Critical Design Review

Perception System Using Stereo Vision and Radar

Team A

Amit Agarwal, Harry Golash, Yihao Qian, Menghan Zhang, Zihao (Theo) Zhang

Sponsored by: Delphi Automotive

December 15, 2016

Abstract:

This report documents and summarizes the progress made by Team Aware (Amit Agarwal, Harry Golash, Yihao Qian, Menghan Zhang, and Zihao Zhang) on the project of developing the perception system using stereo vison and radar for autonomous driving application.

The report starts by presenting an overall description of the project and refined system-level requirements, followed by the updated functional and cyber physical architectures that gives a more detailed depictions of the general structure of the system.

The report also describes and depicts the current status of the system and each subsystem under development. The last part of this report contains the updated project schedule and risk management strategies that we used for keeping the project on track. It also includes a well-scheduled plan for activities to be conducted and goals to be achieved in the next semester.

Table of Contents

1. Project Overview

1.1 Background Information

Current self-driving cars such as those used by Google and Uber have many limitations, especially in their perception systems. These perception systems are bulky, expensive, and hard to maintain, due to the large number of redundant sensors used for perception.

Figure 1. Google Self-Driving Car

Figure 2. Uber Self-Driving Car

On the other hand, this is the self-driving car designed by our sponsors, Delphi Automotive:

Figure 3. Delphi Self-Driving Car

This car is one of the first self-driving cars to make a cross country drive across the US. As can be seen from the above image, this autonomous vehicle has its sensors integrated into the vehicle body, which makes for a sleek design.

What we aim to do in this project is to combine the input from multiple sensors into a standalone sensor system that can be fit to a car and enable perception for autonomous driving purposes. Our perception system will use fewer sensors compared to current driverless cars, and will, therefore, be cheaper and easier to install and maintain. Additionally, our sensors will allow for a low-profile perception system like the one integrated into the car pictured above.

1.2 Project Description:

Stereo vision and radar are two sensor systems that are commonly used for short-range and long-range perception, respectively. In this project, we aim to combine these two sensor systems to develop a stand-alone perception system for use in autonomous vehicles. By doing so, we plan on creating a system that can simultaneously perceive in both the long and short range. Using this, we aim to create a reliable virtual environment around the vehicle that the user can use to gain information about his/her driving surroundings. Eventually, such a system would allow for cheaper, more accessible, and more robust self-driving cars.

2. Use Case

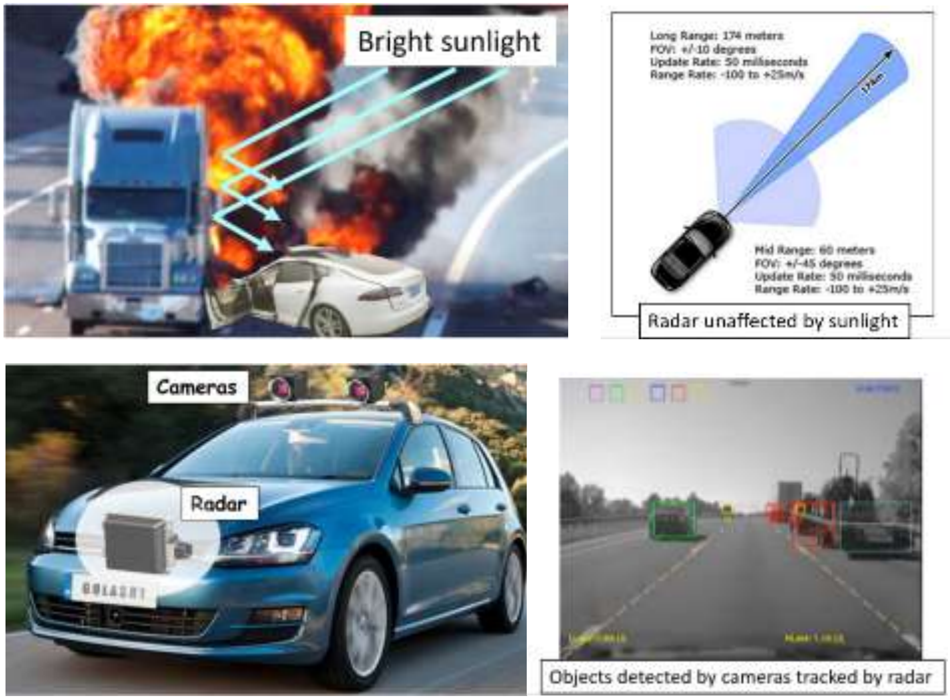


Figure 4. Use Case Depiction

Messla Motors Inc. recently lost one of their autonomous cars. Despite this driverless car being equipped with multiple redundant cameras and LIDAR systems, it crashed into a white truck that reflected sunlight into the car's perception system (top left image). Messla Motors Inc. stock fell sharply, and the CEO of the company, Mr. Dusk, was faced with scrapping its extensive sensor array system project and starting from scratch. Moreover, recent consumer reports had shown Mr. Dusk that a major complaint about their autonomous cars was the bulkiness and difficulty of maintaining the sensor array system. Users were also not happy with the strange and ostentatious appearance of these cars.

The Messla Motors crash scared many small entrepreneurs and aspiring researchers who had intended to venture into the autonomous vehicle market sector. They felt that if a big, well-funded company such as Messla Motors had failed to design a foolproof perception system, then they had no chance of doing so on their limited budget. Consequently, interest and progress in the autonomous vehicle research and development industry started to fall. What were Mr. Dusk and more importantly the small entrepreneurs supposed to do now?

Enter the EagleSense3000 from Team Aware! As can be seen from the top right image above, our perception system uses a radar to simultaneously perceive long and short range objects, at a sweep rate of 20 Hz. By combining both radar and stereo vision technology, we can use our perception system in bright sunlight or pitch darkness. Our radar system can detect virtually all possible obstacles (such as big white trucks) and is unaffected by the ambient lighting. Additionally, the radar unit is much cheaper than a comparable LIDAR unit with similar capabilities. Under favorable conditions, the stereo vision system provides for a very robust means of analyzing our environment in 3D. Since we base our perception system on just three individual sensors – two cameras for stereo vision, and one radar unit – our system does not cost as much as systems like Mr. Dusk's. Our system is designed to be standalone – it does not depend on a car's make or model or on any other perception or actuation systems that a car might have. Moreover, the small form factor of our sensors make for sleek and adaptable perception system that could be used by any manufacturer of autonomous vehicles.

Now that he had scrapped his latest perception sensor array, Mr. Dusk proceeded to try out the EagleSense3000 from Team Aware. He easily installed the sensors into his test vehicle as shown in the bottom left image above. Our sensor system is lightweight, low-profile, and easy to setup – within a couple of minutes Mr. Dusk had already secured the sensors within the car and had connected our computer system to the car's display and actuation systems. Amazed at the ease of installation, Mr. Dusk eagerly entered the car to see our system perform!

As the car drove around the city, the EagleSense3000 astutely and reliably displayed a virtual environment on the car's infotainment screen, while sending outputs to the car's many actuators that enable it to drive. It then started to rain, but our cameras are installed within the car's cabin, and the radar is designed to be weatherproof; this made the rain hardly an issue. Later, the sun came out and shone brightly into the cameras. Once again, this was not an issue for the EagleSense3000 as the radar sensors use radio waves to detect approaching obstacles. Just as the car made its way back to our testing facility, a crazy pedestrian dashed across the road in front of the car. Thankfully, due to our fast sensor sweep and our dual modes of perception, the pedestrian was detected, and the brakes were applied perfectly. Mr. Dusk was completely impressed.

Following the success of Mr. Dusk's testing and review of our system, smaller entrepreneurs and researchers were emboldened to use the EagleSense3000 minimalist sensor system for their projects and autonomous vehicles. Some researchers chose to replace their existing perception systems entirely and use ours instead, while other researchers simply chose to augment their existing system by integrating our system within theirs. As can be seen in the bottom right image above, the stereo cameras and the radar work hand-in-hand to create a sensor system that is full-range and real-time. Moreover, by using stereo vision for depth perception and radar for object tracking, we avoiding using expensive components such as LIDARs. This means that our system is low-cost and can therefore be used safely by limited-budget research and development labs.

3. System Requirements

The system-level requirements are categorized as Mandatory (M) or Desirable (D), as well as Performance (P) and Non-functional (N). The performance requirements and non-functional requirements can be found in Table 1 and Table 2.

We separate our system into four parts, namely the physical mounting structure, the power subsystem, the sensor fusion subsystem, and the perception subsystem. A more detailed description of the system requirements is as the following.

