Intelligent Transportation Systems Impact on Traffic Safety

Ryan Jones
Saint John's University
Collegeville, United States
rdjones@csbsju.edu

***Abstract* - Thousands of people die every year in traffic accidents. The goal of intelligent transportation systems is to make the road a safer place to drive. My goal is to analyze how has intelligent transportation systems improved road safety. To do so, we will look at past, current, and future intelligent vehicle technologies to evaluate how they left their mark on improving road safety. Similarly, I will utilize a LEGO Mindstorms NXT kit to demonstrate my understanding of how intelligent vehicles utilize obstacle detection systems to avoid collisions. Currently, the majority of vehicles on the road do not utilize intelligent transportation systems. However, traffic safety is expected to increase as the number of intelligent vehicles on the road rises. As a result, intelligent transportation systems are a boon to the safety and wellbeing of drivers everywhere.**

CONTENTS

LIST OF FIGURES

I. Introduction

Before the advent of intelligent transportation systems, thousands of people died every year in traffic accidents caused by a lack of information and human negligence. Intelligent transportation systems are comprised of new technologies that had been experimented on for the past 30 years with the goal of improving traffic safety and vehicle efficiency. These technologies grant intelligent transportation systems the ability to scan environmental data in real-time, identify potential hazards, and communicate with other vehicles or the driver via alerts or signals. Altogether, intelligent transportation systems are used to assist drivers in making safe driving decisions.

II. Survey of Previous Intelligent Transportation Systems

A. The Birth of Intelligent Transportation Systems

The automobile was the most influential invention of the 20th century [9], yet it also contained flaws and hazards that endangered the lives and wellbeing of millions of people around the world. It is estimated that about 42,000 Americans die every year in traffic accidents [9]. Likewise, traffic jams are responsible for over 3.7 billion lost man hours and “2.3 billion wasted gallons of fuel” [9]. As a result, scientific efforts to create transportation systems capable of improving road safety and vehicle efficiency. Some of the current research projects in the field of intelligent transportation systems include: Partners for Advanced Transit and Highways (PATH), the SafeTrip-21 initiative, the IntelliDrive program, the Cooperative Vehicle Infrastructure Systems (CVIS) project, Complex Embedded Automotive Control Systems (CEmACS), and the Tokyo Smartway [4]. Two systems have made considerable progress in the field of intelligent transportations systems:

- **The ARGO project**: was a prototype vehicle safety system capable of rudimentary obstacle and road detection that could alert the driver of or maneuver the vehicle around environmental hazards [3].
- **The DARPA Grand and Urban challenges**: were contests held by the Defense Advanced Research Projects Agency (DARPA) in order to progress research in the field of intelligent transportation systems [9].

B. The ARGO Project

Figure 1. The ARGO prototype vehicle (from Bertozzi [3]).

Early intelligent transportation systems were capable of autonomously driving from one pre-determined destination to another with driver supervision and had rudimentary systems capable of detecting and avoiding obstacles and other environmental hazards on the road. One of the first prototypes of an intelligent transportation system was the ARGO project. The ARGO project was an experiment performed by Massimo Bertozzi, Alberto Broggi, and Alessandra Fascioli. The goal of the ARGO project was to develop a vehicle safety system capable of both identifying and avoiding environmental hazards through either direct communication of potential hazards to the driver or by autonomously maneuvering the vehicle around hazards [3]. In theory, this system would be able to reduce the total number of traffic accidents and deaths if it was incorporated into the majority of vehicles.

With traffic safety in mind, Bertozzi and his team developed the GOLD (Generic Obstacle and Lane Detection) system using a single Pentium MMX processor and the Linux operating system [3]. The GOLD system was able to detect obstacles by processing a pair of side-by-side images to determine whether or not an obstacle in a specific region of the image changed position overtime [1]. The GOLD system also utilized lane detection by identifying patterns in an image that were present in road markings [1].

In its first test run, the ARGO vehicle drove autonomously on a 2000km journey over the Italian highway network. The Italian highway network was chosen for its "quickly varying road scenarios, changing weather conditions," and busy traffic in order to extensively test the system [3]. Despite encountering some snags in low-light conditions, the experiment was considered a success and served as the first milestone towards intelligent transportation systems.

