readings were not even mapped on frame. In test dataset, 195 objects were more than 100 meters away from host vehicle, hence ignored.
Table 3.1: Train dataset statistics

<table>
<thead>
<tr>
<th>Item</th>
<th>Train Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of frame</td>
<td>705</td>
</tr>
<tr>
<td>Total timestamps</td>
<td>1179</td>
</tr>
<tr>
<td>Length of video</td>
<td>51.73 seconds</td>
</tr>
<tr>
<td>Total raw radar reading</td>
<td>1714</td>
</tr>
<tr>
<td>Total correct radar reading (manually inspected)</td>
<td>1539</td>
</tr>
<tr>
<td>Total incorrect radar reading (manually inspected)</td>
<td>100</td>
</tr>
<tr>
<td>Accuracy of raw radar detection</td>
<td>89.8 %</td>
</tr>
<tr>
<td>Instances of car to detect by vision algorithm</td>
<td>906</td>
</tr>
<tr>
<td>Number of lead vehicles with forward distance $\leq 25$ meters</td>
<td>590</td>
</tr>
<tr>
<td>Number of total lead vehicles (forward distance up to 100 meters)</td>
<td>630</td>
</tr>
<tr>
<td>Number of radar data where the car is either too small or occluded, but distance of the car from the host vehicle is $\leq 25$ meter</td>
<td>104</td>
</tr>
<tr>
<td>Number of duplicate/repeated radar data</td>
<td>75</td>
</tr>
<tr>
<td>Number of cars missed by the radar completely but visually detectable</td>
<td>155</td>
</tr>
<tr>
<td>Number of cars missed by radar between 2 frames</td>
<td>43</td>
</tr>
<tr>
<td>Number of cars where part of the car is outside the frame area or truncated</td>
<td>14</td>
</tr>
</tbody>
</table>
Table 3.2: Test dataset statistics

<table>
<thead>
<tr>
<th>Item</th>
<th>Test Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of frame</td>
<td>350</td>
</tr>
<tr>
<td>Total timestamps</td>
<td>482</td>
</tr>
<tr>
<td>Length of video</td>
<td>25.868</td>
</tr>
<tr>
<td>Total raw radar reading</td>
<td>729</td>
</tr>
<tr>
<td>Total correct radar reading (manually inspected)</td>
<td>666</td>
</tr>
<tr>
<td>Total incorrect radar reading (manually inspected)</td>
<td>50</td>
</tr>
<tr>
<td>Accuracy of raw radar detection</td>
<td>91.36%</td>
</tr>
<tr>
<td>Number of cars to detect by vision algorithm</td>
<td>304</td>
</tr>
<tr>
<td>Number of lead vehicles with forward distance $\leq$ 25 meters</td>
<td>145</td>
</tr>
<tr>
<td>Number of total lead vehicles (forward distance up to 100 meters)</td>
<td>200</td>
</tr>
<tr>
<td>Number of radar data where the car is either too small or significantly occluded, but distance of the car from the host vehicle is $\leq$ 25 meter</td>
<td>33</td>
</tr>
<tr>
<td>Number of repeated radar data</td>
<td>13</td>
</tr>
<tr>
<td>Number of cars missed by the radar completely but visually detectable</td>
<td>20</td>
</tr>
<tr>
<td>Number of cars missed by radar between 2 frames</td>
<td>43</td>
</tr>
</tbody>
</table>
Chapter 4

Methodology

In this work a radar guided vision model is developed in order to localize, detect and track the lead vehicles. By mapping the radar objects on the video frame, region of interests (ROIs) are generated. These ROIs are candidate image areas for preceding vehicles. Next, the vision algorithm is run on each ROI to detect car and validate the corresponding radar object. To track the detected car in the following frames, we use trajectory generated by the radar-video fusion. Figure 4.1 provided a block diagram of the system.