3.1 Requirements on subsystems

3.1.1 Physical mounting structure

(1) Radar: The mounting height of the center of the ESR on the vehicle should be 300 mm to 860 mm above the road surface [3]. The mounting structure should be firmly attached to the test vehicle and robust to the vibrations of the car.
(2) Stereo Vision: Mounting structure for stereo vision is used to fix the position of the cameras. The cameras should be firmly fixed inside the car using an 80-20 aluminum mounting bar with a baseline of 80 cm. The baseline of the stereo vision (camera pitch yaw, and relative translation and rotation) should not change after driving for a long time over varying road conditions.

3.1.2 Power Subsystem

(1) Power booster: Power booster is used to boost the input voltage to achieve certain voltage to power sensors. The power booster should take 12V input from the cigarette lighter, and then output two classes of voltage. 12V for the two cameras and 24V for radar. The power booster should be robust to the disturbances (-2V ~2V) of the input voltage and should provide a stable output voltage for all the sensors.
(2) Power inverter: Power inverter subsystem is used to power the computer. The power inverter should take the power from the battery and convert it to 110V AC voltage. The output power should be larger than 180 Watt in order to power the computer with a high performance core i7 CPU and a Titan X GPU.

3.1.3 Sensor Fusion Subsystem

The sensor fusion subsystem is used to fuse the data from the Radar and the stereo vision. The input of this subsystem are the depth information from the radar and the stereo vision. This subsystem then synchronizes and fuses the input data to a global depth map that shows the distance between the vehicle and the objects in the surrounding. The system should synchronize the data with less than 10ms delay between two kinds of sensors. The system will generate the depth map at 20fps.

3.1.4 Perception Subsystem

(1) Object detection & classification: The object detection and classification subsystem is used to detect the position and classify the interest objects in the image. This subsystem should take the image captured from the stereo vision as input. The output should be the bounding box that contains the object's position, label (which class the object belongs to), and certainty of the object's classification. This subsystem should be robust in weather conditions (including rain, fog, snow, etc.) and intelligent enough to deal with the image with different lighting environments (including strong sun light, daylight, night, etc.). This system should reach 60% object detect accuracy, 80% object classification accuracy, the system should detect and classify the object at a minimum of 20fps.
(2) Stereo vision: The stereo vision subsystem is used for low level 3-D reconstruction or 3-D mapping. The subsystem should take images captured from stereo vision as the

input, and should output the depth information of the object in the common area of two images. The subsystem should take the images captured from the camera at 20hz and then output the corresponding depth information at a minimum of 20fps. The system should also be accurate enough (accuracy higher than 80%) to detect the depth information of the object.

(3) Object tracking:
Object tracking subsystem is used to track the vehicles' and pedestrians' position. This subsystem takes the output from the object detection system as the input. Then using the extended Kalman Filter or Particle Filter to continuously track the position of the object. The system should be robust to object occlusion. The accuracy of the object tracking subsystem should reach at least 70%.

(4) Radar:
Radar subsystem is used to detect and provide the full range accurate depth information (0-160m) of the object [1][2]. The radar's output are the pre-clustered tracking points of the objects. This subsystem should acquire the data at 20fps. This subsystem should also use the extended Kalman Filter to filter out all the noisy points and provide the correct depth information of the object.

3.2 Changes since Preliminary Design Review

1. Object detection accuracy changed from 80% to 60%:

Currently we use the state of the art object detection algorithm called single short MultiBox detector (SSD), and per its result, for pedestrians and vehicles, the object detection accuracy is around 70%. Also, this algorithm has low performance on small objects. According to the author of this algorithm, this problem could not be solved, which would lead to decreased performance in our real test environment.

2. Object classification accuracy changed from 70% to 80%

Using the off-the shelf algorithm, our algorithm could accurately classify the object at an accuracy of around 98% using the test data we captured from our camera, that is the reason why we would like to increase our classification accuracy. Using more image data and data augmentation methods like depth maps, we hope to increase the accuracy.

3.3 Performance Requirements

Table 1. Performance requirement details

ID	Description
M.P1	The system shall conduct full-range perception up to **150**m
M.P2	The system shall acquire sensor data from camera and radar in real-time up to **20**hz

M.P3	The system shall fuse and unify the data from radar and stereo camera up to **50**m
M.P4	The system shall detect objects with an accuracy of up to **60**%
M.P5	They system shall classify objects with an accuracy of up to **80**%
M.P6	The system shall estimate vehicle motion with an accuracy of up to **80**%
M.P7	The system shall detect object velocity with an accuracy of up to **70**%

3.4 Non-functional Requirements

Table2. Non-functional requirements details

ID	Description
M.N1	The system will work in real-time
D.N1	The system will be robust in weather conditions (Including rain, fog, snow, etc.)
D.N2	The system will work properly in most of the human drivable lighting environments (including strong sun light, daylight, night, etc.)
D.N3	The system will be concealable in the car body
D.N4	The system can be integrated into the already existing vehicle display systems

4. Functional Architecture

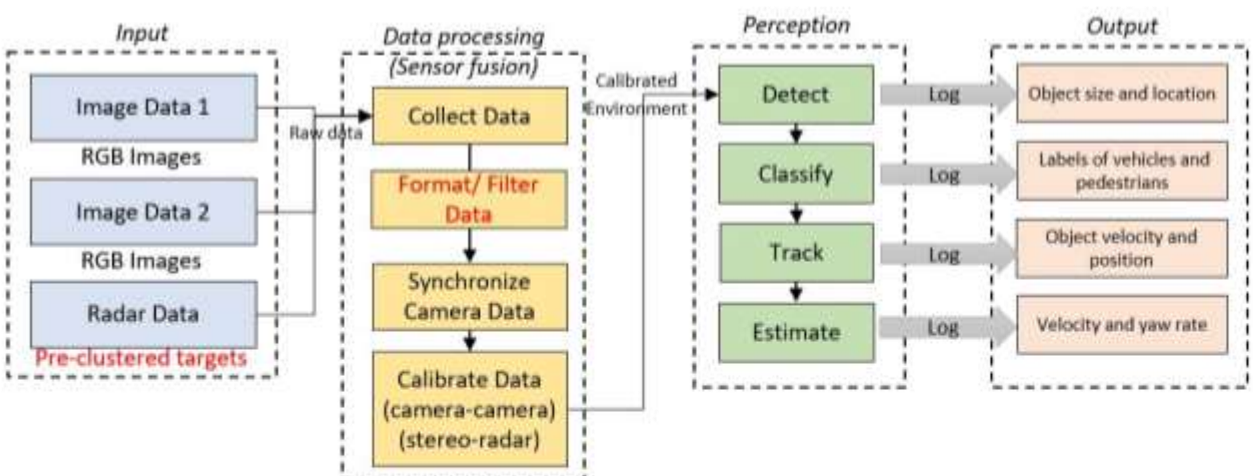


Figure 5. Functional Architecture

The basic structure of our functional architecture has remained the same since the preliminary design review. Regarding updates, a slight change was made to the format of the input data from the radar: thanks to the advanced built-in DSP software with the Delphi ESR 2.5 radar, our system can directly receive the pre-clustered targets instead of raw data points [3]. Furthermore, formatting and filtering data from the radar have been added before the data synchronization and calibration in the sensor fusion module. The team identified it to be a necessary additional function according to the extremely noisy data that we are currently receiving from the radar.

A diagram illustrating the updated functional architecture of our system is shown in figure 5 above. The whole process that transforms raw data from sensors (input) to an interpretation of the targets in the surrounding environment (output) is mainly composed of two steps, namely the data processing (for sensor fusion) and the perception.

4.1 Data Processing

First of all, incoming data from all different sources should be collected together and stored in the system. Noting that the stereo vision system being constructed uses two identical single lens global shutter image sensors instead of an off-the-shelf stereo vision camera, the system receives two different sets of data from the two cameras. Therefore, the cameras need to be synchronized before it can perform any basic stereo vision task. For data from the radar, it has to be first organized into a human readable format before any further processing. Since a majority of the detected targets in the data are random obstacles on street sides, they would not be useful for our application. It is also highly desired to filter out these noise at this stage.

Finally, the time axes of different sets of data need to be aligned and synchronized with each other, since specific applications of the system have suggested relatively high demands on real-time data and its reliability. Processed data from the stereo camera and the radar should also be fused to improve the performance of the perception tasks such as object detection and tracking within 60 meters.