C. The DARPA Grand and Urban Challenges

Figure 2. Stanley winning the DARPA Grand Challenge (from Thrun [9]).

The next milestone toward intelligent transportation systems was the DARPA Grand and Urban Challenges. The Grand Challenge of 2004 was an autonomous car race that stretched over 142 miles of the Mojave desert and had a prize of 2 million dollars to be awarded to the fastest team to complete it within a 10 hour time limit [9]. However, no team was able to complete the course. As a result, DARPA issued the Grand Challenge again in 2005 with a different 132 mile long route through the Mojave desert [9]. This time four competitors were able to complete the challenge within the 10 hour time limit, which demonstrates that there must have been a dramatic increase in autonomous vehicle technology since the last Grand Challenge [9].

Following the success of the Grand Challenge, DARPA announced the Urban Challenge in 2007. The Urban Challenge required contestants to traverse a maze of city roads, empty parking lots, and navigate around active traffic [9]. This tested the autonomous vehicles by forcing them to choose their own path through the city streets and parking lots to their destination [9]. Similarly, the robots had to be aware of other vehicles on the road as potential risks unlike the Grand Challenge. The success of the Grand and Urban Challenges marks another two important milestones on the road to fully autonomous vehicles.

D. Current State of Intelligent Transportation Systems

Despite the advances we have made into intelligent transportation systems, the majority of them are currently under development and have not yet been fully incorporated into current vehicle systems. Some barriers to full implementation of automated vehicle systems result in robotic cars not currently performing at the same level of a human driver and people generally being uncomfortable with the idea of ceding control of their vehicle to an artificial intelligence [9]. Sebastian Thrun proposes that all aspects of an intelligent vehicle system must exceed current safety thresholds and possess better user interfaces for greater control before people will feel comfortable driving an autonomous vehicle [9].

III. Technical Aspects of Intelligent Transportation Systems

A. Sensors

All obstacle and road detection system use sensors to scan their environment for sensory data to then use as the foundation of their analysis. For the past 20 years, the speed of computational processors has increased exponentially allowing for the development of faster and more accurate sensors [8]. There are two kinds of sensors [8]:

- **Active sensors**: detect objects by emitting signals that measure the distance between the sensor and the reflecting signal. Active sensors come in three varieties:
  - **Radar-based sensors**: emit radio waves for detecting objects.
  - **Laser-based sensors**: emit "electromagnetic radiation" at high frequencies.
  - **Acoustic-based sensors**: emit ultrasonic sound waves.
- **Passive sensors**: utilize vision-based detection systems to obtain information in "a non-intrusive way," but has a higher computational cost than active sensors.

Other sensors like Global Positioning Systems (GPS) can operate a vehicle through its relation to the road, its heading, and its GPS localization [13]. The benefits and costs of both active and passive sensor systems must be weighed against each other to determine which system best satisfies the growing demand for intelligent transportation systems.

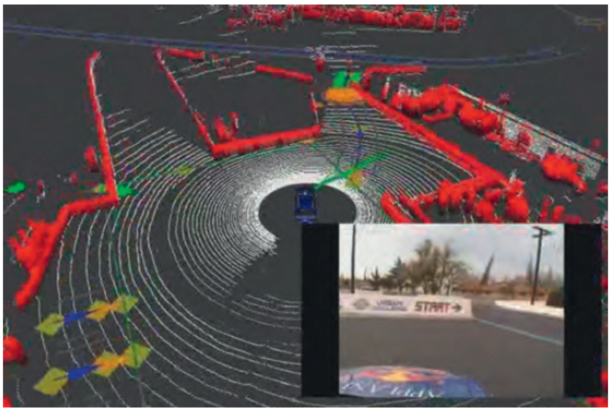

Figure 3. 3-D scans acquired through laser range finer with 64 scan lines. Shown here is a single laser scan, along with the corresponding camera view of the vehicle (from Thrun [9]).

