A Dynamic Collision Avoidance System

ECE4007 Senior Design Project

Section L03 Team Road Rage
Project Advisor: Dr. Erick Maxwell

Graham Fletcher
George Yuchi
Ryan Baker
Garrett Chinsomboon

Submitted

December 15, 2011

Contents

Executive Summary

To reduce the number of motor vehicle collisions, Team Road Rage designed and built a system that detects and alerts drivers of impending collisions. 26.5% of all collisions are front-to-rear, and 64% of those are due to driver inattention. The team's design reduces rear-end collisions by ensuring the driver is alerted of potential collisions ahead of time. The system searches for conditions likely to result in a rear-end collision with a leading vehicle. It takes into consideration road conditions and driver reaction time, and then warns the driver with visual and audio alerts before a collision becomes inevitable. The design includes a laser rangefinder that allows the system to detect objects up to 100 yards away, and its output is relayed to an Arduino microcontroller. The Arduino also interfaces with temperature, humidity, and rain sensors, and reports all sensor data to a computer, which is responsible for estimating braking distances. The system processes input from a USB camera using computer vision algorithms to detect vehicles and better understand the rangefinder data. The reaction time of each driver is estimated by determining the difference in time between the deceleration of the leading vehicle and the following vehicle, and is taken into consideration when determining when to issue an alert. The system issues alerts via a mobile device running Android 2.3, which is connected to the vehicle's stereo system. The mobile device displays visual alerts on its screen and plays audio alerts through the vehicle's speakers. This system differentiates itself from existing systems by taking into account road conditions and the driver's reaction time. Since each car model brakes and handles differently, the proposed design is intended to be implemented during vehicle production, taking into account the vehicle's specific performance specifications. The final fully-functional prototype costs approximately $1800.

1 Introduction

The Road Rage team designed a real-time collision detection system that alerts drivers of potential collisions to reduce traffic fatalities and delays. The team requested $1800 to develop and demonstrate a working prototype of the concept.

1.1 Objective

Team Road Rage designed and built a system comprised of sensors and software processing that detects and alerts drivers of imminent vehicular collisions. The system searches for conditions likely to result in a rear-end collision with a leading vehicle. The design determines the gap between the driver and the nearest leading car using a laser rangefinder and computer vision. After processing changes in distance between the two vehicles, the system detects current road conditions to accurately calculate the minimum braking distance. If a collision is imminent, the system then alerts the driver visually and aurally using an Android mobile device connected to the vehicle's speaker system. Since driving and braking characteristics vary from car to car, the system is intended to be sold to auto manufacturers via original equipment manufacturers (OEMs). The manufacturers can then integrate the design into new cars at the time of production at the factory.

1.2 Motivation

According to the National Safety Council, 26.5 percent of all vehicle crashes are front-to-rear-end. Although these crashes are responsible for only 4.3 percent of traffic-related fatalities, they are responsible for one-third of all crash-caused delays, resulting in 157 million vehicle-hours of delay annually. According to police reports, 64 percent of these rear-end collisions were the result of driver inattention, 14 percent were a result of following too closely, 3 percent were due to poor visibility and 2 percent were due to poor judgment [10]. Our project seeks to greatly reduce the number of preventable rear-end collisions by implementing a system that alerts the driver of an impending collision, providing just the amount of time required for the driver to react.

While collision avoidance has been an active area of research for several decades, most applications have been related to its use in autonomous (driverless) vehicles [11]. Currently available technologies, such as the driver assistance packages outlined in Table 9, offer lane departure warnings and attempt to estimate when a driver might be distracted, but are unable to adapt to the driving habits of different users under various circumstances and weather conditions [2] [4] [6] [9]. Our system could easily be incorporated into future driver assistance packages while minimizing costs through the use of a one-dimensional laser rangefinder and a low-resolution camera. This differs from other systems which use expensive, scanning laser rangefinders or multiple cameras for depth perception [11] [12] [13] [14].

1.3 Background

Collision avoidance systems is an area of active research among auto manufacturers, and currently several solutions are available to consumers in high-end luxury vehicles, such as those offered by Honda, Mercedes-Benz, Toyota and Audi (see Section 7.1). These systems are still in their infancy, and most of the progress in collision avoidance has been confined to closed-course testing of a very wide range of solutions, most of which utilize technologies such as radar [15] [16] [17] [18], LIDAR (a type of radar which uses light) [19] [20], computer vision (extracting information from a camera's video output) [21] [22] and various combinations of the three [15] [23] [24] [25] [26] [27] [28] [29] [30] [31].

An additional area of research that our project utilizes is that of traffic theory, which attempts to mathematically describe traffic flow. For over sixty years, various models have been proposed for modeling traffic flow. The first to gain significant attention took into account the differences in velocity between two vehicles [32]. More models were then proposed that took into account speed difference with a lead car [33], headway distance [34] and an "optimal" velocity [35]. One of the most recent models, the Comprehensive Optimal Velocity Model (COVM), incorporates both the headway and velocity differences between vehicles [36]. We have developed our own model that is more suitable for our application; it includes all the variables present in the COVM as well as additional empirically determined parameters about the driver's sensitivity to changes in headway and velocity.

The use of computer vision for detecting and tracking lanes and vehicles is an active area of research, particularly because of its applications in driver awareness systems and autonomous (driver-less) vehicles [37]. Methods for detecting lane markers include various thresholding techniques [38] [39] and the use of line-detection algorithms (i.e., the Hough transform) [40]. We employed a unique algorithm designed specifically for this project, which examines recent frames to get an "average" image of the road. An adaptive thresholding technique, along with RANSAC curve fitting, is then used to extract the locations of the lane markers. For detecting vehicles on the road, many have utilized distance-measurement techniques, such as radar or LIDAR, with reasonable success [41]. Others have resorted to the use of computer vision due to its relatively low price. Several methods have been proposed, such as searching for symmetrical objects [42] [43] or searching for various features, such as tail-lights [44]. Our system searches for certain Harr-like features determined *a priori* from various training data. This method is similar to the one proposed in the IEEE Journal [45].

Methods for alerting the user of impending collisions have also attracted quite a bit of interest. While several have proposed visual or tactile feedback methods [46], many make use of audio cues to warn the driver of impending collisions [47] [48]. These audio queues are quite diverse, and range from abstract tones to "audio icons," such as the sound of screeching brakes [49]. We are utilizing an unobtrusive, but noticeable, tone so as not to annoy or further distract the driver.

2 Project Description and Goals

The goal of Team Road Rage is to create a collision avoidance system that warns the driver of impending collisions well before they happen based on data gathered about a lead vehicle.

Team Road Rage's collision avoidance system consists of the following components: an Android device, a laser rangefinder, a laptop computer, a web camera, an Arduino microcontroller, a rain tracker and a temperature/humidity sensor. The Android device acts as the warning system and information display for the user. The laser rangefinder measures the distance from the lead vehicle. The laptop computer is responsible for computer vision algorithms that process images received by the webcam and is supplementary to the function of the laser rangefinder; the computer also relays information and warnings to the Android device. Rain and temperature sensors gather environmental data to get an idea about weather conditions outside the vehicle. The Arduino microcontroller collects data from the weather sensors and relays that data to the computer.

The features of this system are

- Predictions of impending collisions
- Warnings that leave the driver with sufficient time to avoid collisions
- Information about time headway from the lead vehicle on display for the driver
- Weather sensors that introduce accountability for inclement weather conditions
- Weatherproof enclosures to protect external parts
- Diagnostic Graphical User Interface (GUI) for detailed sensor data and calibration

3 Technical Specifications

The project makes use of several key hardware components. In particular, a laser rangefinder, a USB camera, an Arduino Mega ADK, a rain sensor, a temperature sensor, an Android device, a DC-DC converter and a laptop computer are used. The following is an overview of the technical specifications for each of these components.