4.1 Mapping Radar Objects on Video Frame

A radar object is represented by two values - the range $w$ and the horizontal offset $u$. Range is the distance from radar to the target vehicle and offset $u$ is the horizontal distance from radar heading direction to the target position. A negative offset means the target is to the right and a positive offset means the target is to the left. The radar-target data representation and radar-camera coordinate system is shown in Figure 4.6. In Figure 4.2b, $(U, V, W)$ represents...
the radar coordinate system and \((X, Y, Z)\) represents the camera coordinate system. Radar data gives target vehicle’s location in radar coordinate system. In this coordinates, ground is defined as \(V = 0\). Radar’s position is given by \((U = 0, V = -h_r, W = 0)\). \(U\)-axis points to the right, \(V\)-axis points down and \(W\)-axis points to the vehicle heading direction. The offset from radar to camera is \((T_u, T_v, T_w)\). The camera’s position is given by \((X = T_u, Y = -h_r - T_v, Z = -T_w)\). Rotation along \(V\) axis is represented by \(\phi\) and rotation along \(U\) axis is represented by \(\theta\).

### 4.1.1 Rigid Transformation

Using imaging geometry 3D point in the radar coordinate system \(P_r\) can be mapped to a 2D point \(P_f\) in an image. At first, through Rigid Transformation (translation and rotation) \(P_r\) is mapped to a point \(P_c\) in camera coordinate system. Eq. (4.1) shows the mapping and the homogeneous matrix equation is shown below

\[
P_c = R(P_r - T) \tag{4.1}
\]

\[
\begin{bmatrix}
X \\
Y \\
Z \\
1 \\
\end{bmatrix} =
\begin{bmatrix}
r_{11} & r_{12} & r_{13} & 0 \\
r_{21} & r_{22} & r_{23} & 0 \\
r_{31} & r_{32} & r_{33} & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 0 & -T_u \\
0 & 0 & 0 & -T_v \\
0 & 0 & 0 & -T_w \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
U \\
V \\
W \\
1 \\
\end{bmatrix}
\]
Defining the rotation matrices $R_u$ along $U$ axis and $R_v$ along $V$ axis, we have:

$$R_u(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R_v(\phi) = \begin{bmatrix} \cos\phi & 0 & \sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\phi & 0 & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

There is no rotation along the $W$ axis in the data, beginning of each video, $\theta$ and $\phi$ are determined and set to values that optimizes the alignment of camera coordinates and radar coordinates. Variables $\theta$ and $\phi$ remain invariant for a particular rider.

### 4.1.2 Perspective Mapping Transformation

After transforming the radar data point to camera coordinate system, the camera is calibrated to determine the relationship between what appears on the image plane and where it is located in the 3D world. The camera coordinate system whose origin is at the center of projection $c$ and whose $Z$ axis is along the optical axis is shown in Figure 4.3. This coordinate system is called the standard coordinate system of the camera. A point $M$ on an object with coordinates $(X,Y,Z)$ will be imaged at some point $m = (x,y)$ on the image plane. These coordinates are with respect to a coordinate system whose origin is at the intersection of the optical axis and the image plane, and whose $x$ and $y$ axes are parallel to the $X$ and $Y$ axes. The relationship between the two coordinate systems $(c, x, y)$ and $(C, X, Y, Z)$ is given by

$$x = \frac{Xf}{Z} \quad \text{and} \quad y = \frac{Yf}{Z} \quad (4.2)$$

Where $f$ is the focal length of camera. This can be written linearly in homogeneous coordinates as

$$\begin{bmatrix} sx \\ sy \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & f & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

where $s \neq 0$ is a scale factor. The actual pixel coordinates $(p,q)$ are identified with respect to an origin in the top left hand corner of the image plane and will satisfy

$$p = u_c + \frac{x}{\text{pixel width}} \quad \text{and} \quad q = v_c + \frac{y}{\text{pixel height}} \quad (4.3)$$

We can express the transformation from three dimensional world coordinates to image pixel coordinates using a $3 \times 4$ matrix. This is done by substituting Eq. 4.2 into Eq. 4.3 and multiplying through by $Z$ to obtain

$$Zp = Zu_c + \frac{Xf}{\text{pixel width}} \quad \text{and} \quad Zq = Zv_c + \frac{Yf}{\text{pixel height}} \quad (4.4)$$
In other words,

\[
\begin{bmatrix}
    sx \\
    sy \\
    1
\end{bmatrix}
= \begin{bmatrix}
    \frac{f}{\text{pixel width}} & 0 & u_c & 0 \\
    0 & \frac{f}{\text{pixel height}} & v_c & 0 \\
    0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
    X \\
    Y \\
    Z \\
    1
\end{bmatrix}
\]

where the scaling factor \( s \) has value \( Z \). In short hand notation, we write this as

\[ \tilde{u} = P \cdot \tilde{M} \]

where \( \tilde{u} \) represents the homogeneous vector of image pixel coordinates, \( P \) is the perspective projection matrix, and \( \tilde{M} \) is the homogeneous vector of world/camera coordinates.