1) Active Sensors: The primary benefit of active sensors is their ability to measure a fixed distance quickly with minimal computations [8]. For example, radar-based sensors can detect obstacles 150 meters away in poor weather conditions (i.e., fog, rain, and snow), whereas a driver could only see 10 meters ahead in similar conditions [8]. Meanwhile, laser-based systems are less expensive and more accurate than radar-based sensors, but struggle in poor weather conditions and have higher operational costs [8]. An example of a laser-based sensor can be seen in Figure 3 where the reflected laser signals are rendered into a 3-D scan of the vehicles surroundings [9]. Even though laser-based systems can accurately detect road boundaries on two-lane roads, they are unable to do so on multilane roads without assistance from vision-based sensors [6]. Another downside of active sensors is the interference that occurs when there is too many vehicles traveling the same direction with similar active sensors [8].

2) Passive Sensors: Vision-based sensors can more easily detect and identify lane markings, traffic signs, and objects than active sensors regardless of road infrastructure. As discussed by Sun et al. [8], vision-based sensors do not suffer interference from other sensors because of its “nonintrusive” data collection procedures. Vision-based sensors can also detect vehicles changing lanes and around curves better than active sensors. However, vision-based sensors perform poorly in “extreme weather or off-road conditions.” According to Yenikaya

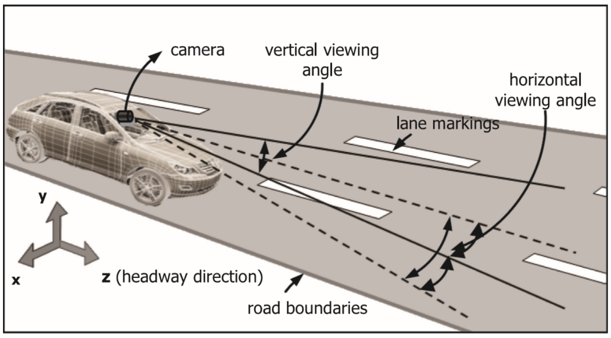

Figure 4. Vision-based road detection system (from Yenikaya [13]).

et al., Vision-based sensors are also plagued with “high computational costs” due to continuously changing road environments [13]. These road environments cause an image to not contain accurate information for periods longer than five to ten seconds. However, the scan rate of passive continues to improve thanks to advances in computational processors. Passive sensors are also cheaper to produce than active sensors [8]. As a result, passive sensors can be inexpensively mounted to both the front and back of a vehicle to obtain a greater field of view than active sensors for a similar cost.

3) Sensor Analysis: Based on our previous analysis of active and passive sensors, it is better to utilize both types of sensors in conjunction to resolve any shortcomings entitled with using only one type of sensor. For example, radar-based sensors would be used in off-road or poor weather conditions to detect oncoming obstacles, because the optical sensor would be rendered ineffective in those circumstances [6]. Similarly, the vision-based sensor wouldnt experience interference in busy traffic from other sensors and could adapt to various road conditions [8]. However, the computational costs associated with using both systems together is greater than using either one individually [6]. This cost can be decreased through faster processors and the development of systems that efficiently incorporate both types of sensors [6].

The data provided by these sensors is then incorporated into various lane and obstacle detection algorithms which all tend to follow a similar pattern. The pattern for interpreting the raw unstructured data into a usable road model includes four steps [13]:

1) **Preprocessing**: sections sensory data into “Regions of Interest (ROI)” to conserve time by not processing the entire image for each scan and to eliminate noise.
2) **Feature Detection**: analyses preprocessed images to determine if the shapes or lines in the image belong to real world objects, such as roads or vehicles.