3.1 Laser rangefinder

Table 1 contains the specifications for the laser rangefinder. The data rate of 10Hz (calibrated) ensures that the distance measurements are constantly up to date. This rate is crucial for especially for high speeds where there may be large discrepancies between each sampling. The off-the-shelve product for the Opti-Logic RS100 laser rangefinder satisfies these specifications however, the data collection rate is only at 200Hz. After further communication with the engineer at Opti-Logic, the team is able obtain one at 400Hz raw counts. The engineer at Opti-Logic tested the device and was certain that the laser rangefinder remains eye-safe at this rate. Hence, the Opti-Logic RS100 was selected to be the primary ranging device for our project.

Table 1. Laser Rangefinder Specifications [50]

Component	**Specification**
Connectivity	RS232
Data Rate	~10 Hz for calibrated operation ~400 Hz raw counts for uncalibrated operation
Accuracy	+/- 1 m on 1x1 m2 diffuse target with 50% (+/-20%) reflectivity, and up to 300 yards
Dimensions	32 x 78 x 84 mm

Power Supply	7-9 V

This component is mounted externally and is exposed to all weather conditions, therefore, the laser rangefinder was completely water-proofed. An aluminum enclosure was built to protect the device from rain, in addition, caulk and silicon was applied to the box to ensure that the device was sealed in and that water cannot seep through. The team ran a cable from the internal frame of the vehicle, where the laptop computer is stationed, to the hood of the vehicle in order to connect to the ranging device. Wire connecters were made to ensure that the laser rangefinder can be disconnected and reattached at a later time.

The rangefinder initially took raw measurements. After calibrating with the physical distance in meters, a script was created to convert the raw readings into distance measurements in meters where they could be used for processing later on. Once the power and data lines were connected, the laser ranger was aligned to ensure that the device was aimed straight ahead and not down into the ground or up into the sky. This process was done with the assistance of the webcam. Through the webcam, the team drew a line directly in front of the vehicle on the computer screen. A reflective object was placed far away along that line at the desired height and the laser was considered to be aligned when readings were obtained reflecting off the object's surface. After this process was completed, the laser rangefinder was mounted on the vehicle.

The laser rangefinder has trouble detecting dark surfaces of certain vehicles and will return an invalid reading when it is unable to detect anything within its range. In order to mitigate these unreliable readings, a controlling thread was written to ignore invalid results and wait until a valid response is recorded. During this period, the Kalman filter predicts the position of the lead vehicle based on previously acquired results. The input voltage to the laser rangefinder is 9 V and is obtained through the DC-DC converter and the step-down power MOSFET.

3.2 USB Camera

The USB camera is used for vehicle detection. Its dimensions shown in **Table 2** allow for it to be mounted up behind the windshield on the inside of the vehicle. The sensor resolution provides adequate clarity of vehicles ahead of it and compensates for closer vehicles that extend behind the minimum distance of detection of the laser rangefinder. The Logitech C-600 webcam satisfied these specifications and was chosen for our design project.

Table 2. USB Camera Specifications [51]

Feature	Specification
Connectivity	USB
Dimensions	7” x 5.5” x 7.5”
Sensor Resolution	2 MP

3.3 Arduino ADK

The connectivity specifications shown in **Table 3** for the Arduino ADK allow for communication with a computer, an Android device and the environmental sensors of the system. The Arduino is the communication hub of the entire system; it relays sensors information to be processed and processed information such as time headway or inclement weather conditions that may affect driver reaction time to the user.

Table 3. Arduino ADK Specifications [52]

Feature	Specification
Dimensions	4” x 2.1”

Power Supply	7-12V
Clock Speed	16 MHz
Connectivity	USB, Serial

The arduino board itself consists of an embedded, programmable FPGA microcontroller which receives updated runtime code compiled on a PC/Mac and sent to the arduino via a USB cable. The particular model we chose, the Arduino ADK, is the product of an initiative from both Google and the Arduino foundation to produce a microcontroller that could quickly and easily interface with an Android smartphone. Given the complexity of delivering alert information via the incredibly broad USB standard specification, this key ingredient saved the team months of development time.

Using the arduino, the system can receive updates from serial and analog devices. With serial communication baked in, and with a built-in 10-bit A to D converter in the arduino hardware, the arduino platform was the obvious choice for our system's needs. The arduino operates on a modified version of C++ commonly referred to as Wiring. We used the native development environment maintained by the Arduino community to prototype and eventually implement the hardware-level code needed to interface with the various sensors in our system and quickly and easily report them to the laptop's control thread to inform system-wide logic decisions such as when to issue alerts to the user. Further, issuing alerts becomes a painless process, as just one API call allows the device to pass information directly to the Android device via USB to a properly configured receiving application on the phone.

For our particular project implementation, the arduino was mounted inside the car dash and was the front end for the various sensor data inputs. It was powered directly via the 5V rail on a USB hub that is also installed inside the car dash. The connection through the USB hub allows for the device to communicate with the laptop computer while keeping wiring concerns to a minimum and simplifying connectivity with the laptop and the control thread running on it.

To implement the sensors in hardware, an overarching architecture was required. The team developed a continuously running thread which records the data sent to it from the various sensors asynchronously, then packages the information and transmits it to the control thread. Doing this required building out information interpretation for the rain, temperature, and humidity sensors. The rain sensor operated by varying its output between zero and five volts, an input easily interpreted by the Arduino’s A/D converter. The A/D detects the information from the “analog” pin subset on the device and maps the value to a 10-bit signed integer with a value from 0 to 1054.

The temperature/humidity sensor, the Sensirion SHT_15 combination breakout board, required a significantly more in-depth communication architecture. The device operates on a modified version of the I2C protocol, which is a variation on strictly serial communication. To implement the necessary procedures, an in-depth library was incorporated as a header file and contained all the necessary clocking information and register-setting information needed to establish communication. From there, looking at the data sheet for the product, the necessary calibration constants were determined and incorporated into the code. These are necessary to shift the incoming data into human-intelligible numbering and dimension, such as the Celsius or Fahrenheit scale.

3.4 Rain Sensor

The rain sensor, attached to the windshield inside the vehicle, uses its infrared beams mentioned in **Table 4** to detect the presence of rain. The model selected was the RT-50A Rain Tracker manufactured by Hydreon.

Table 4. Rain Sensor Specifications [53]

Feature	Specification
Sensor Operation	Invisible infrared beams (880 nm)
Window transmittance requirement	12% - 100% at 880 nm
Dimensions	4.45” x 2.45” x 1.15”
Power Supply	10 - 16V

The Rain tracker is another off-the-shelf product which the team integrated into our project. The product itself consists of two main components, the rain sensor and the relay circuit. The rain sensor is responsible for detection of dark external conditions, light rainfall and heavy rainfall. The output voltage from the rain sensor activates the relays in the circuit and in turn, instructs the vehicle to turn on the head lights and the windshield wiper. For this project, we did not set up the entire system in the vehicle since we only require the functionality of the rain sensor. By testing the relay circuit through input voltage simulation, the team determined that the voltage threshold for the detection of rainfall was at 2.5 V. Once this threshold was found, the rain sensor output line was connected to the input pin of the Arduino ADK. A code was written to detect the voltage level from the rain sensor, and if this voltage exceeded the predetermined threshold then rain was detected. The controlling laptop would obtain this information from the Arduino ADK and factor the result into the final alert. The 12 V input voltage to the sensor is regulated by the DC-DC converter.