This transformation is performed using calibration data achieved by careful measurement of intrinsic and extrinsic camera parameters measurement. Because these parameters are measured once and no real-time update is done to take into account roll and pitch, sometime radar data mapping can be erroneous, especially at curves. Taking roll or leanangle information into account for better radar object mapping is left for future work. We also discard radar data which maps outside the video frame area. These radar objects are usually more than 200 meters away from the host vehicle.
4.2 Validating Radar Objects using Trajectory and Car Detection

After the radar objects are mapped onto video frame, trajectory of mapped radar points on video frames are used to determine persistent radar objects. Using the location of these persistent radar objects as first guess, we further validate each of these objects using car detection algorithm.

4.2.1 Determine Persistent Radar Objects

Radar data is not always accurate, and contains readings that are not of interest in this work. Sometime radar detects stationary objects in the background as moving and continues detecting that point for few consecutive frames. Therefore, relying on radar data based on a single timestamp can generate incorrect region of interest. A horizontal profile of mapped radar points along the time axis shown in Figure 4.4 reveals the unique characteristics of true and false targets. In this figure, the $x$-axis represents the number of columns of a frame and $y$-axis represents time. At each timestamp all the mapped radar objects’ $u$ value from Eq. 4.3 are plotted, which tells us the lateral position of radar objects on a frame. Consecutive profiles along the time axis generate a condensed spatio-temporal trajectory plot $S(x,y)$ of the radar points. When the object is close on the road the trajectory is vertical, whereas it sweeps to become horizontal for background. Also, image velocity of a pixel on buildings/background passing by is high, and at the distance down the street, its low. Therefore, when a vehicle is moving within the road area even in an irregular zig zag way, it may stay in the video frame. But background captured by radar will soon leave the image frame. Thus radar objects that points to moving vehicle produces more vertical and compact trajectories compared to the radar objects which points to the background. In our data we have noticed, a vehicle on road does not produce a gap of more than 5 unit of euclidean distance between two consecutive frames. False radar points will produce more spread out horizontal trajectory.
Figure 4.4: One-dimensional profile of mapped radar object on frames for 2000 timestamps of a video from Train Dataset

Figure 4.5 depicts this in a schematic form. These trajectories are recorded to get the persistent radar object. An outline of the algorithm to get the trajectory is given in Algorithm 1. Only the surrounding area of persistent radar objects are considered as region of interest for vehicle detection using vision algorithm.
Figure 4.5: Trajectories of vehicles in motion and background from stacked horizontal-profiles along the time axis. The zig zag (red) is more likely the motorcycle moving back and forth or wiggling the radar at lower speed, Y. Tan, F. Han, and F. Ibrahim. A radar guided vision system for vehicle validation and vehicle motion characterization. In Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE, pages 1059?1066. IEEE, 2007, Used under fair use, 2015.

Persistent radar objects indicates a point on the frame, it does not provide any dimensional guidance about the region of interest. To get the dimension estimation, ground truth bounding boxes are used. The size of the cars in frames varies with distance: larger distances create smaller sizes. Estimating an appropriate image area is necessary for minimizing computational demand and minimizing false detections. For radar objects at long distances, using a smaller area reduces computational requirements. Moreover, using a smaller image area reduces the possibility of false positive detection by the vision algorithm. Therefore, when a persistent radar object is detected, a rectangular image area of double the size of it’s distance group is cropped with the point in center and used as input image by the car detection algorithm. Figure 4.6a and 4.6b shows how height and width changes as distance changes. Based on the ground truth data, we divide 25 meter distance in front of host vehicle into four groups. The average aspect ratio of ground truth bounding boxes in Train dataset is approximately 0.65 (height/width). Table 4.1 shows the groups and average height and width of cars in each group. Figure 4.7 illustrates a car’s size in frames from different distance groups.