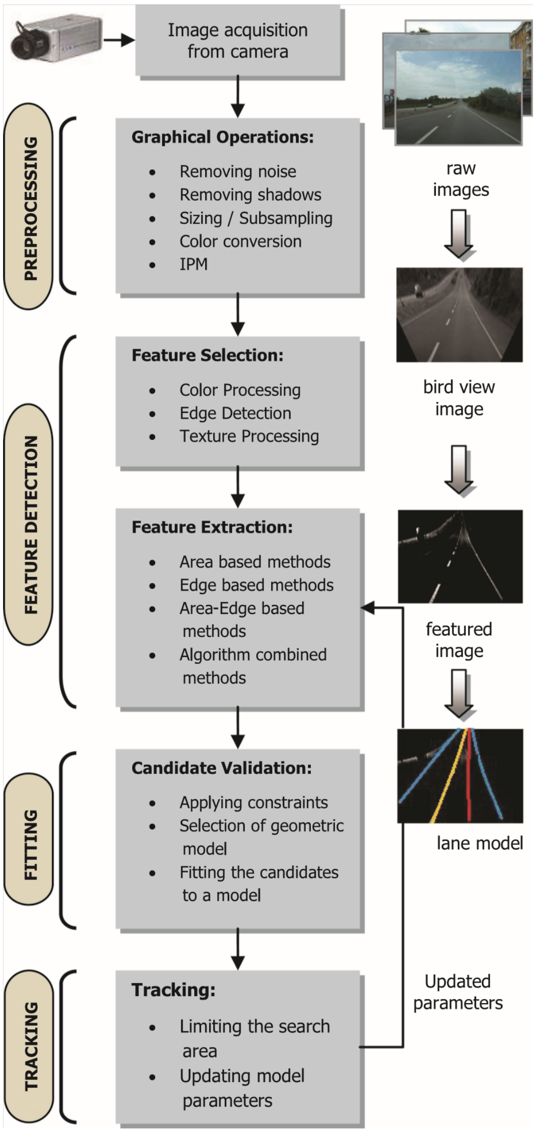

Figure 5. The four steps of road model creation (from Yenikaya [13]).

3) **Fitting**: applies rules to the set the features to assist in predicting their future location and trajectory.
4) **Tracking**: analyses the accuracy of the applied rule set.

This paper will be primarily focusing on the analysis of preprocessing and feature detection.

B. Preprocessing

The size and locations of the ROI possess different meanings for lane detection and tracking algorithms. The lane detection algorithm requires the ROI to be large enough to contain enough useful information to be able to identify the location of the roads lane boundaries, but not so large that the ROI loses its advantage

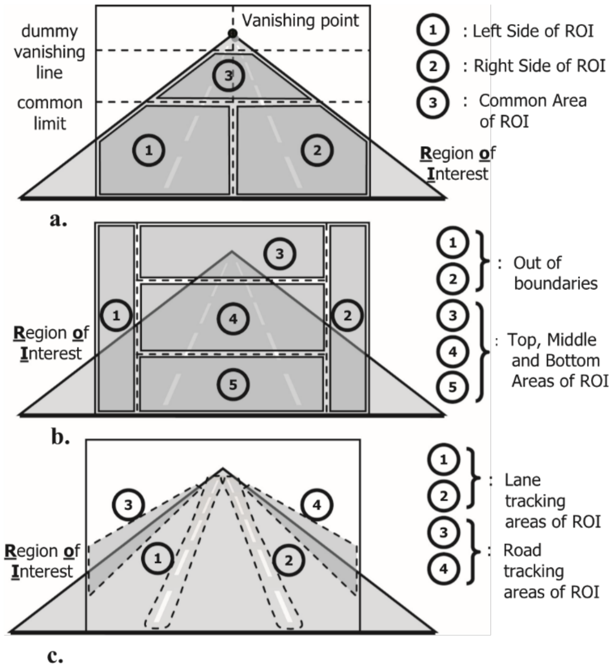

Figure 6. “Regions of Interest (ROI): (a) Vanishing point based (detection mode); (b) area based (detection mode); (c) area based (tracking mode)” (from Yenikaya [13]).

of saving computational time [13]. This is evident in Figure 6(b) where the left- and right-hand borders of the image are excluded from the ROI [13]. Meanwhile, the lane tracking algorithm has “prior knowledge of the road geometry,” so tighter boundaries can be imposed upon the ROI for the possible locations of new lane edges [13]. Several assumptions about the roads features must consistently hold true in various road conditions before any feature can be interpreted. These assumptions include [6]:

- road and lane texture
- road and lane width
- road markings with consistent placements and appearances
- “The road is a flat plane or follows a strict model for elevation change”

Given these assumptions, a foundation of *a priori* knowledge is formed to increase the accuracy of feature detection [6].