4.2 Perception

In the perception module, tasks can be roughly divided into four categories, namely, the object detection, classification, tracking, and motion estimation. With fused data coming into the module, these tasks will be executed in a sequential order as demonstrated in figure 5. The order is determined based on the dependencies of each task.

The first task is to detect objects of interest in the scene formed by the fused data. Secondly, the detected objects will be classified and identified. The system should be able to distinguish between vehicles and pedestrians by extracting features from image data. Within the group of vehicles, the type of a vehicle should also be recognized. Labels of these classified objects should be displayed in the output of our perception system.

Furthermore, detected objects from the first task should also be identified as either static or moving. Specifically, the system should be able to track the moving objects to investigate their positions and velocities over time. For static objects, their size and location are desired instead.

Again, the desired information from both static and moving objects should be available in the output of the system.

Last but not least, the 3D motion of the perception system, expressed in terms of velocity and yaw rate of the cameras, can be estimated based on results from previous tasks such as object detection and tracking. Since the perception system will be rigidly mounted on the testing self-driving vehicle, the 3D motion of the vehicle can, therefore, be estimated, too.

In summary, raw input data from sensors will be processed first and fused to form a calibrated environment of the perceived scenes in the real world. In this environment, objects should be detected and classified with labels. Their kinematic and spatial information, along with ego-motion of the vehicle, should also be provided. This augmented reality environment containing all the information above will be the final output of the perception system. It can be directly interpreted by an onboard computer to help generate commands to control the vehicle, or displayed to driver and passengers onboard through the vehicle infotainment display system.

5. Cyber-physical Architecture

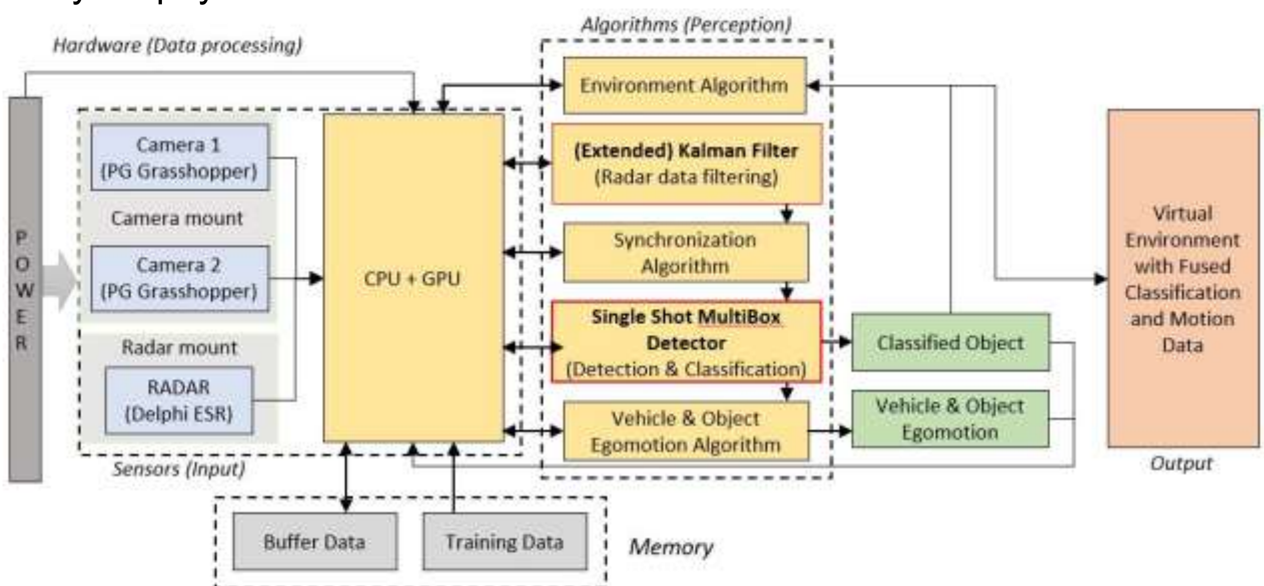


Figure 6. Cyber-physical Architecture

There are two major sections in the cyber-physical architecture, namely the hardware and the software. Due to the focus of the project on building a perception system, all arrows except the ones outwards from power, represent information flow. The arrows outwards from power represent the flow of energy. A diagram illustrating the updated cyber-physical architecture of our system is shown in figure 6 above.

5.1 Hardware

The hardware section, represented on the left, contains the power source, the sensors and the processing unit. The power source supplies energy in the form of electricity to the sensors and the processing unit. Different power sources have been used under different stages of the project. For example, the regular AC power socket of 120 Volts was used for indoor testing of the sensors in the earlier stage, while a DC cigarette lighter receptacle of 12 Volts is currently used more often for outdoor experiments on the testing vehicle. Other more powerful alternatives will also be considered in the future, especially after the more heavy-duty CPU and GPU are installed and used onboard.

The sensors used for building the system include an electronically scanning radar (Delphi ESR 2.5) and a stereo vision system composed of two identical cameras (Point Grey Grasshopper 3), which are mounted separately on the testing vehicle. These sensing units altogether should be able to collect information of object of interest in front of the vehicle in real time. All raw data should be simultaneously passed to the processing unit, where all data processing tasks including filtering, synchronization, and calibration, as well as all software programs for perception tasks, will be implemented at the same time. The processing unit essentially serves as the center of all communications among our standalone perception system.

5.2 Software

The software section, represented on the right of the cyber-physical architecture, contains the algorithms, the memory, the data flowing in and out through software programs based on the algorithms, and the final result. The central processing unit, most likely with the assistance of a separate graphics processing unit, processes the incoming data using the appropriate algorithms. These algorithms in turn yield subsets of data required to the final algorithm, which processes them to form the virtual environment along with all desired information of objects (category, size, relative position, and relative velocity) inside the environment.

In terms of updates on the cyber-physical architecture since the conceptual and preliminary design review, a new component has been added in the software section for filtering data from the radar. The filtering process should take place before the synchronization between the radar and the stereo camera, as it may provide more ease for multi-sensor calibration and further reduce the computational load for sensor fusion. The team is currently considering applying the extended Kalman filter (EKF) for this task, as it is the most widely used algorithm for driving-related object detection and tracking according to our ongoing literature review.

For object detection and classification, the state-of-the-art Single Shot MultiBox Detector, commonly known as the SSD algorithm, has been implemented this semester. The SSD algorithm involves the use of deep neural network, which is able to achieve both object detection and classification at the same time. Based on the decent detection rate and accuracy it can provide, it is currently considered as our best choice for object detection and classification and will most likely be still used in our final system.

6. Current System Status

6.1 Targeted System Requirements

Below are the functional and performance requirements that the team aimed to achieve by the end of the Fall semester of 2016:

Functional:

- The system will use multiple sensors
- The system will be self-contained

Performance:

- The system shall detect objects (pedestrians & vehicles) up to 40 m
- The system shall acquire sensor data at up to 20 Hz
- The system shall detect object distance (stereo vision) with an accuracy of up to 70%
- The system shall detect object with an accuracy of up to 60%
- The system shall classify objects (pedestrians & vehicles) with an accuracy of up to 80%

In the following paragraphs, targeted requirements will be categorized specifically for each involved subsystem of our overall perception system.

Physical Mounting Structure

Targeted design requirements:

- All sensors shall be rigidly attached to the testing vehicle
- The radar should be attached 30mm to 86 mm above road surface in front of the vehicle
- The stereo vision system should remain functional under various weather conditions including moderate rain and snow.
- The baseline of the stereo vision system should be fixed under normal driving conditions

Power Subsystem

Targeted design requirements:

- The subsystem shall be able to power the two cameras and the radar simultaneously
- The subsystem shall provide stable and consistent DC voltage of 24V to the radar
- The subsystem shall provide stable and consistent DC voltage of 5-12V to the cameras
- The subsystem shall provide proper protection to the sensors

Sensor Fusion Subsystem:

Targeted design requirements:

- The system shall detect the depth with an accuracy up to 80% (for the stereo vision)

- All sensors shall be able to acquire data at up to 20 Hz

Perception Subsystem:

Targeted design requirements:

- The system shall detect size of objects of interest with an accuracy of up to 60%
- They system shall classify objects of interest with an accuracy of object up to 80%
- The system will work in real-time

6.2 Subsystem Descriptions/ Depictions

The current overall system can be roughly divided into two parts: the hardware and the software. The hardware system contains power supply, two Point Grey cameras, and one Delphi ESR radar. The software system contains the functions and algorithms for stereo vision, object detection, and radar processing. The basic structure of our current system is shown as the following:

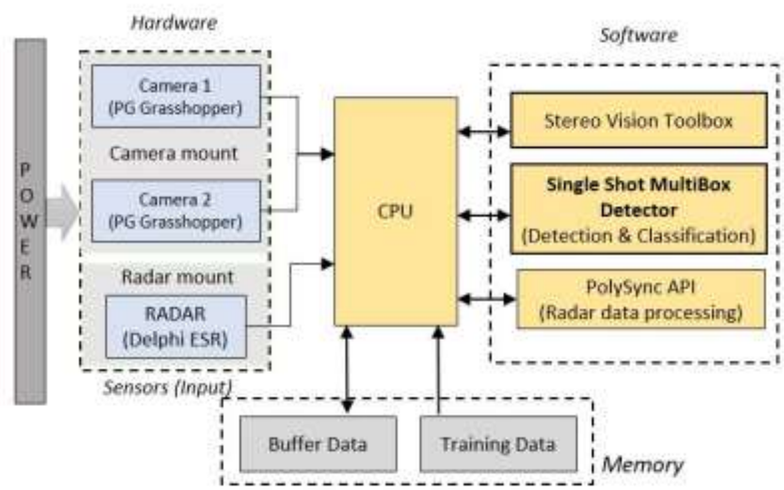


Figure 7. Current System Status

6.2.1 Sensor Mounts

All sensors for building our perception system, which include two Point Grey Grasshopper cameras for stereo vision and one Delphi ESR 2.5 radar, are successfully mounted on the testing vehicle.

Considering other crucial tasks that demanded more progress and the amount of time already consumed in designing and fabricating the water-proof camera housing, the team decided to pursue a different plan from our initial mounting solution: instead of fixing both cameras on the mounting rack above the roof of the testing vehicle, the cameras are now fixed in front of the windshield inside the vehicle. More specifically, the current camera fixture is composed of two side supports and an 80/20 aluminum beam. The supports are installed on the testing vehicle at the location where the two sun visors were originally attached to the vehicle. Therefore, both sun visors and their fixture were removed in advance. Currently, each camera is mounted inside our customized camera housing, and the two housings are rigidly attached on the 80/20 beam hung over the dashboard by the two supports. The picture below demonstrates the on-board setup of the cameras that is illustrated above:

Figure 8. On-board setup of the cameras

Note that the current mounting structure of the cameras is placed inside the vehicle, it eliminates the need for developing a more advanced weatherproof camera housing while still satisfying the weather-proof requirement of our perception system at the same time.

As for the onboard setup of the Delphi ESR radar, it is mounted in front of the vehicle's grille which is about 50 cm above the ground, right on the vehicle centerline. Recalling from the Delphi ESR user manual that the radar's mounting location should be between 300 mm and 860 mm from the road surface, we figured out that the current mounting location would be the ideal one after taking the front exterior of the testing vehicle for reference.

The radar is mounted on the grille using three identical fixtures that are self-designed and 3D-printed. Each fixture can be inserted into the vehicle grille. The two hooks on each fixture can clamp the fixture to the grille from the back. The radar can be then attached rigidly with the fixtures using M6 screws and locknuts so that it is also attached rigidly on the grille and the vehicle.

Figure 9. On-board setup of the radar

Figure 10. CAD model of the radar fixture

6.2.2 Power Source

The team has successfully fabricated a printed circuit board (PCB) for distributing power to the sensors on the testing vehicle, as shown in the picture below. Its functionality has also been tested in the lab.

Figure 11. Power distribution PCB

The vehicle's battery supplies the power input to the PCB through the cigarette lighter socket, which can provide a DC voltage at 12 volts. The PCB can distribute the power to the two cameras at the same voltage. It also contains the power boosting feature, which converts the voltage from 12V DC to 24V DC to power the radar. To protect the sensors from unnecessary damage, fuses and diodes with appropriate values are applied at each output port to prevent overpowering the sensors beyond their limits. Specifically, a fuse with a rated current of 2A and a fuse with a rated current of 500mA are used for the cameras (with the operational current around 1.5A) and the radar (with the operational current around 350mA), respectively.

Note that the design of this PCB does not take the power consumption of any additional device to the current system into consideration, for example, a computer with heavy-duty processing units (CPU and GPU) that might need to be equipped on the testing vehicle for more advanced perception tasks in the future.

6.2.3 Delphi ESR 2.5 Radar

The team can successfully receive data from the Delphi ESR 2.5 radar on the computer. On the hardware side, the Kvaser Leaf Light V2 device is used to establish communication between the radar and the computer through a USB connector so that the Controller Area Network (CAN) messages can be converted and recognized by the computer. On the software side, the incoming messages from the radar to the computer are processed using the PolySync middleware platform.

The real-time data from the radar (in the form of pre-clustered targets) can be visualized in 3D space using the PolySync Studio, as shown in the picture below. It can also be recorded and stored similar to a ROS bag file. In addition, by exploring the PolySync API documentation online, useful information regarding each detected target, such as the tracking status, the x-y-z position and velocity, the range/range rate, and the amplitude, can be now extracted for more detailed analysis and further data processing.

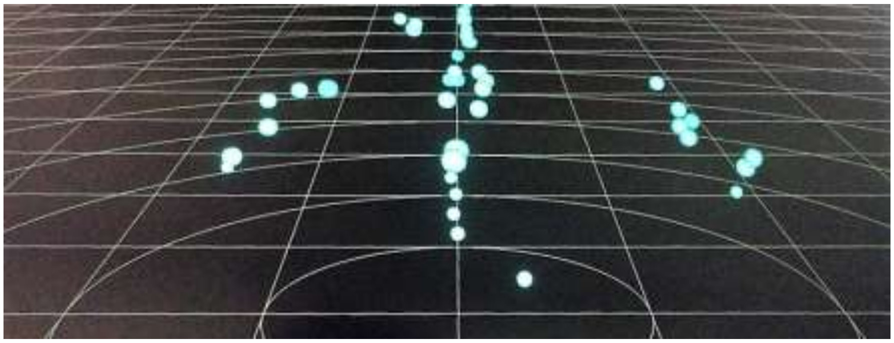


Figure 12. 3D visualization of data from Delphi ESR 2.5 Radar

Currently, the team is working on finding the proper method to filter out the consistent noise (useless detected targets) in the real-time radar data. Meanwhile, software programs are also being developed (similar to the subscriber in ROS) on the PolySync platform to implement the filtering algorithms and fetch only the desired data to interact with data from the stereo camera for system's perception tasks.

6.2.4 Stereo Vision

Our stereo system is designed with a long baseline (about 80 cm) in order to ensure accuracy of the depth calculation at relatively long distance (around 60 m). We also use advanced calibration method to compute the extrinsic and intrinsic parameters of the cameras. Our system is currently able to achieve around 88% accuracy. More details will be discussed in the two following sections:

Stereo Vision Calibration:

The purpose of this subsystem is to compute the intrinsic, extrinsic, relative rotation and translation of two cameras. The flow chart below shows how the stereo vision calibration algorithm works.

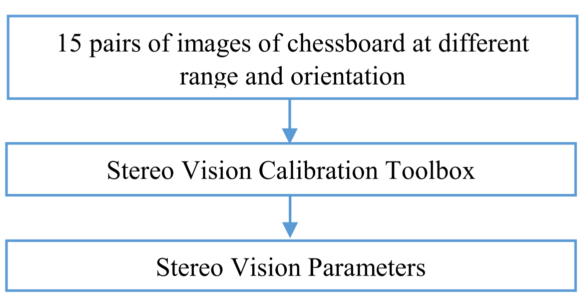


Chart 1. Stereo Vision Calibration

The Stereo Vision toolbox from MATLAB is used to compute those parameters. The toolbox takes several images pairs of a chessboard as the input, then it computes an optimized estimate of the mentioned parameters.

Stereo Vision Depth Computing:

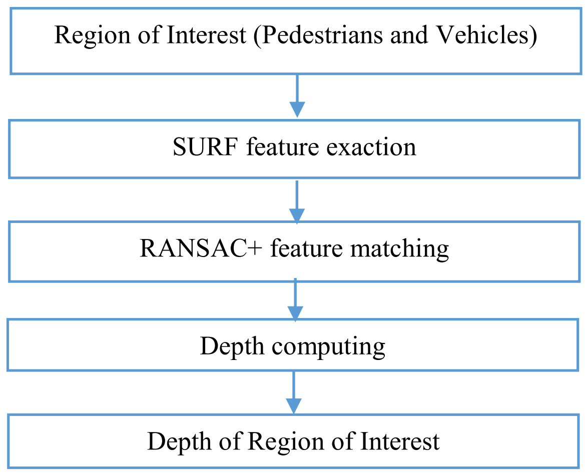


Chart 2. Stereo Vision Depth Computing

The stereo vision system takes the output from the object detection algorithm and two images captured from the stereo vision as input. Using the SURF feature detector, the system can detect high-quality, unique features in both images. Then we combined RANSAC with a feature matching algorithm in order to get the correct matching pairs of features. Then the subsystem can use the triangulation method in order to compute the depth of the object.