Since the sensor functions by emitting infrared light and detecting the reflected light, the rain sensor was mounted on the windshield inside the vehicle. If water is present on the windshield, it will spread the infrared light outwards and a small amount of light will be reflected back to the sensor. On the other hand, most of the light will be reflected back off the windshield if it is not raining. Through this method, the sensor is able to detect whether it is raining and if so, whether it is heavy or light rain.

3.5 Temperature Sensor

The temperature sensor, as specified in **Table 5**, is used to determine if the environmental conditions are cold enough to warrant precautions against icy roads.

Table 5. Temperature Sensor Specifications [54]

Feature	Specification
Operational Temperature	-67 to 257 °F
Accuracy	+-0.9 °F (from 14°F to 185°F)
Power Supply	3 to 5.5 V (through data line)

The Sensirion SHT_15 weather sensor is a combination temperature/humidity sensor on a convenient breakout board designed for rapid prototyping. The design calls for the implementation of a variation of a popular serial communication protocol known as I^2C. The protocol allows for error-controlled send-receive commands with built-in conversion of sensor values to byte-formatted data. Additionally, both the temperature and humidity sensor readings are accessible for a given reading using just one set of wires and a slightly different register setting in software.

The sensors require anywhere from 3 to 5 volts DC, which was initially a challenge to produce within the confines of an automobile. Given that the DC-DC converter is being used to filter out alternator/engine transients and produce clean, steady 12 volts to the rain sensor and to the laser range finder's power mosfet, the team had to look elsewhere for power. Given the presence of the USB hub, however, the team realized that the arduino itself would receive 5V indirectly from the laptop, and would go through all the filtering within the usb hub and the Arduino itself. This made the Arduino's 5V regulated DC output a prime candidate for powering the SHT_15 sensor chip.

The sensor chip's pins on the breakout board connects to the insulated communications cable which was snaked into the engine compartment of the car via a boot in the firewall in front of the front passenger seat. This routing position was ideal for insulation for the elements, and allowed for the communications cable to easily reach the temp/humidity sensor. The sensor required a power (+5) pin, as well as access to the system's main ground bus. In addition, the modified version of the I^2C serial protocol implemented by the Sensirion chip requires both a clock and a data pin which are tied to the main communication cable. Initially, the temperature sensor was giving high temperature readings in both Celsius and Fahrenheit. Additionally, the humidity values we were seeing were slightly off. Given that the chip uses the temperature findings to calculate the humidity reading, the logical answer was to examine the temperature reading first. Upon closer inspection, it was determined that the initial placeent of the temperature sensor in the engine compartment was too close to the engine block and was overheating due to radiation from the hot exterior of the engine itself. After relocating the sensor, both readings returned to expected values. As with resultant sensor data from the other sensors such as the rain sensor, both temperature and humidity data were recorded by the Arduino and packaged in a single byte array before being sent out via direct serial communication to the control thread running on the laptop for further processing. Before processing, the values are adjusted using the fixed constants found in the part's data sheet. Afterwards, the values are converted to unsigned bytes for simpler digestion in the software running on the control thread.

3.6 Android Device

Auditory warnings is produced by a mobile phone running the Android Operating System, which is connected through the headphone jack mentioned in **Table 6** to the automobile's auxiliary port allowing warnings to be played over the car speakers.

Table 6. Android Device Specifications [55]

Feature	Specification
Audio output	3.5 mm headphone jack
Screen Size	3.7 inches
Dimensions	4.56 inches x 0.54 inches x 2.36 inches

The Android device's main purpose is to deliver processed information about the driver's driving conditions to the driver through an easy to interpret GUI which receives its information from the Arduino ADK board development. The architecture for receiving data from the Arduino ADK was based heavily off of the USBAccessoryKit Demo code and the USB Accessory APIs that were released by Google.

3.7 DC-DC Converter

The DC-DC converter allows for conversion of DC voltage from 6-34 V to 5-24 V as shown in **Table 7.** The USB connectivity is such that the DC voltage conversion rate is programmable to fit the requirements of the device connected.

Table 7. DC-DC Converter Specifications [5]

Feature	Specification
Connectivity	USB
Dimensions	135mm x 37mm
Input Voltage Range	6-34 V

Output Range	5-24 V

This component is essential to regulate the voltage coming into the system from the car battery. The DC-DC converter takes the input voltage of 12 V from the car battery and outputs a modulated 12 V to the system. This output is wired directly to the rain sensor within the car to power the device. The modulated 12 V also goes through the step-down power MOSFET which outputs 9 V to power the laser rangefinder in front of the vehicle. The DC-DC converter is utilized to eliminate the voltage fluctuations from the car battery and prevent the hardware components from being potentially damaged.

3.8 Laptop Computer and Laptop GUI

The computer hosts the GUI that displays data on the computer screen, mentioned in **Table 8**, such as sensor information or real time video stream from the USB camera. The GUI also allows for the calibration of the laser rangefinder and the camera.

Table 8. Computer and its GUI Specifications [56]

Feature	Specification
Video display	Real-time camera video, image processing and distance overlays
Interface	Access to modify calibration and operating parameters
Computer Display	15.6 inches
Connectivity	USB

The laptop computer is responsible for calculating the time headway of the vehicle in real time as well as operating the controlling thread which governs all other hardware components. Since the sensors are updating at different rates, we needed a governing code to regulate the data. The thread handles asynchronous updates and calculates the current time headway based on those input data. The range information is provided directly to the laptop computer. The rain and temperature/humidity information is provided to the laptop through the Arduino ADK. After time headway calculations are made, the alert is forwarded to the Android device through the Arduino ADK.

ABIT MORE ABOUT THE CONTROLLING THREAD HERE – Graham Fletcher

4 Design Approach and Details

Our design approach utilizes a traffic-flow model which is used to predict information about a driver's reaction time. This information, along with estimates of the coefficient of friction between the road and the tires, is taken into account to estimate a total stopping distance. The total stopping distance, along with data, such as a lead vehicle's speed and deceleration, is used to determine whether or not the driver is following too closely for the given speed and conditions. A custom computer vision algorithm is utilized to detect a lead vehicle in the current lane.

4.1 General Design Approach

Detecting Potential Collisions

The primary function of our system is to determine the risk of a collision, and once that risk exceeds a certain threshold, notify the user through an audible alert. In order to define the level of risk at any given moment, we are utilizing an extended version of the Comprehensive Optimal Velocity Model (COVM), proposed by Tian et al. [36]. The model attempts to make predictions about traffic flow based on the following formula:

$$\frac{dv_n}{dt}(t) = \kappa[V_1(\Delta x_n(t)) - v_n(t)] + \lambda V_2(\Delta v_n(t))$$

At any given time, t, the speed of the following vehicle is given by $v_n(t)$. The headway between a lead vehicle and a following vehicle is given by $\Delta x_n(t)$ and the difference in the velocities of the two vehicles is given by $\Delta v_n(t)$. κ and λ were treated as constants in [36], but we have dynamically determined these values, as outlined below. Unlike other models, such as that proposed by [35], the COVM has been shown to remain stable under a wide range of conditions, including scenarios in which sudden braking or acceleration may occur. This stability is critical, as we require that our system be able to react to situations in which a lead vehicle brakes suddenly. With this model, we can predict how a driver will react to changes in a lead vehicle's headway or velocity, and if we determine that a collision will occur without immediate action being taken, an alert will be issued.

The value for $v_n(t)$ is read from the OBD-ii port in the vehicle through the OBD scanner. The value of $\Delta x_n(t)$ is the distance reading reported by the rangefinder, and the second derivative of this value is $\Delta v_n(t)$. The determination of values for κ and λ is unique to each driving session, and is empirically determined by finding the least-squares solution to the above equation, given the actual values measured for $\frac{dv_n}{dt}(t)$. Knowing these values allow us to acquire constants that relate to relative reaction time and sensitivity to changes in depth, which enables the system to adapt to situations in which the driver may be impaired due to a variety of circumstances, such as sleep deprivation or poor visibility.