C. Feature Selection

Feature detection can be divided into two steps: feature selection and feature extraction. Feature selection translates the ROI from “unrelated pixels” [13] into features such as “edges, motion vectors, and textures” [6] to mark the possible locations of vehicles, objects, and road/lane boundaries. There are three main road features used to perform feature selection. These categories include [13]:

- Color
- Edge
- Texture

These features are also known as knowledge-based methods, because they all utilize prior information about the road environment to predict the location of certain features [8].

1) Color-based Feature Selection: Research into color for the purpose of feature detection has been viewed as a way of harnessing new information that goes beyond standard grayscale or "monochromatic imagery" [13]. Some feature detection systems has used extracted colors from the road and road-markings as a "model to detect lane boundaries" [13]. Jill Crisman and Charles Thorpe developed a prototype that could determine if a selection of pixels belonged to the road by applying a normal or "Gaussian [(bell curve)] distribution" to a "six-dimensional [RGB] color space" [8]. This color space was generated by overlapping the images of "two closely positioned cameras" set to respectively capture the shadowed and sunny areas of an image [8]. However, color processing is not commonly used due to the high computational costs associated with processing RGB color over grayscale colors [13]. The color processor also fails to distinguish between a vehicle and the lane boundary when their colors are similar [13].

2) Edge-based Feature Selection: The detection of pronounced vertical and horizontal edges is a strong indicator for predicting the location of a vehicle or lane boundaries [8]. However, the accuracy of these edges can be dubious at times due to the "noisy" nature of structured roads [13]. For instance, a high contrast is produced between a vehicle and the road, while shadows and worn lane boundaries produce a low contrast [13]. In this case, the edges surrounding the vehicle will stand out in the image, whereas the shadows and worn lane boundaries do not appear as solid edges [13]. Another issue occurs when the edge-based identifier includes unnecessary edges like "trees" and "telephone poles" in the ROI. These unnecessary edges create noise that increases computational costs and distracts the feature extractor from identifying important edges, such as those belonging to vehicles [13].

Sun et al. proposed "a multiscale [edge-based] approach which combines subsampling with smoothing" to provide a more resilient method for predicting vehicle locations [8]. Sun's multiscale approach incorporates four steps [8]:

Figure 7. Multiscale hypothesis generation at three different image sizes: 90 x 62 (first row), 180 x 124 (second row), and 360 x 248 (third row) (from Sun [8]).

1) **Low pass filtering**: generates the "profile" or rough location of objects in the image (see first column of Figure 7).
2) **Vertical edge detection**: is applied to the low pass profile to generate the "vertical edge map" portraying the image's vertical contours (see second column of Figure 7).
3) **Horizontal edge detection**: is separately applied to the low pass profile to generate the "horizontal edge map" portraying the image's horizontal contours (see third column of Figure 7).
4) **Local maxima and minima detection**: is applied to the combination of the previous two profiles to generate their "the peaks and valleys" (see fourth column of Figure 7). These "peaks and valleys" predict the likelihood of whether or not a vehicle is present in the image [8].

Sun's multiscale approach is more resilient to noise, because it breaks an image down to the "lowest level of detail" to only display objects with robust structural features [8]. It is also faster than other edge-based methods, because it uses "low-resolution images" with simple structural features for its analysis instead of high-resolution images with complex structural features [8].

a. **b.**

Figure 8. Texture analysis: (a) Original image; (b) refined image (white fields point to the road, blue fields point outside) (from Yenikaya [13]).

3) Texture-based Feature Selection: As discussed by Sun et al. [8], texture segmentation is a commonly used detection method because of its ability to constrict "the search area for vehicle detection" based on the varying textures in the image. For texture segmentation, entropy measures the intensity or difference between a pixel and all of its adjacent pixels to determine if the selected pixel belongs to the same object as its adjacent pixels. High entropy regions are analyzed for possible objects, because they are more likely to contain objects than low entropy regions. An example of texture segmentation can be seen in the refined image for Figure 8(b), where the white section represents the road and the blue sections represent objects that are not included as being part of the road [13].

Another method of computing texture analysis utilizes co-occurrence matrices rather than entropy. Similarly discussed by Sun et al., a co-occurrence matrix includes the predicted co-occurrence probabilities of different pairs of pixels based on their "predefined geometrical and intensity constraints" [8]. These constraints include "energy, contrast, entropy, and correlation." Co-occurrence matrices have better accuracy than the previous "entropy-based methods" as a result of using four obstacle detection measurements instead of entropy exclusively. However, co-occurrence matrices are more computationally expensive than "entropy-based methods" due to its additional measurement comparisons.