4.2.2 Car Detection using Dual-Tree Branch-and-Bound

Persistent radar objects generate region of interests (ROIs) for car search in video frame. To validate the radar data we look for a car in these ROIs. For each ROI, we use efficient
Prakriti Banik Chapter 4. Methodology

Table 4.1: Average height and width of cars at different distance groups

<table>
<thead>
<tr>
<th>Distance $d$ (in meter)</th>
<th>Avg. Height</th>
<th>Avg. Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d \leq 5$</td>
<td>160</td>
<td>240</td>
</tr>
<tr>
<td>$5 &gt; d \leq 10$</td>
<td>120</td>
<td>185</td>
</tr>
<tr>
<td>$10 &gt; d \leq 15$</td>
<td>60</td>
<td>92</td>
</tr>
<tr>
<td>$15 &gt; d \leq 25$</td>
<td>45</td>
<td>70</td>
</tr>
</tbody>
</table>
Dual Tree Branch-and-Bound (DTBB) (Kokkinos, 2011) object detection algorithm with deformable part models. Object detection using cascade classifiers built from deformable part-based models (DPM) from Felzenszwalb et al. (2010a) is an efficient and frequently used algorithm. The main concept of DPM is to optimize the classifier score $S(x)$ in Eq. 4.5 with respect to the part displacements and the global object pose. Kokkinos (2011) presents the details.

$$S(x) = \sum_{p=0}^{P} \max_{x_p} s_p(x)$$

$$= \sum_{p=0}^{P} \max_{x_p} m_p(x_p, x)$$

$$= \sum_{p=0}^{P} \max_{x_p} U_p(x_p) - (h_p - h - \eta_p)^2 H_p - (\nu_p - v - \nu_p)^2 V_p$$

(4.5)

Though DPM’s are excellent for object detection, they are slow. Detecting cars in one image frame takes on average 1.8 secs in a CPU. DTBB (Kokkinos, 2011) approach takes the classifier of Felzenszwalb et al. (2010a) and improves the optimization problem of the Mixture-of-DPM by replacing the Generalized Distance Transform (GDT). As a result it becomes faster. The DTBB approach uses Dual Tree algorithm (Gray and Moore, 2003) for upper bound computation and Efficient Subwindow Search (ESS) (Lampert et al., 2009) for detection.

The DTBB approach maximizes non-parametric, non-convex or non-differentiable functions. Using a prioritized search strategy, DTBB looks for the interval that contains the function’s
Figure 4.7: Four frames with cars from each of the four distance groups. (from top to bottom) a) car distance 4 meter, b) car distance 8 meter, c) car distance 14 meter, and d) car distance 24 meter, S. B. McLaughlin, S. R. Fritz, S. L. Williams, and T. Buche. Overview of the msf 100 motorcyclist naturalistic study, 2014, Used under fair use, 2015.
maximum value. Function’s upper bound within it determines the priority of an interval. The maximization problem starts from an interval that contains the whole function domain and iteratively reaches for the solution. A priority queue contains a domain of the function being maximized. At each step, DTBB pops an interval of solutions from that priority queue, splits it into subdomains (Branch), and a new upper Bound is computed using Dual Tree data structure (Gray and Moore, 2003) for the subdomains (Bound). The sub-intervals are again inserted into the priority queue and the process is repeated until a singleton interval is popped. The first singleton will be the global maxima of the function if the bound is tight for singletons. Optimization problem for a multiple parts object can be easily done using this approach. On average it is 10-20 times faster than Felzenszwalb et al. (2010a). The DTBB approach is built on Felzenszwalb et al. (2010b); Girshick et al., and trained on PASCAL VOC 2007 and 2010 dataset. Car is one of the twenty classes in PASCAL VOC dataset. Though trained on PASCAL VOC dataset, the model learned by the DTBB approach worked well for the cars present in video frames of our Train datasets too. Therefore, we have used the publicly available trained model from Kokkinos (2011, 2012b,a), for car detection in ROIs and did not learn a new car detection model from our dataset.