In order to estimate the time it takes for a collision to occur, we used the following equation, given Δx for t_0 and its preceding two samples:

$$\Delta x(t) = \Delta x(t_0) + [\frac{\Delta x(t_0) - \Delta x(t_0 - 1)}{dt}]t + \frac{1}{2}[\frac{\Delta x(t_0) - 2\Delta x(t_0 - 1) + \Delta x(t_0 - 2)}{dt^2}]t^2$$

To determine the stopping distance for a given speed, we considered the coefficient of friction between the tires and the road. We used information from the rain and temperature sensors to estimate this value.

While there is little correlation between the atmospheric temperature and the friction coefficient, we used the temperature sensor to choose one of two formulas. If the temperature is freezing or below, we will use the following approximation from:

$$\mu = -0.1409\ln(m) + 0.6318$$

where m denotes the percent of moisture in the atmosphere present on the road [57]. This value is roughly estimated based upon measured precipitation. For conditions above freezing, we are using experimentally determined values acquired under controlled conditions by [58]. With this coefficient of friction, we used the following formula to determine the stopping distance, L_b:

$$L_b = \frac{v^2}{2\mu g}$$

where v is the vehicle's velocity and g is the acceleration due to gravity. This roughly corresponds to the distance the vehicle would travel without the braking torque overcoming the force of static friction, which is the case for a vehicle equipped with anti-lock brakes (ABS). Once the values for κ and λ is determined, the driver's reaction time is estimated and incorporated into the previous equation for the time of collision. At that instant, an audible warning is issued from the Android device through the car's speakers.

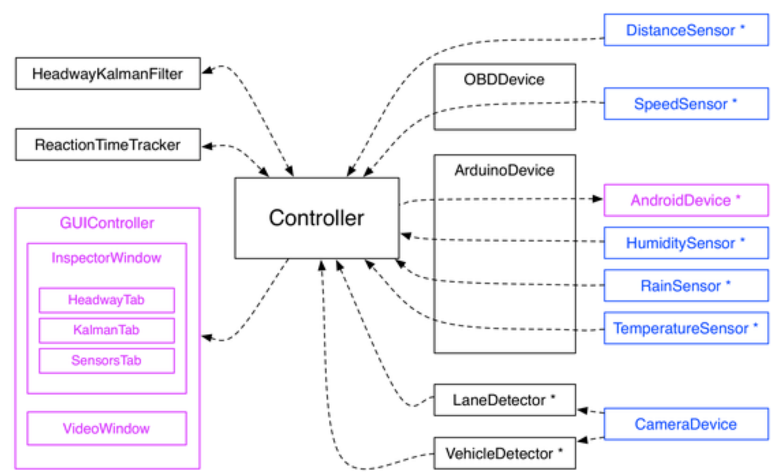
Detecting the Presence of a Lead Vehicle

While the aforementioned model is sufficient for determining the potential for collisions and warning the driver in time, it makes the assumption that there is a lead vehicle present, directly ahead of the following vehicle. Our system is robust enough to be able to respond to situations in which there is no lead vehicle, or a lead vehicle suddenly appears (from a lane change). Since the laser rangefinder reports distance readings to anything ahead of it, not just vehicles, and the system determines whether or not the data is valid. This is accomplished using computer vision. First, the current lane is detected, using a method similar to that proposed by [45]. Next, vehicles are detected by searching for Haar-like features that match characteristics found in our training data. When a vehicle is detected, its location is followed for several frames, and if it is determined to be in the same lane as the following vehicle, it is classified as the lead vehicle and the aforementioned method for predicting collisions is employed.

4.2 Software Design

The following Figure 1 provides a very simplified overview of the major software components of our system. The pink components are system outputs, and the blue components are system inputs.

Figure 1. Software components overview



The goal was to organize the software in a way that would minimize any blocking operations and allow each sensor to update at its maximum rate.

Part of the solution was to divide up the operations among different threads. Each object with an asterisk has its own runloop which lives in a thread independent of the others. When a result becomes available, it is passed only to the objects which are interested in it using Qt's signals and slots mechanism.

Several classes were created specifically to safeguard data in the multi-threaded environment. For example, the purpose of the ArduinoDevice is to make sure that only one object on one thread is allowed to communicate with the Arduino at any given instant. Without this protection, objects on different threads could try to communicate with the Arduino at the same time, which would fail.

4.3 Kalman Filter

In order to accurately track our system's critical measurements, the distance to the lead vehicle and the following vehicle's velocity, our team designed and implemented a Kalman filter.

First, the Kalman Filter provides more reliable measurements with time, as it promotes sensor readings with the lowest covariance over more noisy readings. Second, the Kalman Filter provides a mechanism through which we can predict future values of sensors given past values. This is critical in situations in which a lead vehicle is present, as confirmed through computer vision, but not aligned with the laser rangefinder. An example of such a scenario would be when the driver is driving along a curve in the road.

The states we chose to track, shown in Figure 2, are the range to the lead vehicle and the velocity of the following vehicle. Two additional states were added to specifically track changes in the first two states, allowing us to also perform derivations of the values.

Figure 2. Update Equations and State Matrices

Time update equations:

$$\hat{x}_k^- = A\hat{x}_{k-1}^-$$
$$P_k^- = AP_{k-1}A^{\mathrm{T}} + Q$$

Measurement update equations:

$$K_k = P_k^- H^{\mathrm{T}}(HP_k^- H^{\mathrm{T}} + R)^{-1}$$
$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$
$$P_k = (I - K_k H)P_k^-$$

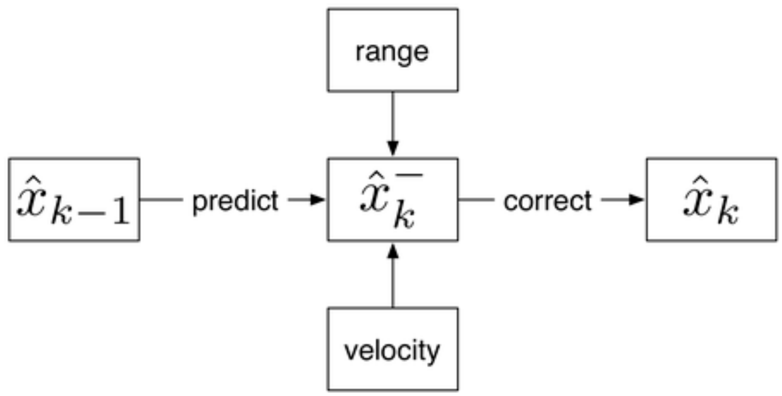
States: $x = \begin{vmatrix} \text{range}_k \\ \text{range}_{k-1} \\ \text{velocity}_k \\ \text{velocity}_{k-1} \end{vmatrix}$ Transition Matrix: $A = \begin{vmatrix} 2 & -1 & -dt & dt \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & 1 & 0 \end{vmatrix}$

With these four states, we are able to directly calculate the time headway, the distance headway, the lead vehicle velocity, the following vehicle velocity, and the following vehicle acceleration.

General Kalman Filter

A general Kalman filter would be implemented as shown in Figure 3. Given the last state of the system, the next value is predicted using the transition matrix specified previously. After the new state has been estimated, it is corrected given incoming sensor readings, and the resulting state is a weighted sum of the prediction and correction.