4) Feature Selection Comparison: In comparison of the previous three feature selection methods, a combination of both edge-based and texture-based methods yields the best feature selection results. Color-based methods are susceptible to changing environmental conditions like "illumination" and weather [8]. Both texture- and color-based methods have higher computational costs than edge-based methods due to multiple pixel comparisons [8]. Edge-based methods maintain a high accuracy rating on structured roads with painted lane and road markings [13]. They can also quickly recover from false positive vehicle identifications by using the aspect ratio of a vehicle as a guideline for its identification [8]. However, texture-based methods performs better on unstructured roads than edge-based methods due to the lack of road markings [13]. As a result, an algorithm that utilizes texture and edge data for feature selection would be more accurate than using either of the two individually [13].

D. Feature Extraction

Feature extraction processes the data it receives from feature selection to "extract image features of road areas, road markings, or road boundaries [through] various filters or statistical methods" [13]. There are two types of methods used for feature extraction including:

- Appearance- or Area-based methods
- Template- or Edge-based methods

1) Area-based Feature Extraction: As discussed by Sun et al. [8] and Yenikaya et al. [13], area-based methods, also known as appearance-based methods, utilize classification techniques to determine if the features in the image belong to the categories of either the "road or nonroad" or a vehicle/nonvehicle. An example of

Figure 9. Area detection: (a) Original image; (b) area detected image (pure black fields) (from Yenikaya [13]).

this can be seen in Figure 9, where 9(a) represents the original image, while 9(b) displays "the extracted road area" as a black region in the image [13]. The key to developing a strong classifier for an appearance-based method depends upon how well its training set performs during its initialization. The training set of an area-based classifier is composed of features taken from a list of training images that are then applied to initialize the probability threshold between two or more classes (i.e. vehicle vs nonvehicle or road vs nonroad). The probability threshold is utilized to determine if a certain feature belongs to one class or another. As a result, the accuracy of the classifier is dependent on the quality of the training set. One area-based classifier is the "Support Vector Machine" [13]. A Support Vector Machine processes the RGB values of an image to classify each pixel as belonging to either a road or nonroad surface. However, Support Vector Machines are incapable of identifying the class of previously untried data.

One example of an area-based method involves "using Gabor filters 'for the purpose of' vehicle feature extraction" [8]. In practice, Gabor filters are individually applied onto the various subdivided sections of an image and returns "the mean, the standard deviation, and the skewness" of each section. These values are then used to acquire the orientation of the different sections of texture throughout the image. The various texture orientations are then employed to locate "strong edges and lines" that are commonly present in vehicles. Gabor filters have "an accuracy of 94.81%" while using a Support Vector Machine as its classifier [8].

2) Edge-based Feature Extraction: As discussed by Yenikaya et al. [13], edge-based methods, also known as template-based methods, utilize extracted edge data to classify certain features as a vehicle or as a road/lane boundary. Figure 10 provides an example of edge extraction. First, the 10(a) original image is converted into 10(b) an edge map by an edge-based feature selection process like Canny or Sobel filters. Then, the resulting edges 10(c) and 10(d) can be processed using a Hough transform to decipher if the edge data belongs to either

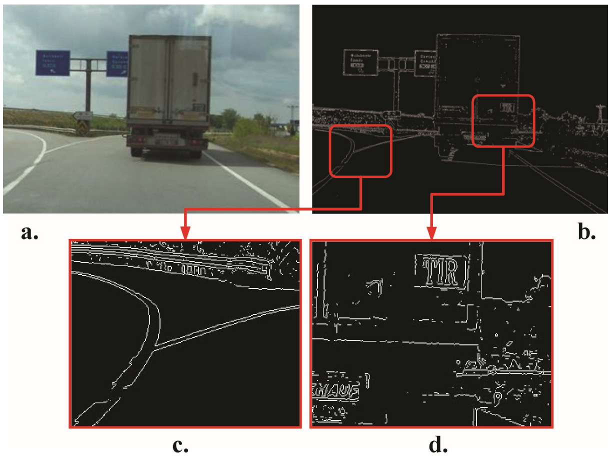

Figure 10. Edge detection: (a) Original image; (b) edge detected image; (c) and (d) details of the edge detected image (from Yenikaya [13]).