As mentioned, in the motorcycle datasets, the data acquisition system retains from zero to eight radar objects at one timestamp. Therefore, in each frame there can be multiple cars. To speedup the detection process, we run Dual-Tree Branch-and-Bound (DTBB) algorithm in parallel for all the persistent radar objects at the same time. For each ROI, the standard detection scenario is considered where all detections having a score above a certain threshold is returned. Thus multiple detections might be returned for each ROI by the DTBB approach, but we need only one detection containing that corresponds to the persistent radar object. To select the final detection, road horizon, persistent radar objects’s distance, car’s average height and width from distance groups in Table 4.1, and location of previous detection are used (in case of tracking). A detection box outside the much above or below the road horizon, too small compared to the distance group height and width, and far away from radar point is discarded. In Figure 4.8, we plotted \( v \) value of all radar objects from the training set. Most of the points fall in the interval of 89 to 91 row number. Therefore, on video frame, row number 100 is taken as road horizon. Finally, the detection with minimum distance from the persistent radar point on video frame and of size close to the distance group it belongs to is considered to be the correct detection. Figure 4.9 shows an example of a video frame with persistent radar object mapped, the ROI, two detections returned by DTBB, and the final detection.

### 4.3 Lead Vehicle Detection

In the previous section we validated a radar object at a given timestamp. More specifically, we detected all persistent radar objects that exist at a timestamp, and if the vision algorithm identified a car at those locations then we obtained a bounding box around each of these
Figure 4.8: Vertical profile of radar objects on video frame. This determines the road horizon persistent radar objects. After obtaining validated radar objects, we determine the lead vehicle at that timestamp. The Closest In-Path Moving Vehicle (CIPV) is called the lead vehicle as illustrated in Figure 1.1. The lead vehicle usually be moving on the same lane as the motorcycle. Correct lead vehicle determination requires lane information. Though we have integrated the lane estimation module available in MATLAB Computer Vision System Toolbox in our system, the estimation is not robust to occlusion. Therefore we have used lane width and lateral distance measurements by radar for each validated objects along with an assumption to determine the in-path moving vehicles. According to the U.S. Department of Transportation, standard lane width is 12 feet (3.6 meters). Therefore, our assumption is that a preceding vehicle on the same lane not more than 5 feet away laterally from the motorcycle can be called a lead vehicle. A scenario where a preceding vehicle on the same lane exists but it is not called a lead vehicle is illustrated in Figure 4.10.
Figure 4.9: Example of final detection selection, when multiple detections are returned by the DTBB algorithm. A video frame with one persistent radar object (top), the ROI with multiple detections returned by DTBB (middle), and the final detection (bottom), S. B. McLaughlin, S. R. Fritz, S. L. Williams, and T. Buche. Overview of the MSF 100 motorcyclist naturalistic study, 2014, Used under fair use, 2015.
Figure 4.10: No lead vehicle even when preceding vehicle exists on the same lane

To determine the lead vehicle, all validated radar objects with less than 5 feet lateral distance from the motorcycle is selected. Among these radar objects, the one with the minimum forward distance is marked as the lead vehicle at that timestamp.

4.4 Tracking the Detected Car

Once a vehicle is detected it needs to be tracked in the subsequent frames. One way to track the objects can be to run the detection algorithm at each frame. This solution is computationally expensive. To make the system run faster, we use the overall trend of the trajectory of validated radar points in the subsequent frames. For 14 frames we adjust the position of the detection box based on the movement of radar point compared to its previous location. After 14 frames we run the detection algorithm again to detect the car again. Detecting more frequently than every 14 frames does not save much time and detecting less often than 14 frames can introduce large changes in the location of the car, especially on highway. The value 14 is chosen by running experiments on Train dataset. Our main objective is to validate the radar data using vision and thus exact overlap of the bounding box is not very important. Detecting a car in the persistent radar object’s surrounding area is useful enough to say that the radar reading is coming from a moving vehicle. In most of the cases this simple tracking technique computes bounding boxes of more than 40% overlap with car. Hence this serves our purpose.
Chapter 5

Results

In this section we discuss the experiments performed in order to determine appropriate values for different algorithm parameters and present the results on both Train and Test dataset. We have tuned each parameter on training dataset. The Train dataset was used to determine the final values of all parameters. For training and testing purpose we have annotated two trip videos. Statistics of Train dataset is given in Table 3.1 and statistics of Test dataset is given in Table 3.2 (Chapter 3, Section 3.2).