6.2.5 Perception

The algorithm that we are using is called the Single Shot MultiBox Detector (SSD).

At current stage, we have compared the performance of two state-of-the-art object detection algorithms, namely the Faster R-CNN and the Single short MultiBox Detector. In terms of accuracy, these two algorithms are similar. In terms of FPS, the Single Short MultiBox Detector (SSD) is able to reach 50-60 fps, while the Faster-R-CNN can reach only 5 fps using a Titan X GPU [5][6]. For this reason, the SSD algorithm is considered to be the best option for our object detection and classification tasks.

The Single Shot MuliBox Detector (SSD) is a convolutional neural network, which is composed of a VGG-16-Net, a fully connected layer, and a non-maximum suppression layer [6]. The input to the network is the image captured from the camera. The outputs are:

1. The bounding box for each object of interest that describes the relative position of the object.

2. The category that the object may belong to.

3. The probability that the object may belong to this category.

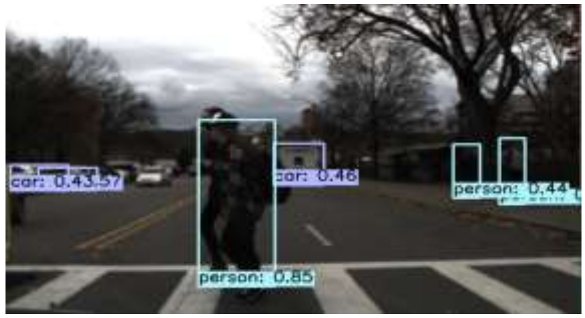


Figure 13. Object Detection and Classification using SSD

This subsystem is currently being tested on a computer with I5-4600U processor, 8G RAM, GT740M graphic card. The fps of the SSD is 2 fps, which is 40 times faster than the Faster R-CNN (20s/image). With high performance GPU, such as Titan X, it could reach 50-60 fps. The object detection accuracy of our test set is around 60%, the classification accuracy is around 90%.

6.3 Modeling, Analysis, and Testing

The following modeling, analysis, and testing tasks have been performed during the past semester:

1. All basic components of the physical mounting structure for our perception system have been modeled using the Computer Aided Design software (SolidWorks).

   These mounting components were modeled by carefully examining components' dimensions on the testing vehicle. A more detailed discussion about the modeling process is included in the previous section (6.2.1 Sensor Mounts) for reference. These CAD models have also been fabricated and used to mount the two cameras and the radar on the testing vehicle.

2. The baseline for our stereo vision system has been determined through calculation and experiments.

   The experiments were conducted in the tunnel right outside the lab. The two cameras were mounted at appropriate, fixed height and orientation, similar to their mounting configurations in the testing vehicle. According to the design requirements for the stereo vision system, it should be able to provide enough resolution in order to detect objects of interest within 60 meters in front of the vehicle. Therefore, a checkerboard with the appropriate size was placed at around 60 meters in front of the vehicle as the target.

   The two cameras were placed in parallel with different baselines (one baseline for each trial). For trial, calibration of the stereo vision was performed by taking advantage of the Stereo Vision Toolbox in MATLAB. Then, the depth of the target (checkerboard) was calculated using the triangulation method. The accuracies of the depth information collected from all trials were then compared and analyzed. The final result suggested a baseline of approximately 80 centimeters in order to achieve decent accuracy of depth (80%) of objects within 60 meters.

3. Several on-road testing has been conducted to verify effectiveness of mounting design and basic sensor functionalities.

   After setting up all the sensors on the car, the team went out to test the basic functionality of the sensors and to make sure the mounting method was robust.

   First of all, the camera's resolution test in adverse weather conditions and lighting conditions was tested. The team drove the car around the campus during light snow, moderate rain, sunset (when sunlight falls directly on the camera's lens), and night with differing speeds of the vehicle. It turned out that the cameras were still able to yield high-resolution pictures that could be used to detect objects in most of the cases. Then, the two cameras were calibrated in order to get basic parameters for the stereo vision system. For the radar, it was able to respond to the changing environment and collect data in real-time. The detected targets (including both objects of interest and noise from the surroundings) were visualized in 3D space using the PolySync middleware. The relative position, range information, and tracking status were also investigated through the same method.

4. Performance of the object detection and classification was analyzed using the testing data.

   After testing all sensors' functionality and verifying that the mounting method is robust, the team used the data collected during the outdoor experiments to build the stereo vision and performed object detection and depth calculation. Using the images captured by both cameras at the same time, the team was able to build the stereo vision system that can provide depth information of the interest points with an accuracy above 80%. Using the video recorded by the cameras, the object detection for pedestrians and vehicles is above 60% (shown in figure 13). These results indicate the satisfying performance of our stereo vision system. For the radar, it can provide the distance of a given object of interest (manually identified from all detected targets) with an accuracy of 80% (shown in figure 12).

6.4 FVE Performance Evaluation

There are four sections included in the Fall Validation Experiments, which cover the physical mounting structure, the stereo vision performance, the sensor synchronization, and the object detection. The team has achieved (fully achieved for three experiments and partially achieved for one experiment) all the requirements mentioned in the FVE plan. The detailed performance for the four experiments is illustrated in the tables below.

Table 3. Performance of Fall Validation Experiment A

Subject	Goal
Sensor mounts	Show robustness of sensor rack and mounting method **(achieved)**

Actual performance	All sensors were rigidly attached on the testing vehicle. The relative position of sensors changed by less than 5 mm in any direction after 20 minutes of test drive

Table 4. Performance of Fall Validation Experiment B

Subject	Goal
Stereo Vision	Stereo Vision can work in adverse weather condition and give depth information above 20% accuracy **(Achieved)**
Actual performance	Stereo vision can give the depth information of objects with an accuracy of > 80%

Table 5. Performance of Fall Validation Experiment C

Subject	Goal
Synchronization	Show synchronization between the two cameras. **(Achieved)** Show synchronization between the stereo vision system and the radar **(Need improvement)**
Actual performance	Both cameras were triggered at the same time with less than 1ms difference Less than 0.1 s difference between timestamps of the cameras and radar Depth info of the same (static) objects was acquired from both the cameras and radar

Table 6. Performance of Fall Validation Experiment D

Subject	Goal
Object detection	Object detection algorithm can detect vehicles and pedestrians with an accuracy of 60% **(Achieved)**
Actual performance	Detection results with accuracy of 63% for vehicles, 44% for pedestrians. Classification accuracy was above 97%

6.5 Strong/ Weak Points

Strong points:

- Robustness of sensor mounts

  Our hardware system is robust. After repeated outdoor driving tests in various road conditions and weather conditions, the relative position between the sensors stayed the same. The effectiveness of our current mounting solution can provide a firm and reliable basis for all the remaining on-road perception tasks to be conducted during next semester.

- Object classification accuracy

  The object classification accuracy is above 95%, which exceeds our expectations. This will help in developing a more accurate perception system.

- Stereo vision accuracy

  Stereo vision system can give the depth information of the objects of interest with an accuracy of around 88%. When combined with the radar, the accuracy of depth of the objects may be further improved after the multi-sensor calibration.

Weak points & Refinements:

- Noisy Radar data

  The radar still gives some noisy data even if the testing environment is an empty garage. This may confuse the system to give wrong estimate of the objects and their positions. In the future, the team is considering using an extended Kalman Filter (EKF) to filter the noisy points in order to extract useful information from the radar.

- Stereo vision & object detection latency

  Currently without a high-performance GPU, it takes longer than estimated to build the stereo vision and perform object detection and classification tasks in real time. Since the system shall work in real-time as it is specifically designed for autonomous driving related application, latency might cause a major problem. In the future, the team is planning to get a high-performance GPU along with the CPU to solve the latency problem.

- Synchronization between radar and stereo cameras

  To use both radar and stereo cameras together for the perception system, two sensors must be properly synchronized in order to detect and give correct information of the same objects at the same timestamp. For now, the team has completed the synchronization between cameras, but there is still work needed to be done on the synchronization between the radar and stereo cameras starting from next semester.