a vehicle or road/lane boundary. The “standard Hough transform algorithm” evaluates clusters of pixels to ascertain if an edge is present in the pixel data [13]. However, standard Hough transforms are not applicable to grayscale images. As a result, the edge selection tool has to display the outline of various levels of brightness (i.e. shadows) with an edge before the standard Hough transform can process it.

A template proposed by Bertozzi et al. predicted the location of vehicles on the basis that a vehicle is contained within “a rectangular bounding box which satisfies specific aspect ratio constraints” [1]. If an edge map contained two bottom corners within the specified size and perspective restrictions, then the algorithm would proceed to search for the correlating top corners [8]. Upon locating the top corners, the algorithm would mark the resulting bounding box as the location of a vehicle [8]. Despite its fast computational speed, the algorithm is capable of mistaking distant buildings and other non-vehicular objects as vehicles because of its loose bounding box requirements [8].

3) Feature Extraction Comparison: As discussed by Sun et al. [8] and Yenikaya et al. [13], the union of an area- and edge-based feature extraction algorithm, also known as an area-edge-based method, is a more robust system than using either of the two algorithms separately. Area-based methods are more accurate than edge-based methods, because area-based methods use a classifier training set to predict what a collection of pixels represent. The same classifier training set requires a significant amount of computations to run. Therefore, area-based methods are more computationally expensive than edge-based methods. However, computational costs are expected to decrease overtime as processor speeds continue to increase exponentially. Meanwhile, edge-based methods depend on road and lane markers to maintain its accuracy. As a result, poor road conditions hinder the effectiveness of both edge- and area-edge-based methods. When combined, an area-edge-based algorithm can perform edge detection and road area extraction in tandem. One example of an area-edge-based algorithm used by Tsai et al. could classify a pixel as belonging to either a “road surface, lane markings, [or] nonroad object” based on its “smoothness, color, and lane-marking segmentation” [13]. Having access to both edge detection and road area extraction simultaneously allows for greater accuracy at the cost of computational time due to the total increase in the number of features used for classification.

IV. DEMONSTRATION

Figure 11. A LEGO Mindstorms NXT robot

My demonstration of these road model creation techniques utilizes a LEGO Mindstorms NXT kit to create a robot capable of obstacle detection. I chose to use this as my demonstration because it performs obstacle detection using sensors and feature detection techniques that are similar to an intelligent vehicle system. Lego Mindstorms NXT is a robotics development kit composed of “619 pieces” that can be assembled into a robot [12]. One of these pieces is an ultrasonic sensor that is visible at the top of the robot in Figure 11. This active sensor emits ultrasonic sound waves that are used to measure the distance between the sensor and its reflected sound waves. The effective range of this sensor is 170 cm. If the robot does not receive its reflected signal, then it marks the distance to the next object as out-of-range with a value of 255. Actual vehicles would not use an acoustic sensor because it has less range than a radar- or laser-based sensor [2].

Another piece of this robot is called the “brick” (See Figure 12). The brick is the computational processor of the NXT robot and runs on a java-based language

Figure 12. A LEGO Mindstorms NXT brick. From(Lejos.com [12])

called LeJOS [12]. I have written a LeJOS program that enables the robot to perform the following motions:

1) Scan with the ultrasonic sensor to detect any nearby objects
2) If there are no objects ahead, move forwards
3) Otherwise, stop and turn to the left
4) Repeat steps 1 through 3