Figure 3. General Kalman Filter Update Procedure

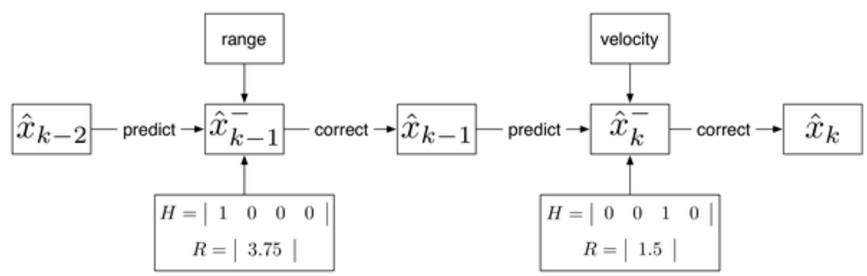


However, there is an underlying assumption that all sensor readings are available at the same time for the correction step. Given that each sensor in our system operates independently and asynchronously, the general Kalman filter isn't sufficient for our application, especially given the scenario in which the velocity of the following vehicle is readily available to update the Kalman filter, but the distance to the lead vehicle is not known, perhaps when driving around a curve.

Asynchronous Kalman Filter

Our solution to the limitations of the General Kalman Filter is to modify the parameters of the Kalman filter in real time, based on the available readings.
Figure 4 shows a possible scenario in which the Kalman filter is updated with an available reading from the rangefinder, and then updated again when the speed reading becomes available.

Figure 4. Asynchronous Kalman Filter Update Procedure



In order to make this actually work, we have to modify the matrix relating the sensor readings to the state space and the variance of the sensor providing the update. As long as these two parameters are changed to the correct values before the update procedure, the Kalman filter functions as if it is receiving synchronous updates.

4.4 Codes and Standards

1. RS232 or Serial Port is a 25-pin serial connector which communicates data from the laser rangefinder to the rest of the system. It features
   - Highly customizable pin conventions
   - 8-bit data transmissions [59]
2. USB (Universal Serial Bus) is a ubiquitous 4-pin serialized connector and handshake standard for communication between the webcam, Arduino, and computer. It features
   - Self-powering connector
   - Vast interoperability, library interface to Arduino
   - 480 Mbit/sec transmission rate [60]

3. I2C (Inter-Integrated Circuit or two-wire interface) interfaces the temperature/humidity sensor to the Arduino and is a popular IC-level interconnect between micro-controllers and device drivers. It features
    - 400 Kbits/sec data transmission
    - Configurable master/slave instances
    - Supported by the Arduino "Wire" library for rapid prototyping. [61] [62]
4. The Class 1 (eye-safe) laser standard applies to the laser emission cavity in the laser rangefinder. It specifies
    - A laser wavelength of 905 nm +/- 10nm
    - Low enough power output to prevent harmful exposure [63]
5. *Manual on Uniform Traffic Control Devices*, published by the U.S. Department of Transportation, outlines:
    - Normal pavement markings are "4 to 6 inches wide"
    - "Broken lines should consist of 10-foot line segments and 30-foot gaps"
    - Minimum highway lane width of 12 feet [64]
6. ISO-9141-2 is an OBD-ii protocol used by Chrysler and many European and Asian vehicles. Communication occurs on a single, bidirectional line without handshake signals.
    - Operates at 10.4 Kbaud
    - Message length is restricted to 12 bytes, including CRC [65]

4.5 Constraints, Alternatives and Tradeoffs

Alternatives

An alternative to sensors for detecting weather conditions is to use the android device to receive weather forecast information from a service like the National Weather Service (NWS). However, weather information from such services is often reported with delay whereas the sensors offer instantaneous measurements.

Alternatives to the laser rangefinder would be radar or Light Detection and Ranging (LIDAR). Radar distance finders are popular and cheap but only available in short distances; long distance radar is far more expensive than the laser rangefinder. LIDAR is a much better alternative to the laser rangefinder but is also ruled out due to not only its expensive price but also the tedious calculations that LIDAR technology would entail.

Constraints

One debilitating limitation of the Early Collision Avoidance System is the conditions necessary for adequate visibility. Both computer vision and the laser rangefinder cannot operate under low visibility conditions such as fog. The laser rangefinder is also limited by surfaces with poor reflectivity (dirty cars).

Other objects that may share the road with vehicles include cyclists and pedestrians. The narrow field of view of the laser makes it impossible to detect, not to mention measure, the distance from a small object like a pedestrian or cyclist.

Tradeoffs

LIDAR or radar technology would have been the ideal technology to work with instead of laser but as mentioned previously, the price and the calculations required to harness the potential of either technology would have likely significantly delayed progress on other aspects of the project.

Using a weather service in place of weather sensors to detect weather conditions does not only provide delayed information but also introduces the problem of reliability. The android device is reliant upon cellular data service which is in turn reliant on the availability of cellular coverage.

5 Schedule, Tasks and Milestones

Please refer to the Gantt chart in Appendix A for project schedule and milestones. Since it is particularly large, another copy of the chart will also be available on the team website. The initial plan was to have a working prototype by November 15, however, during to unexpected challenges along the process, the team is able to complete the product at a later date. The group finished the final product in time for the demonstration during the second week of December, however, such delays mean that the limited time remained for testing and troubleshooting the product. Apart from the final stages of the product, all team members were able to meet and develop the prototype as outlined in the Gantt chart from the proposal.

When the individual team members worked on the hardware components separately, the team is able to get our hardware communications operational as stated in the schedule. However, there was a delay installing the parts inside and on the vehicle. As a result, there was a delay in writing the governing software for the whole system. Regardless, the team is able to deliver the final product in time for the demonstration and presentation.

6 Project Demonstration

To demonstrate successful operation of our system, we installed all of the components in an actual vehicle and tested it on the highway and local roads. In order to prove that the system operates correctly, we implemented a series of tests in which another team member drove a lead vehicle ahead of the vehicle in which the system is installed. We then implemented a series of maneuvers, such as sudden braking and lane changes. Specifically, the lead vehicle created situations that involved sudden braking, gradual breaking and sudden lane changes, and our system issued alerts as necessary. We had the advisers come out with us to test the prototype and witness the system first-hand.

To test the prototype throughout the design process, we routinely drove the car in traffic and gathered data from the installed sensors. We recorded that data so that not only can we test the system in real time, but also simulate the trip later on if we make changes to the software. We used this prerecorded data to create a simulation, which was demonstrated before the rest of the class. There is also a brief video of the system in operation on a local highway to show that that prototype functions in real-time.

7 Marketing and Cost Analysis

7.1 Marketing Analysis

Marketing

Since the Warning System for Collision Avoidance requires the installation of many device components, it will be difficult for "teen" drivers to implement the system in their personal vehicles by themselves, especially if they lack the necessary expertise. Moreover, the cost of the overall system rose significantly with the inclusion of the laser rangefinder and laptop. For the above reasons, aftermarket installations consumer base is expected to be extremely narrow and such a product likely will not survive. In turn, the Warning System for Collision Avoidance's target audience is high-end automobile manufacturers such as BMW, Lexus, Audi, Mercedes-Benz and Bentley. The intention is to manufacture the vehicles with the system built-in before they are shipped to consumers.

Similar Products

Many new vehicles are now manufactured with collision avoidance systems pre-installed. Generally, the pre-crash systems available today are based on laser and radar range detection. The new system incorporates computer vision, in addition to the laser rangefinder device, into the prototype to detect other vehicles traveling in the same lane as the user. Cameras have certainly been in the automobile manufacturing industry for a while, but are rarely utilized in such manner. Rear-end camera applications are becoming increasingly common in modern automobiles, such as the Audi A8, to assist with back-up parking. Moreover, the Lexus LS 460L [66] is shipped with the camera-based automatic park assist system. BMW 7 Series provides night vision camera to assist with night-time driving and in dark environments [67]. Unlike our system, these do not take weather and environmental conditions into account in the alert process. Our design is also unique since it implements a customized algorithm to weigh each environmental factor in real-time before the system outputs an alert. For example, the system notifies the driver sooner when experiencing rainy weather conditions. **Table 9** lists some manufacturers which implements a similar system to the Warning System for Collision Avoidance in their vehicles.