7. Project Management

7.1 Work Breakdown Structure

Figure 14. Work Breakdown Structure (WBS)

The overall flow of the Work Breakdown Structure (WBS) for this project has not changed much since the beginning, and this is because this project primarily involves many stages of improvement of our initial requirements. Moreover, our requirements have not fundamentally changed. In the WBS above, the items are listed based on our deliverables for this project. Items highlighted in green are completed items, and items in yellow are ongoing tasks that we hope to complete in the spring semester.

The cameras have been synchronized with each other such that we can simultaneously acquire pictures from both cameras without any time delay. This was accomplished by using a hardware trigger using an Arduino, which was interfaced with a GUI made in C#. We have successfully gathered real-world data, and we have used that data for object detection and classification. Currently, we are working towards optimizing the efficiency of our algorithms and our computing speed.

The radar data that we currently have in noisy and will require appropriate filtering before we can rely on it for object tracking (in tandem with the stereo vision tracking). We will also need to synchronize the data from the radar with the images taken from our stereo cameras; currently, the data acquired from the cameras and radar are synchronized via software to within a 100 ms maximum delay.

Initially, the rack system for our sensors was to be mounted on the roof of the car. However, we decided that weatherproofing the camera enclosures to industry specifications (say IP 57) was too time-consuming and detracted from the key focus of this project. Currently, the cameras and mounted within the car's cabin, so the enclosures do not require any weatherproofing. The radar unit is weatherproof by construction, and it is mounted on the grille of our test vehicle using custom-made 3D-printed mounts.

The power supply PCB that we currently use for this project converts the 12 V DC input from the car's battery into dual 12 V DC channels for the cameras, and a single 24 V DC channel for the radar. It is essentially a DC-to-DC step-up converter. Further testing of our PCB will be required (testing under various loads) to verify its robustness before we install it in the vehicle.

Currently, we implement object detection and classification at around 4 FPS using the Single Shot Multi-Box Detector (SSD) algorithm running on an average laptop. In the spring we will work on choosing faster algorithms and computing hardware to improve our performance and make our perception system work in real time.

Our project management solution consists of managing our work done via a master schedule, managing risks via risk detection and mitigation, and managing budget via accurate accounting. We hope to keep this up in the spring semester as well.

7.2 Schedule

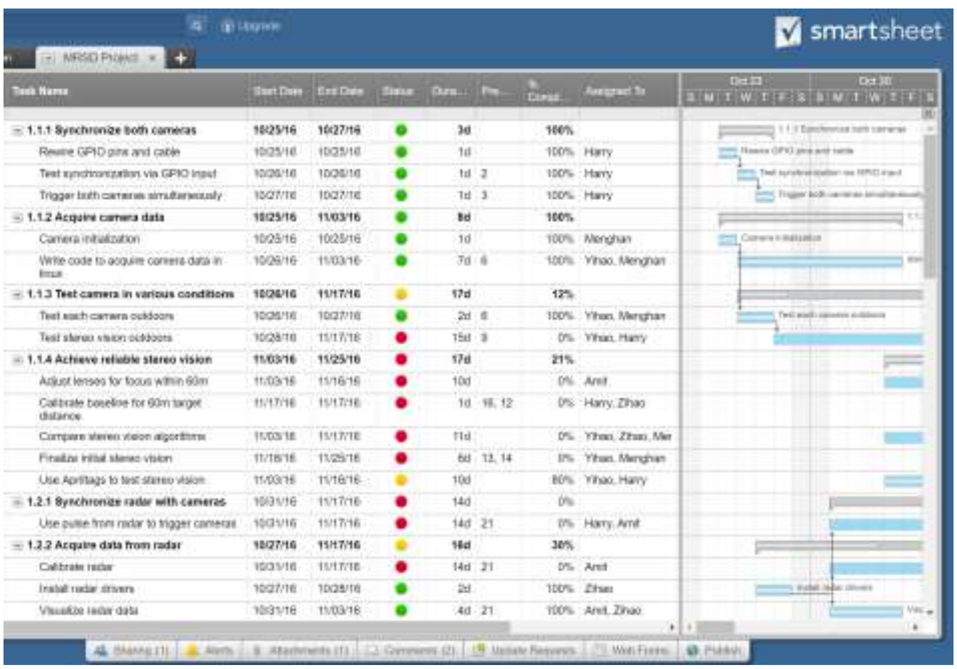


Figure 15. Screenshot of the past semester's schedule on Smartsheet

Our team used Smartsheet this semester to create a master schedule. We were able to assign tasks to team members, track task progress, and create dependencies, as can be seen from the screenshot of our early project breakdown above. The complete schedule and Gantt chart that we created and followed is quite large and would take several pages.

Overall, we are slightly ahead of schedule at the end of this semester. For the spring semester, we aim to accomplish the following tasks on a tentative bi-weekly schedule as follows:

Jan 30 – Begin testing parallel processing on GPU using CUDA.

Feb 14 – Object detection using parallel processing. Clean up radar data using filtering.

Feb 28 – Demonstrate basic object tracking.

March 14 – Object detection and tracking in real time. Enhance object classification if necessary.

March 28 – Proper, reliable sensor fusion in real time

April 14 – Estimate vehicle ego-motion based on vision and radar tracking

April 28 – Improve overall system accuracy and reliability. Further testing and tuning.

This semester we were able to complete all the hardware setup and calibration that we had planned to do. Additionally, we got started on object detection and classification using our stereo vision system, which we had originally planned to do in the spring semester. At this point, we are able to create 3D depth maps using our stereo cameras. We are unable to clearly track targets using the radar, but this is because the data we are obtaining is fairly noisy. Going forward, we intend to improve the speed and accuracy of our perception system using better algorithms, tuning, and better computing hardware. We hope to complete most of our objectives by mid-April and then spend the remaining time fine-tuning and improving our system.

7.3 Test Plan

The milestones that the team plans to achieve by each progress review during the Spring semester are shown in the table below:

Table 7. Project milestones during the Spring semester

Name	Deliverable Functionality	Method to Test
A. PR 7: Late January	Radar data filtering	Be able to show the sensible data from radar and get the distance information of the objects
B. PR 8: Mid-February	Sensor synchronization	Be able to get the data from both stereo cameras and radar in same time
C. PR 9: Late February	3-D reconstruction	Get point cloud data from radar and stereo vision of camera and give a 3-D map of the interests points in the environments
D. PR 10: Mid-March	Sensor fusion	Used the synchronized data from sensors and give the depth information of the objects in the same time
E. PR 11: Early April	Object detection & tracking	Identify objects in the pictures
F. PR12: Mid-April	Parallel computing	Give the labels of the objects detected by the sensors

The following tables and text illustrate a detailed plan of the Spring Validation Experiments (SVE) for our project:

Location: Streets around the CMU campus
Equipment: Test vehicle, two Point Grey Grasshopper3 cameras, one Delphi ESR 2.5 radar, camera housings, camera housing fixture, radar mount, power converter
Environment: Streets at least 15 m wide with about 30 cars/min traffic flow in favorable weather conditions

Table 8. Planned Spring Validation Experiment A

Experiment:	**A**	
Description:	Sensor Fusion	
Step:	**Step description**	**Success condition**
A.1	Mount all sensors on the test vehicle and drive around for 20 minutes. Monitor different sets of data collected by the stereo cameras and the radar	Data from stereo camera and radar should be synchronized and displayed in real-time (less than 100ms delay). Perception results should be formed using data from both the stereo cameras and the radar

Table 9. Planned Spring Validation Experiment B

Experiment:	**B**	
Description:	Object detection and classification	
Step:	**Step description**	**Success condition**
B.1	Drive around for 20 minutes. Detect objects of interest on the road and return their sizes, relative position, and velocity. Classify objects and give the labels. Display the information on a screen	Using the LIDAR data provided by Delphi as ground truth, the system will detect object size with an accuracy of > 60%, distance with an accuracy of > 60%, classify objects with an accuracy of > 80%

Table 10. Planned Spring Validation Experiment C

Experiment:	**C**	
Description:	Object tracking	
Step:	**Step description**	**Success condition**
C.1	Drive around for 20 minutes. Track the vehicle and pedestrians on the road and display the information on a screen	The system shall be able to track objects and give the bounding box of objects with an accuracy of > 60% and give the velocity information with an accuracy of >60%

Table 11. Planned Spring Validation Experiment D

Experiment:	D	
Description:	Vehicle Egomotion	
Step:	**Step description**	**Success condition**
D.1	Drive around for 20 minutes. Estimate the test vehicle's velocity, yaw rate, and direction using tracking data from the cameras and radar.	The system will be able to estimate the test vehicle's motion with an accuracy of > 60%

The plan for the team's Spring Validation Experiments are briefly described above. We have set up four experiments to test the functionality of our perception system and establish its performance benchmarks.