There are four aspects of this work which need to be tested in order to test the performance of the whole algorithm.

1. Persistent radar object detection
2. Car detection and tracking
3. Lead vehicle detection
4. Determining epochs of preceding vehicles on road

We will evaluate each of these tasks and present the results.

5.1 Persistent Radar Object Detection Results

The purpose of detecting persistent radar object is to discard those radar objects that point to background objects such as parked cars, buildings, light posts, traffic posts, etc.. If radar detects an object for some number of consecutive frames then we call it a persistent radar object. We represent this number by parameter $V$. Once an object is declared as persistent, it continues to be persistent even if it is missed by radar in some number of consecutive frames. Parameter $M$ represents this number of consecutive frames. We have experimented
with different value combinations of $V$ and $M$ for persistent radar object detection on Train dataset. The result is presented in Table 5.1. Finally, we chose $V= 3$, 5 and $M= 3$, 4 for next experiments. The last two rows of Table 5.1 contains results for Test dataset.

Figure 5.1 shows the precision-recall graph and Figure 5.2 illustrates the ROC curve for persistent radar object detection on train dataset. A confusion matrix for all four combinations and three categories of radar objects (correct, incorrect, and duplicate with respect to frame number) are shown in Figure 5.3.

We used precision, recall, and F1 score to evaluate the performance of persistent radar object detection. Definitions of precision and recall are as follows:

**Precision:** The fraction of retrieved instances that are relevant.
**Recall:** The fraction of relevant instances that are retrieved.

**F1 Score:** Harmonic mean of precision and recall.

\[
\text{Precision of persistent radar object detection} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}}
\]

\[
\text{Recall of persistent radar object detection} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}
\]

\[
F1 \ Score = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}
\]

Table 5.1: Persistent radar object detection on Train and Test dataset

<table>
<thead>
<tr>
<th>$V$</th>
<th>$M$</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 Score</th>
<th>Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>93.88</td>
<td>99.74</td>
<td>96.72</td>
<td>Train</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>96.78</td>
<td>85.90</td>
<td>91.01</td>
<td>Train</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>97.13</td>
<td>79.30</td>
<td>87.31</td>
<td>Train</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>98.47</td>
<td>67.06</td>
<td>79.78</td>
<td>Train</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>96.57</td>
<td>87.08</td>
<td>91.58</td>
<td>Test</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>96.47</td>
<td>77.06</td>
<td>85.68</td>
<td>Test</td>
</tr>
</tbody>
</table>
Figure 5.1: Precision-Recall curve for persistent radar object detection on Train dataset

Figure 5.2: ROC curve for persistent radar object detection on Train dataset
Figure 5.3: Confusion matrix for persistent radar object detection for different combinations of $V$ and $M$ on Train dataset. Class 1: correct radar objects; Class 2: incorrect radar objects; Class 3: duplicate radar objects with respect to frame number

5.1.1 Qualitative Results of Valid Radar Object Detection

In Figure 5.4 we present eight consecutive frames from the video in the Train dataset. We plot both raw radar data points and persistent radar data points on frame. Non-persistent radar points are coded in red. The trajectory for 30 frames starting from frame number 820 is shown in Figure 5.5
Frame no: 820  No. of raw radar object: 3  No. of persistent radar object: 0

Frame no: 821  No. of raw radar object: 3  No. of persistent radar object: 0

Frame no: 822  No. of raw radar object: 3  No. of persistent radar object: 2

Frame no: 823  No. of raw radar object: 3  No. of persistent radar object: 2
Figure 5.4: Raw and persistent radar objects mapped on video frames ($V = 3$, $M = 3$). Incorrect radar objects stays non-persistent (red) even after appearing in three consecutive frames. While correct radar object becomes persistent (green) after appearing in three consecutive frames (frame 820 to 822). When a persistent radar object is missed by radar in one frame, it still remains persistent and shown in circle (green) in frame 825 and 826.