Table 9. Manufacturers of Collision Alerts Systems and Their Features

Manufacturer	**Honda [2]**	**Mercedes-Benz [4]**	**Toyota [6]**	**Audi [9]**
Model	Inspire, Acura	S-Class, E-Class	Lexus LS	Audi A8

7.2 Cost Analysis

Development Costs

Engineers with some expertise are expected to be responsible for manufacturing and installing the Warning System for Collision Avoidance on high-end vehicles as they are coming off the assembly line. An estimate for an engineer's wage is at $30 per hour. **Table 10** displays the breakdown of the engineers' labor hours for the course of the project.

Table 10. Final Engineer Costs

Tasks	Engineering Hours	Costs
Integrate Hardware Components	120	$3,600
Make Laptop Preparations	15	$450

Final Product Costs

The cost of the system is heavily based on the cost of the hardware components and product assembly while also taking into consideration the engineers' costs. **Table 11** displays the component costs for producing the prototype.

Table 11. Final Cost of Prototype Components [1] [3] [5] [7] [8]

Prototype Parts	**Unit Cost**
Arduino Mega ADK	$85
Android Mobile Phone	$200
Laptop	$860
Webcam	$50
OBD II Scanner	$50
DC Power Converter	$60
Laser Rangefinder	$320
Rain Sensor	$90
Enclosure	$30
Phone Mount	$15
Wires, Cables and Connectors	$14
USB Hub	$24
Waterproofing Materials	$10
TOTAL	**$1,808**

The final cost for the project prototype is $1808, approximately $40 higher than initially projected in the proposal. Out of this amount, only $495 was taken from the senior design budget, the remaining amount was lent to the project by team members, bought by team members or obtained from external sources. The final amount spent on the project, including both the hardware components and the labor costs, is $13,208.

It is worth noting that the cost of the prototype is higher than that of the manufactured products in the plant since the components for the prototype were bought in single-quantities rather than in bulk. Also keep in mind that the labor cost for production could be lower; an engineer is not necessarily needed at the factory since the product development stage is over. As such, there is no need for fringe benefits during the production process.

Based on the final component costs of $1,808 for the prototype, the producers can market the product for $2,000 each and earn a profit of $192 per unit. Since high-end vehicles today are sold for $70,000 to $90,000, this additional cost should not pose as a significant financial burden for the consumers. Since the hardware components are ordered in bulk, a discount rate of approximately 5% can be expected. In turn, this lowers the component costs to $1,718 resulting in an even higher profit margin of $282 per unit for the car manufacturer. Such marketing techniques could quickly recover the initial engineer costs and reach a break-even point after selling 47 vehicles. The time required to reach this point depends heavily on the rate of the vehicles manufactured and sold to consumers.

8 Summary

The Early Collision Avoidance System designed by Team Road Rage alerts distracted drivers of impending collisions. The objective is to reduce the risk of rear-end accidents on the road since most of the accidents in America are rear-end collisions and are caused by “teen” drivers. In order to fulfill this objective, the team designed a system that consists of a laser rangefinder, a webcam, an Android device, an Arduino ADK, a rain sensor, a temperature/humidity sensor and a laptop computer. The ranging device, mounted at the front of the vehicle, obtains the distance to the leading vehicle for the system. The weather sensors, consisting of the temperature/humidity sensor and the rain tracker, provide information about the outside environment. The laptop computer, gathering the data through the Arduino ADK, calculates the current time headway to the leading vehicle and outputs an alert based on weather conditions if the time headway falls to a dangerously low level. An auditory alert, based also on the weather conditions, is then issued through to the Android device and the vehicle speakers. In order to make the system more user-friendly, the team implemented GUIs on both the laptop computer and the Android in order to display to current information obtained from the system. The computer vision component and the Kalman filter aid the system in predicting the positions of vehicles when the laser rangefinder is unable to track them. The team is able to complete the project before the demonstration and is able to test the prototype against the proposed objectives.

10 References

[1] Dell. (2011, September 20, 2011). *XPS 15z*. Available: http://www.dell.com/us/p/xps-15z/pd

[2] Honda. (2006, September 19, 2011). *Collision Mitigation Brake System* [Product Manual]. Available: https://techinfo.honda.com/rjanisis/pubs/om/JA0606/JA0606O00325A.pdf

[3] Scantool. (2011, September 21, 2011). *OBDLink SX Scan Too*. Available: http://www.scantool.net/obdlink-sx.html

[4] Mercedes-Benz. (2011, September 21, 2011). *Mercedes-Benz TecDay Special Feature: PRE-SAFE And PRE-SAFE Brake* [Website]. Available: http://www.emercedesbenz.com/Nov08/12_001507_Mercedes_Benz_TecDay_Special_Feature_PRE_SAFE_And_PRE_SAFE_Brake.html

[5] Minibox. (2011, September 21, 2011). *DCDC-USB, Intelligent DC-DC converter with USB interface* [Website]. Available: http://www.mini-box.com/DCDC-USB

[6] Lexus. (2011, September 21, 2011). *Advanced Pre-Collision System (APCS) with Driver Attention Monitor* [Website]. Available: http://www.lexus.com/models/LS/features/safety/advanced_precollision_system_apcs_with_driver_attention_monitor.html

[7] Opti-Logic. (2011, September 14, 2011). *Opti-Logic Pricing*.

[8] Hydreon. (2011, September 20, 2011). *RT-50A Rain Tracker*. Available: http://www.raintracker.com/product_details/rt-50a.php

[9] Autopressnews.com. (2010, September 20, 2011). *New 3rd generation Audi A8 deploys it technologies from its own onboard network to received satellite data* [Website]. Available: http://www.autopressnews.com/2010/03/audi/audi_a8_8.shtml

[10] T. Brown and J. Lee, "Human performance models and rear-end collision avoidance algorithms," *\ldots : The Journal of the Human \ldots,* 2001.

[11] R. Aufrère, "Perception for collision avoidance and autonomous driving," *Mechatronics,* vol. 13, pp. 1149-1161, December 2003.

[12] F. J. Jimenez, F. and J. E. Naranjo, "Improving the obstacle detection and identification algorithms of a laserscanner-based collision avoidance system," *Transportation Research Part C-Emerging Technologies,* vol. 19, pp. 658-672, Aug 2011.

[13] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection: a review," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 28, pp. 694-711, 2006.

[14] O. Inoue and S. Ahn, "Following vehicle detection using multiple cameras," *\ldots Electronics and Safety,* 2008.

[15] S. Tokoro, K. Moriizumi, T. Kawasaki, T. Nagao, K. Abe, and K. Fujita, "Sensor fusion system for pre-crash safety system," in *2004 IEEE Intelligent Vehicles Symposium*, ed. 345 E 47th St, New York, NY 10017 USA: IEEE, 2004, pp. 945-950.

[16] R. Dixit and L. Rafaelli, "Radar requirements and architecture trades for automotive applications," in *1997 IEEE MTT-S International Microwave Symposium Diges, Vols I-III: High Frequencies In High Places*, ed. 345 E 47th St, New York, NY 10017: IEEE, 1997, pp. 1253-1256.