In Experiment A, we will use a common software application or GUI to acquire and log the data from the cameras and the radar. Currently, the cameras are already being triggered simultaneously using a hardware trigger. To get the radar data that corresponds to the timestamps of the camera images, we can use software filtering using a custom application or using PolySync to synchronize the data to within 100 milliseconds. We can test this by logging our data and then acquiring the time discrepancies between the radar and camera data.

In Experiment B, we will detect objects of a measured size at a measured distance. Our system shall be able to return size and distance information from the detection bounding box (see Fig 1.) to greater than 60% accuracy, at a range up to 100 m. Object classification accuracy will have to be manually determined by hand-labelling of the image sets acquired by the cameras, and we will check to see that at least 80% of the objects we detect are classified correctly as car / bus / pedestrian etc.

In Experiment C, we will have cars of a known velocity drive in front of our test vehicle. We will treat the speedometer data as ground truth, since modern cars have fairly accurate speedometers (optionally, we could use Delphi's LIDAR system for ground truth). Our perception system shall use an efficient tracking algorithm to display the velocity of the vehicle being tracked to at least 60% accuracy.

Using data from the radar and the stereo cameras (combined), our perception system shall be able to estimate the ego0-motion of our test vehicle (yaw rate, velocity, roll) to at least 60% accuracy. We will use the vehicle's speedometer, an accelerometer, and/or an IMU to establish ground truth for this test.

Our sponsors initially suggested that we treat the data obtained from their extensive perception system on their test vehicle as our ground truth. If their vehicle (or it's sensor readings) are made available to us in time for our testing, we can use their data for ground truth as an alternative.

7.4 Budget

Table 12. Updated budget

No.	Description	Cost	Quantity	Sponsored	Total
1	Grasshopper3 3.2 MP Color USB3 Vision (GS3-U3-32S4C-C)	$ 975	2	Yes	0 (~~$ 1950~~)
2	Delphi ESR 2.5 24V Radar	$ 3300	1	Yes	0 (~~$ 3300~~)
3	Tamron M118FM08, 8mm, 1/1.8", C mount Lens	$ 210	2	Yes	0 (~~$ 420~~)
4	Thule 53” Aeroblade	$ 570	1	No	$ 570
5	UINSTONE 150W Power Inverter	$ 16	1	No	$ 16
6	Belkin 6-Outlet Surge Protector	$ 10	1	No	$ 10
7	Step-Up Circuit PCB (12V to 24V)	$ 15	1	No	$ 15
8	Electromagnets and chargers	$ 25	1	No	$ 25
9	Kvaser CAN connector and adapter	$ 380	1	No	$ 380
10	Mounting Rack Material (McMaster-Carr)	$ 81	1	No	$ 81
Total					$ 5000 - $ 1097 = $ 3903

The table above shows the team’s updated budget. There has been no incurred cost since the Preliminary Design Review. Excluding the two Points Grey cameras, the two camera lenses, and the Delphi ESR radar that were purchased and provided by our sponsor, the team has spent $ 1097 out of the total budget of $ 5000. The main incurred costs are from the Thule Aeroblade (52%), which were originally planned to be our mounting rack, and the Kvaser device (35%), which is for establishing the connection between the radar and computer. Note that the team has changed the mounting solution and will no longer need the items intended for building the original mounting structure (marked in red in table 12), these items can be sold back to the market to further reduce our current budget.

7.5 Risk Management

The team identified three major risks that could hamper the team's progress. Two of these risks were presented at the Preliminary Design Review and have been tracked and updated since then. The risks are discussed in the following paragraphs.

The first risk is the lack of a wide dynamic range of the camera. This risk is considered to impact the project from a technical, schedule and a cost point of view. The risk is defined well in the risk management chart as shown below. Being able to capture clear images reliably at all times of the day is vital to the success of the project. The likelihood of this risk has been reduced to 10% after the mitigation strategies listed were applied. The cross mark in gray in the likelihood-consequence table stands for the status of the risk identified before the Preliminary Design Review. The cross mark in white in the same table stands for the status of the same risk identified currently.

Table 13. Risk Management Table A

Risk title:	Risk owner:	Date submitted:	Date updated:
Dynamic Range of Camera	Amit Agarwal	October 20, 2016	December 12, 2016

Description of risk:
The Dynamic Range of the camera may not be sufficient for the camera to work in all lighting conditions.

Consequences if risk is realized:
A new and better camera with higher dynamic range will be needed. This will halt development on most activities related to stereo vision which will lead to schedule problems as well.

Risk type
X Technical
X Schedule
X Cost
□ Programmatic

Risk Reduction Plan Summary

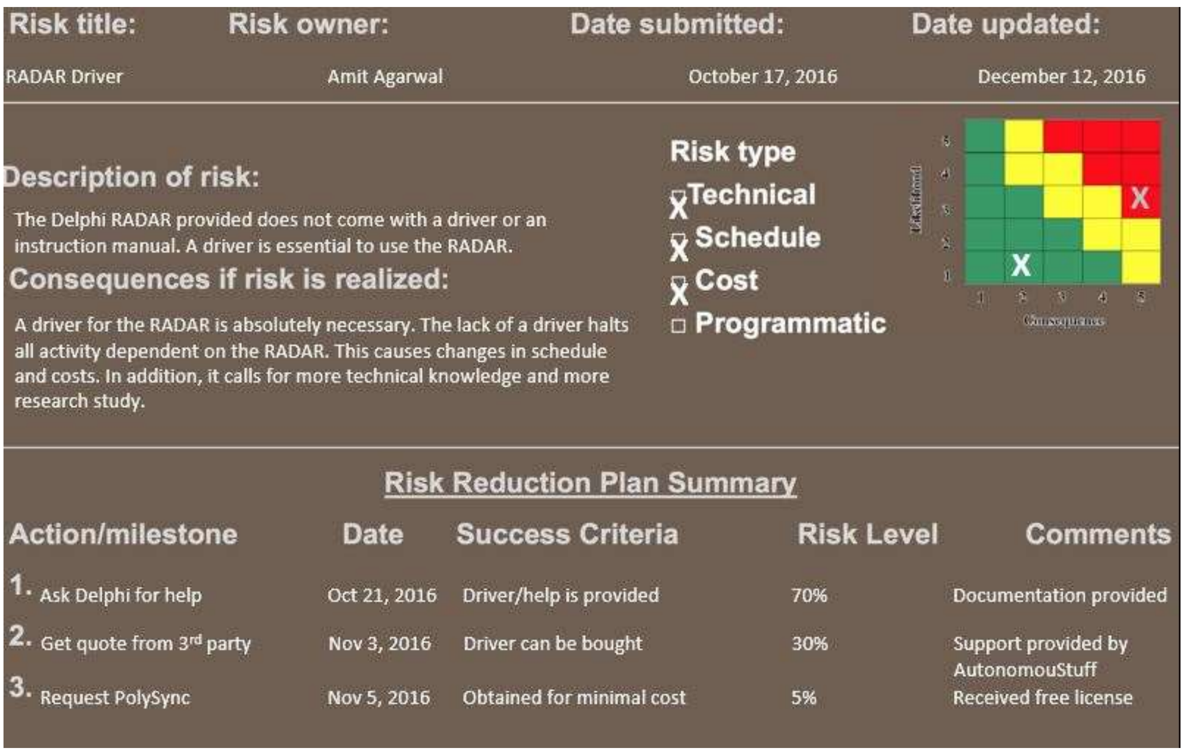
Action/milestone	Date	Success Criteria	Risk Level	Comments
1. Test in afternoon & night	Nov 3, 2016	Produces reasonably detailed images	10%	Object Detection can be performed
2.				

The second risk relates to the problems faced in acquiring data from the Radar. Acquiring data from the radar is instrumental in performing sensor fusion between cameras and the radar. Initially the team encountered great difficulty making progress regarding the Delphi ESR radar. This is mainly because that this radar model is a relatively new product in the market and is specifically designed for use in automobiles; its state-of-the-art technology as well as limited field of applications both result in limited resources we could potentially find in the literature. In addition, the fact that many of these resources are considered as confidential by the manufacturers further adds the difficulty to find any useful information on the Internet.