5.2 Car Detection and Tracking Results

In the Train dataset, there are total 1104 images of cars in total 705 frame. Out of 1104 images of cars, 906 instances were detected and rest 198 instances were missed by radar. With the goal of validating radar objects, if no radar object is presented, the vision algorithm will not be run. We have used the publicly available implementation of DTBB (Kokkinos, 2011, 2012b,a). After Dual Tree Branch and Bound (DTBB) algorithm detects a car, we start tracking the car for $N$ numbers of consecutive future frames. At $N + 1^{th}$ frame we run DTBB algorithm again to detect the car. We have experimented with $N = 10, 14, 20$. Car detection results from using Dual Tree Branch and Bound and tracking (DTBB-T) algorithm together is shown in Table 5.2. Test result is also included in Table 5.2. Average computation time to detect each car is also shown.

We have also compared our test results with four baseline approaches. **Baseline 1:** Running detection using Dual Tree Branch and Bound algorithm at each frame and no tracking on raw radar data (DTBB-R); **Baseline 2:** Running detection using Dual Tree Branch and Bound algorithm and tracking for 14 frames on raw radar data (DTBB-T-R). **Baseline 3:** Running detection using Deformable Part-based Model algorithm on valid radar object at each frame and no tracking (DPM); **Baseline 4:** Running detection using Deformable Part-based Model algorithm for valid radar objects and tracking for 14 frames (DPM-T). Detection results along with average runtime for each vehicle is presented in Table 5.3.
Table 5.2: Car detection and tracking results on Train and Test dataset for the DTBB-T algorithm

<table>
<thead>
<tr>
<th>V</th>
<th>M</th>
<th>N</th>
<th>AP</th>
<th>Avg. Runtime (secs)</th>
<th>Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3</td>
<td>10</td>
<td>76.29</td>
<td>0.1174</td>
<td>train</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>14</td>
<td><strong>76.32</strong></td>
<td>0.0826</td>
<td>train</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>20</td>
<td>76.10</td>
<td>0.0709</td>
<td>train</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>10</td>
<td>75.27</td>
<td>0.1144</td>
<td>train</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>14</td>
<td>75.45</td>
<td>0.0902</td>
<td>train</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>20</td>
<td>75.31</td>
<td>0.0740</td>
<td>train</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>14</td>
<td><strong>77.68</strong></td>
<td>0.0808</td>
<td>test</td>
</tr>
</tbody>
</table>

Table 5.3: Car detection test results comparison with baseline methods

<table>
<thead>
<tr>
<th>V</th>
<th>M</th>
<th>N</th>
<th>Method</th>
<th>AP</th>
<th>Avg. Runtime (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>DTBB-R (Baseline 1)</td>
<td>74.16</td>
<td>0.4176</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>14</td>
<td>DTBB-T-R (Baseline 2)</td>
<td>74.77</td>
<td>0.1717</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>DPM (Baseline 3)</td>
<td>75.44</td>
<td>1.6513</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>14</td>
<td>DPM-T (Baseline 4)</td>
<td>77.68</td>
<td>0.3967</td>
</tr>
<tr>
<td><strong>3</strong></td>
<td><strong>3</strong></td>
<td><strong>14</strong></td>
<td>DTBB-T (proposed)</td>
<td><strong>77.68</strong></td>
<td><strong>0.0808</strong></td>
</tr>
</tbody>
</table>
5.2.1 Qualitative Results of Detection and Tracking using Vision

We present some qualitative results on one video in Figure 5.6.
Chapter 5. Results
Figure 5.6: Qualitative results for 16 consecutive frames of video for $V = 3$, $M = 3$, $N = 14$. We show the raw radar data at each frame, valid radar data, bounding box detected by DTBB-T algorithm, lead vehicle. We also show the bounding box returned by DTBB-T algorithm in thick magenta box when the car was not even detected by radar, S. B. McLaughlin, S. R. Fritz, S. L. Williams, and T. Buche. Overview of the MSF 100 motorcyclist naturalistic study, 2014, Used under fair use, 2015.
In Figure 5.7 we plot the runtime for each of 705 frames processing from train dataset. Average runtime to process one frame is 0.0385 seconds.