[17] S. Le Beux, V. Gagne, E. M. Aboulhamid, P. Marquet, and J.-L. Dekeyser, "Hardware/Software exploration for an anti-collision radar system," in *IEEE MWSCAS'06: Proceedings of the 2006 49th Midwest Symposium on Circuits and Systems*, ed. 345 E 47th St, New York, NY 10017 USA: IEEE, 2006, pp. 385-389.

[18] R. Abou-Jaoude, "ACC radar sensor technology, test requirements, and test solutions," *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS,* vol. 4, pp. 115-122, SEP 2003.

[19] A. Ewald and V. Willhoeft, "Laser scanners for obstacle detection in automotive applications," in *PROCEEDINGS OF THE IEEE INTELLIGENT VEHICLES SYMPOSIUM 2000*, ed. 345 E 47TH ST, NEW YORK, NY 10017 USA: IEEE, 2000, pp. 682-687.

[20] A. Kirchner and C. Ameling, "Integrated obstacle and road tracking using a laser scanner," in *PROCEEDINGS OF THE IEEE INTELLIGENT VEHICLES SYMPOSIUM 2000*, ed. 345 E 47TH ST, NEW YORK, NY 10017 USA: IEEE, 2000, pp. 675-681.

[21] H. Elzein, S. Lakshmanan, and P. Watta, "A motion and shape-based pedestrian detection algorithm," in *IEEE IV2003: INTELLIGENT VEHICLES SYMPOSIUM, PROCEEDINGS*, ed. 345 E 47TH ST, NEW YORK, NY 10017 USA: IEEE, 2003, pp. 500-504.

[22] C. Caraffi, S. Cattani, and P. Grisleri, "Off-road path and obstacle detection using decision networks and stereo vision," *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS,* vol. 8, pp. 607-618, DEC 2007.

[23] D. Gavrila, M. Kunert, and U. Lages, "A multi-sensor approach for the protection of vulnerable traffic participants - the PROTECTOR project," in *IMTC/2001: PROCEEDINGS OF THE 18TH IEEE INSTRUMENTATION AND MEASUREMENT TECHNOLOGY CONFERENCE, VOLS 1-3: REDISCOVERING MEASUREMENT IN THE AGE OF INFORMATICS*, ed. 345 E 47TH ST, NEW YORK, NY 10017 USA: IEEE, 2001, pp. 2044-2048.

[24] T. Kato, Y. Ninomiya, and I. Masaki, "An obstacle detection method by fusion of radar and motion stereo," in *SICE 2003 ANNUAL CONFERENCE, VOLS 1-3*, ed. 345 E 47TH ST, NEW YORK, NY 10017 USA: IEEE, 2003, pp. 689-694.

[25] A. Gern, U. Franke, and P. Levi, "Advanced lane recognition - Fusing vision and radar," in *PROCEEDINGS OF THE IEEE INTELLIGENT VEHICLES SYMPOSIUM 2000*, ed. 345 E 47TH ST, NEW YORK, NY 10017 USA: IEEE, 2000, pp. 45-51.

[26] N. Shimomura, K. Fujimoto, T. Oki, and H. Muro, "An algorithm for distinguishing the types of objects on the road using laser radar and vision," *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS,* vol. 3, pp. 189-195, SEP 2002.

[27] R. Aufrere, J. Gowdy, C. Mertz, C. Thorpe, C. Wang, and T. Yata, "Perception for collision avoidance and autonomous driving," *MECHATRONICS,* vol. 13, pp. 1149-1161, DEC 2003.

[28] R. Labayrade, C. Royere, and D. Aubert, "A collision mitigation system using laser scanner and stereovision fusion and its assessment," in *2005 IEEE Intelligent Vehicles Symposium Proceedings*, ed. 345 E 47TH ST, NEW YORK, NY 10017 USA: IEEE, 2005, pp. 441-446.

[29] K. C. Fuerstenberg and B. Roessler, "Results of the EC-Project INTERSAFE," in *ADVANCED MICROSYSTEMS FOR AUTOMOTIVE APPLICATIONS 2008*, ed. HEIDELBERGER PLATZ 3, D-14197 BERLIN, GERMANY: SPRINGER-VERLAG BERLIN, 2008, pp. 91-102.

[30] N. Floudas, A. Polychronopoulos, O. Aycard, J. Burlet, and M. Ahrholdt, "High level sensor data fusion approaches for object recognition in road environment," in *2007 IEEE INTELLIGENT VEHICLES SYMPOSIUM, VOLS 1-3*, ed. 345 E 47TH ST, NEW YORK, NY 10017 USA: IEEE, 2007, pp. 269-274.

[31] K. Furstenberg, P. Baraud, G. Caporaletti, S. Citelli, Z. Eitan, U. Lages, and C. Lavergne, "Development of a pre-crash sensorial system - the CHAMELEON project," in *VEHICLE CONCEPTS FOR THE 2ND CENTURY OF AUTOMOTIVE TECHNOLOGY*

vol. 1653, ed. PO BOX 10 11 39, W-40002 DUSSELDORF, GERMANY: V D I-V D E - VERLAG GMBH, 2001, pp. 289-310.
[32] L. PIPES, "A MATHEMATICAL ANALYSIS OF TRAFFIC DYNAMICS," *JOURNAL OF THE OPERATIONS RESEARCH SOCIETY OF AMERICA,* vol. 1, p. 151, 1953.
[33] R. CHANDLER, R. HERMAN, and E. MONTROLL, "TRAFFIC DYNAMICS - STUDIES IN CAR FOLLOWING," *OPERATIONS RESEARCH,* vol. 6, pp. 165-184, 1958.
[34] G. NEWELL, "NONLINEAR EFFECTS IN THE DYNAMICS OF CAR FOLLOWING," *OPERATIONS RESEARCH,* vol. 9, pp. 209-229, 1961.
[35] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama, "Dynamical model of traffic congestion and numerical simulation," *Phys. Rev. E,* vol. 51, pp. 1035-1042, Feb 1995.
[36] T. Jun-Fang, J. Bin, and L. Xing-Gang, "A New Car Following Model: Comprehensive Optimal Velocity Model," *COMMUNICATIONS IN THEORETICAL PHYSICS,* vol. 55, pp. 1119-1126, JUN 2011.
[37] C. Su, "An Effective and Fast Lane Detection Algorithm," *Advances in Visual Computing,* 2008.
[38] M.-J. Jeng, C.-Y. Guo, B.-C. Shiau, L.-B. Chang, and P.-Y. Hsiao, "Lane detection system based on software and hardware codesign," in *Autonomous Robots and Agents, 2009. ICARA 2009. 4th International Conference on*, ed, 2009, pp. 319-323.
[39] Y. Wang, "Lane detection and tracking using B-Snake," *Image and Vision Computing,* vol. 22, pp. 269-280, April 2004.
[40] W. Wijesoma and K. Kodagoda, "Road edge and lane boundary detection using laser and vision," *\ldots and Systems,* 2001.
[41] M. Bertozzi, A. Broggi, M. Cellario, A. Fascioli, P. Lombardi, and M. Porta, "Artificial vision in road vehicles," *Proceedings of the IEEE,* vol. 90, pp. 1258-1271, July 2002.
[42] G. MAROLA, "Using symmetry for detectiong and locating objects in a picture," *Computer Vision Graphics and Image Processing,* vol. 46, pp. 179-195, MAY 1989.
[43] T. Zielke, M. Braukmann, and W. Vonseelen, "Intensity and edge-based symmetry detection with an application to car-following," *CVGIP-Image Understanding,* vol. 58, pp. 177-190, SEP 1993.
[44] L. HRL Laboratories, "Collision avoidance system utilizing machine vision taillight tracking," February 2000.
[45] H. Bai, J. Wu, and C. Liu, "Motion and haar-like features based vehicle detection," in *12th International Multi-Media Modelling Conference Proceedings*, ed. 345 E 47th ST, New York, NY 10017 USA: IEEE, 2006, pp. 356-359.
[46] C. Ho, N. Reed, and C. Spence, "Multisensory In-Car Warning Signals for Collision Avoidance," *Human Factors: The Journal of the Human Factors and Ergonomics Society,* vol. 49, pp. 1107-1114, December 2007.
[47] J. D. Lee, D. V. McGehee, T. L. Brown, and M. L. Reyes, "Collision Warning Timing, Driver Distraction, and Driver Response to Imminent Rear-End Collisions in a High-Fidelity Driving Simulator," *Human Factors: The Journal of the Human Factors and Ergonomics Society,* vol. 44, pp. 314-334, April 2002.
[48] C. L. Baldwin and J. F. May, "Loudness interacts with semantics in auditory warnings to impact rear-end collisions," *Transportation Research Part F: Psychology and Behaviour,* vol. 14, pp. 36-42, January 2011.
[49] R. Graham, "Use of auditory icons as emergency warnings: Evaluation within a vehicle collision avoidance application.," *Ergonomics,* 1999.
[50] "Opti-Logic RS100 Manual," pp. 1-6, June 2009.
[51] CNET. (2009, September 20, 2011). *Logitech Webcam C600 Specs* [Website]. Available: http://reviews.cnet.com/webcams/logitech-webcam-c600-web/4507-6502_7-33783072.html