After receiving the user manual from our sponsor, the team learned the proper startup process for this specific radar model. Also with the help from companies in the commercial sector such as AutonomouStuff, the team identified and purchased the right tool to establish the connection between the radar and computer. Currently, the team is taking advantage of the PolySync middleware platform to process data from the radar and has been making decent progress.

Therefore, after the three steps of mitigation strategies, the risk has been minimized to a 5% likelihood as detailed in the risk management chart shown below. The cross mark in gray in the likelihood-consequence table stands for the status of the risk identified before the Preliminary Design Review. The cross mark in white in the same table stands for the status of the same risk identified currently.

Table 14. Risk Management Table B



Risk title:	Risk owner:	Date submitted:	Date updated:
RADAR Driver	Amit Agarwal	October 17, 2016	December 12, 2016

Description of risk:

The Delphi RADAR provided does not come with a driver or an instruction manual. A driver is essential to use the RADAR.

Consequences if risk is realized:

A driver for the RADAR is absolutely necessary. The lack of a driver halts all activity dependent on the RADAR. This causes changes in schedule and costs. In addition, it calls for more technical knowledge and more research study.

Risk type

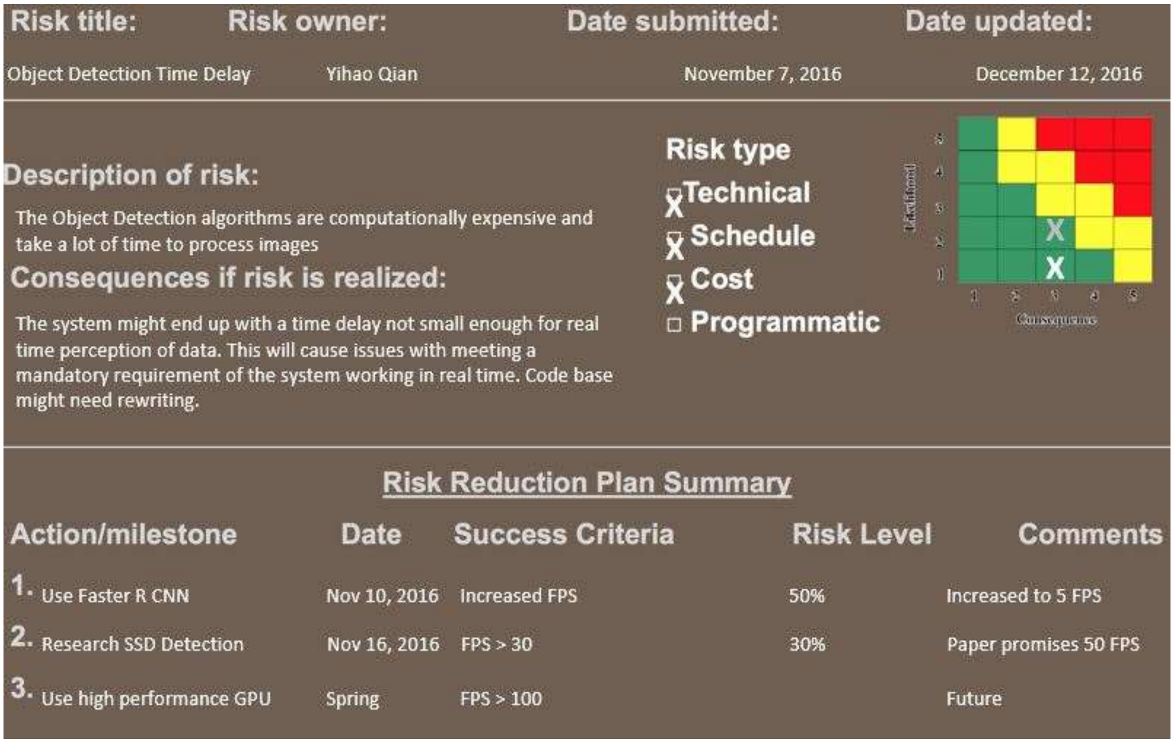
- X Technical
- X Schedule
- X Cost
- ☐ Programmatic

Risk Reduction Plan Summary

Action/milestone	Date	Success Criteria	Risk Level	Comments
1. Ask Delphi for help	Oct 21, 2016	Driver/help is provided	70%	Documentation provided
2. Get quote from 3rd party	Nov 3, 2016	Driver can be bought	30%	Support provided by AutonomouStuff
3. Request PolySync	Nov 5, 2016	Obtained for minimal cost	5%	Received free license

The third risk is new and relates to performing object detection at a fast enough rate for real time processing. This risk was noticed when the team realized that the Faster R-CNN algorithm we planned to use was giving us approximately 0.2 fps which is unacceptable for a real-time perception system for an autonomous vehicle. This risk has been minimized to 30% likelihood by the use of a faster algorithm. A third mitigation strategy will be applied in the spring to further better the frame rate of the algorithm using high-performance GPUs. This risk is also detailed in the risk management chart shown below.

Table 15. Risk Management Table C



8. Conclusion

8.1 Lesson Learned

Through individual work as well as team collaboration on the project this semester, every team member has learned a lot from both the technical and non-technical perspectives. Some key lessons that we have learned from this semester's experience are summarized as the following. We will also keep them in mind to help further improve the team performance and productivity in the Spring semester.

1. Team communication is important

There have been a few misunderstandings between teammates since the team was formed initially. The team climate was freezing at the beginning so that the progress is a little bit behind the schedule. But as time went on, teammates got to know each other better through more personal interaction. Based on the increasing understanding, team members also figured out how to communicate with each other more effectively to make the collaboration more efficient and effective. It turns out that active communication is really helpful.

2. Resources from commercial sector can be extremely helpful

After receiving all the sensors, the team was stuck on the radar for a long time, due to the limited public resources and difficulty for our sponsor to provide much information for various reasons. As a result, it was extremely hard to learn where to start and how to get the data properly from the radar; for a long time, our team could not get useful data from radar. However, after interacting with several other companies from the industry who have worked on the Delphi ESR 2.5, the Autonomoustuff was willing to offer some help and guidance on the setting up the radar. The team also tried to reach out the company called PolySync, whose service is currently used by the team for processing the radar data. Therefore, various resources from commercial sector can be more helpful than expected.

3. Trade studies improve productivity and efficiency

The team spent plenty of time on designing and prototyping the weather-proof camera housing at the beginning of this semester. Due to the unexpected difficulties, the team was still not able to finish the work on time. The significant amount of time devoted into this tasks also indirectly hampered progress for other more crucial tasks of this project. Therefore, the team decided to abandon the original mounting solution and brainstormed our current mounting solution through careful trade studies. The current mounting solution was implemented and tested only within two days and proved to be able to provide the same level of reliability as the original solution, as well as the additional weather-proof feature. If the trade study could be done more early on during the semester, the team might be able to switch to the new plan earlier and would waste less time getting stuck on the original plan since the beginning.

4. Well-planned schedule is the key to project success

At the beginning of this semester, the team did not have a detailed schedule to guide and push the team to work. The team structure was not so organized so that the team was not able to make much progress. After the second System Engineering presentation, the team made a reasonable and detailed Work Breakdown Structure and an effective schedule. It turns out to be really helpful for facilitating the teamwork, and because of that the team was able to achieve all the basic goals set for this semester during the Fall Validation Experiments.

8.2 Key Spring activities

As addressed previously in the Project Management section (7.2 Schedule), the team will work on these six major parts to further improve our current work and eventually complete building our standalone perception system. They are listed in the chronological order as the following:

1. Parallel Computing
2. Radar Data Filtering
3. Sensor Synchronization
4. Reliable Sensor Fusion
5. Object tracking
6. 3D Reconstruction

9. Reference

[1] “Delphi Electronically Scanning Radar”, Delphi Automotive LLP. Web. 30 Sept. 2016, <http://www.delphi.com/docs/default-source/old-delphi-files/b87cda8b-468d-4f8e-a7a8836c370fc2c2-pdf>

[2] “Technical Characteristics for Electronically Scanning Radar 2.5”, Delphi Automotive LLP. Web. 30 Sept. 2016, <http://www.delphi.com/docs/default-source/old-delphi-files/b87cda8b-468d-4f8e-a7a8836c370fc2c2-pdf>

[3] “Delphi ESR Startup Guide” v2.0, Delphi Automotive LLP. 27 Apr. 2015.

[4] S. Ren; K. He; R. Girshick; J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.PP, no.99, pp.1-1

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. E. Reed. SSD: single shot multibox detector. CoRR, abs/1512.02325, 2015.