5.3 Lead Vehicle Detection Results

Our goal is to detect the lead vehicle at every timestamp of the trip. Lead Vehicle is defined as the Closest In-Path Moving Vehicle (CIPV). The lead vehicle detection algorithm explained in Section 4.3 of Chapter 4 produced a precision of 90.65% and a recall of 80.51% for the Train dataset. For Test dataset we obtained 98.61% precision and 90.53% recall. The results of lead vehicle detection for both dataset is presented in Table 5.4. We also included results of lead vehicle detection from raw radar data on the Test dataset without applying any module of the algorithm. This is our baseline method for lead vehicle detection and the second row of the table shows the results.

5.3.1 Preceding Vehicles Epoch Recording Results

One additional purpose of detecting lead vehicle is to be able to analyze rider behavior in the presence of lead vehicle. Therefore, for each preceding vehicle we record two timestamps: (1) the timestamp when a vehicle is first detected by the system and (2) the timestamp when
the vehicle is not detected by the system anymore. Thus we can get the epochs for all lead vehicles during a trip. One issue with radar data is that, during an epoch, radar assigns multiple target ID to the same car. Our algorithm assigns a single vision ID to a car during it’s epoch. Thus we can retrieve all the radar target IDs those were assigned to the same car from this table.
Chapter 6

Conclusion

According to the National Highway Traffic Safety Administration estimates, “In 2013, there were 2,182 two-vehicle fatal crashes involving a motorcycle and another type of vehicle. In 42 percent (922) of these crashes, the other vehicles were turning left while the motorcycles were going straight, passing, or overtaking other vehicles. Both vehicles were going straight in 456 crashes (21%).” Frontal collision is one of the major crash categories. Thus how a rider rides when they are following another vehicle on road is an important question. And detecting and tracking preceding on-road vehicles is a crucial task.

This study has focused on an important task of vehicle detection and tracking from radar data. Radar data includes many targets that are not immediately relevant to vehicle operators, and so are not relevant in research. For example, radar return measures for buildings, parked cars, traffic posts, etc.. Therefore, relying solely on radar data for lead vehicle detection is not sufficient. Our method uses computer vision algorithm on top of radar data for this purpose. One cannot distinguish correct versus incorrect radar data from a single timestamp. However object trajectory over multiple timestamps exhibit distinct characteristics that can distinguish the correct from the incorrect. We take advantage of these characteristics of radar data by recording the trajectory of the objects. Thus we partially validate the radar data even before passing it to the vision algorithm for further confirmation. The radar validation from it’s trajectory itself is one of the main contribution of this work. Our persistent radar object detection algorithm achieved 96.57% precision and 87.08% recall.

Detecting cars in still images is different from detecting lead and preceding vehicles in video. Computation time is an important factor for a system that detect cars in video. Therefore, instead of using a comparatively slower state-of-the-art object detector we have taken a faster but less accurate object detection algorithm. After determining the persistent radar data using it’s trajectory we pass radar objects to the vision algorithm. Only those radar points are considered for further validation by the car detection algorithm which are less than 25 meters away from the host vehicles. In our case, detecting the exact location of the car is not critical. Instead a confirmation on the presence of car is more important. Therefore, if there
is 40% overlap between the predicted box and ground truth box then its a positive detection. Our simple tracking process saves a lot of computing time without effecting the detection results. For detection and tracking of nearby cars (radar range \( \leq 25 \) meters), the vision algorithm gives an average precision (AP) of 77.68\%. The computation time per vehicle is 0.0808 seconds. Lead vehicles detection system achieved a precision of 98.61\% and recall of 90.53\%.

As a whole, the system validates radar objects with vision algorithm, detects vehicles at each time stamp, records epochs of each on-road preceding vehicle’s presence and predicts the lead vehicle even when the radar reading is missed. Experimental results illustrates the effectiveness of the approach. To improve the performance of the system, as a future work, this system can be further extended (1) to take roll, lean angle, and pitch information into account at each timestamp for better object mapping on video frames, (2) to learn models for all types of vehicles, and (3) to estimate lane for more accurate lead vehicle detection.
Bibliography