[52] Arduino. (2011, September 20, 2011). *ArduinoBoardADK* [Website]. Available: http://www.arduino.cc/en/Main/ArduinoBoardADK
[53] R. Tracker, "MODEL RT-50A Installation Instructions," 2008 2008.
[54] Maxim, "Programmable Resolution1-Wire Digital Thermometer," ed, 2008, pp. 1-2.
[55] Amazon. (2011, September 20, 2011). *Verizon Motorola Droid A855 Android Cell Phone* [Website]. Available: http://www.amazon.com/Verizon-Motorola-A855-Android-contract/dp/tech-data/B0046NR5PK/ref=de_a_smtd
[56] Newegg. (2011, September 22, 2011). *Dell XPS 15 (L502x)*. Available: http://www.newegg.com/Product/Product.aspx?Item=N82E16834200291&nm_mc=OTC-Froogle&cm_mmc=OTC-Froogle-_-Notebooks-_-DELL-_-34200291
[57] L. X. Wu and G. Z. Cheng, "Speed Management on Icy and Snowy Pavement of Urban Road," *Advanced Materials Research,* vol. 225-226, pp. 593-596, April 2011.
[58] M.-T. Do, "Model for Estimating Expectable Braking Distance," Intelligent Roads2/7/2006 2006.
[59] Omega, "The RS-232 Standard," ed, 2011.
[60] USB.com, "USB 3.0 Specification," ed, 2011.
[61] Arduino. (2011, September 21, 2011). *Wire Library*. Available: http://www.arduino.cc/en/Reference/Wire
[62] IICBUS. (2011, September 21, 2011). *I2C-Bus: What's that?* Available: http://www.i2c-bus.org/
[63] Science Buddies. (2011, September 21, 2011). *Laser Safety Guide*. Available: http://www.sciencebuddies.org/science-fair-projects/project_ideas/Phys_Laser_Safety.shtm
[64] US Department of Transportation. (2011, September 21, 2011). *Manual on Uniform Traffic Control Devices (Section 3A.06)*. Available: http://mutcd.fhwa.dot.gov/pdfs/2009/pdf_index.html
[65] International. Organization for Standardization, "CARB requirements for interchange of digital information," in *Road vehicles Diagnostic systems Part 2*, ed, 1994.
[66] Autospectator. (2011, September 20, 2011). *2011 Lexus LS 460 L* [Website]. Available: http://www.autospectator.com/cars/models/2011-lexus-ls-460-
[67] BMW. (2008, September 20, 2011). *BMW Night Vision* [Website]. Available: http://www.bmw.com/com/en/newvehicles/7series/sedan/2008/allfacts/ergonomics/night_vision.html

Appendix: Gantt chart

ID	Task Name	Duration	Start	Finish
1	Technical Review Paper	6 days	Wed 8/31/11	Wed 9/7/11
2	Design Proposal	13 days	Thu 9/8/11	Mon 9/26/11
3	Order Parts	30 days	Sun 9/18/11	Thu 10/27/11
4	Intermediate Assessment	7 days	Sat 9/24/11	Mon 10/3/11
5	Draft Project Summary	4 days	Fri 10/14/11	Wed 10/19/11
6	PDR	9 days	Tue 10/18/11	Sat 10/29/11
7	Web Site	67 days	Wed 9/14/11	Thu 12/15/11
8	Final Project Presentation	6 days	Wed 12/7/11	Wed 12/14/11
9	Final Project Demonstration	5 days	Sun 12/11/11	Thu 12/15/11
10	Final Project Report	12 days	Thu 12/1/11	Fri 12/16/11
11	**Prototype**	**73 days**	**Tue 9/6/11**	**Thu 12/15/11**
12	**Computer Vision**	**35.5 days**	**Tue 9/6/11**	**Tue 10/25/11**
13	GUI	30 days	Thu 9/8/11	Wed 10/19/11
14	Object Detection	7 days	Tue 9/6/11	Wed 9/14/11
15	Lane Detection	20 days	Wed 9/14/11	Wed 10/12/11
16	Vehicle Detection	17 days	Thu 9/22/11	Sat 10/15/11
17	GPU Implementation	19.5 days	Wed 9/28/11	Tue 10/25/11
18	**Arduino Platform**	**50 days**	**Sun 9/18/11**	**Sat 11/26/11**
19	Rain Sensor Integration	10 days	Mon 10/10/11	Fri 10/21/11
20	Computer Interface	49 days	Tue 9/20/11	Sat 11/26/11
21	Temp/Humid Sensor Integration	6 days	Wed 9/28/11	Wed 10/5/11
22	Arduino Communication	52 days	Sun 9/18/11	Sat 11/26/11
23	Laser Sensor Integration	26 days	Mon 10/24/11	Sat 11/26/11
24	**Android Platform**	**40 days**	**Mon 10/10/1**	**Fri 12/2/11**
25	GUI	21 days	Sat 11/5/11	Fri 12/2/11
26	USB Accessory Mode	17.5 days	Mon 10/10/11	Wed 11/2/11
27	**Control Architecture**	**50 days**	**Sat 9/24/11**	**Sun 12/4/11**
28	Hardware Interfacing	20 days	Mon 11/7/11	Sun 12/4/11
29	Fused Kalman Filter	50 days	Sat 9/24/11	Thu 12/1/11
30	Inter-object Communicatior	12 days	Thu 11/17/11	Fri 12/2/11
31	Input Hardening	11 days	Thu 11/17/11	Thu 12/1/11
32	Multithreading	31 days	Sat 10/22/11	Fri 12/2/11
33	**Car Implementation**	**42.5 days**	**Sun 10/2/11**	**Wed 11/30/1**
34	Cable Routing	9 days	Thu 11/17/11	Tue 11/29/11
35	Dash Modifications	9 days	Thu 11/17/11	Tue 11/29/11
36	Sensor Installation	9.5 days	Thu 11/17/11	Wed 11/30/11
37	Laser Enclosure	3.5 days	Sun 10/2/11	Wed 10/5/11
38	**Troubleshooting**	**5 days**	**Fri 12/9/11**	**Thu 12/15/11**
39	Hardware	5 days	Fri 12/9/11	Thu 12/15/11
40	Software	5 days	Fri 12/9/11	Thu 12/15/11
41	Road Test	5 days	Fri 12/9/11	Thu 12/15/11