The source code for the program:

```
import lejos.nxt.*;
import lejos.util.*;

public class MoveToOb {

//motor and sensor values
private static final int NORMAL_MOTOR_SPEED = 540;
UltrasonicSensor motionSensor = new UltrasonicSensor(
  SensorPort.S1);
int distance = 255;

  /**
   * Robot drives forward until an obstacle is
   * detected, turns left until the obstacle is
   * no longer in its path, and then continues
   * forward.
   * @author rdjones
   * @date 4.29.2014
   */
  public void MoveToObstacle() {

   Motor.A.setSpeed(NORMAL_MOTOR_SPEED);
   Motor.B.setSpeed(NORMAL_MOTOR_SPEED);
   motionSensor.setMode(1); //Ping

   while(!Button.ESCAPE.isPressed()) {

    //Scan for objects
    motionSensor.ping();
    distance = motionSensor.getDistance();

    //Is there an object ahead?
    if(distance < 45){

      //Stop the Robot
      Motor.A.stop();
      Motor.B.flt();

      //Turn Left
      Motor.B.rotate(-180);

      //Scan to see if object is cleared
      motionSensor.ping();
      try{
        Thread.sleep(250);
      } catch (InterruptedException ie)
    }
    else{ //Drive Forwards
      Motor.A.forward();
      Motor.B.forward();
    }
   }
  }
  public static void main(String[] args) throws Exception {
    MoveToOb RunRobot = new MoveToOb();
    RunRobot.MoveToObstacle();
  }
}
```

The algorithms that control intelligent transportation systems are significantly more advanced than the simple program listed above. However, the NXT robot possesses the basic functions of an obstacle detection system. It demonstrates this through its ability to detect an object and maneuver itself until the object no longer remains in its path.

V. Future Trends

Currently, intelligent vehicle systems are capable of limited autonomy thanks to “machine learning” methodologies which permit them to modify themselves based on their surroundings. One machine learning system is the Stanley system used by Stanford’s robot Stanley to win the DARPA Grand Challenge [9]. The Stanley system utilized an “online self-supervised learning system” to modify its classifier regarding whether a certain piece of terrain was drivable or not [5]. Stanley and other machine learning systems are capable of incorporating new environmental information with previous classifier data to assist in classifying future occurrences of that environment [8]. As a result, the Stanley system can predict whether certain terrains are drivable at even greater distances [5]. On the other hand, machine learning systems need to be able to prioritize protecting the driver during an emergency. For example, a self-supervised vehicle would engage its emergency brakes after a sudden and unavoidable obstacle appears rather than calculating if it could maneuver out of the way in time [5]. However, machine learning techniques must acquire even greater accuracy before it can be trusted for public usage [9].

A. Vehicle Area Networks

Vehicle area networks encourage safe driving environments by enabling a cooperative hazard detection system and by monitoring abnormal driver behavior. A vehicle area network is a network that enables cooperative communication between the driver, their vehicle, other nearby drivers, and roadside infrastructure in order to provide information about both internal and external hazards [4]. Vehicle area networks have two methods of communication [4]:

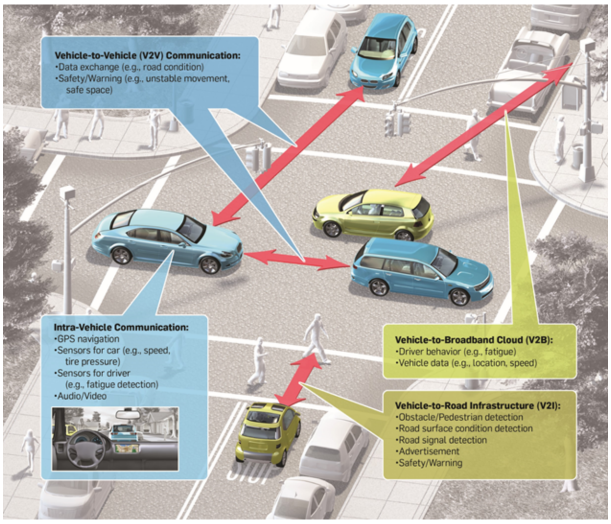

Figure 13. An example of a vehicle area network (from Faezipour [4])

- **Intra-vehicle communication**: between the driver and their vehicle.
- **Inter-vehicle communication**: between the driver's vehicle, other vehicles, and roadside infrastructure.

1) Intra-Vehicle Communication: Intra-vehicle communication allows the vehicle to detect the driver's current level of fatigue in two ways. One way is through the compilation of the vehicle's current "speed, pressure on the brake or gas pedal, [and] steering wheel rotation" to determine if the driver is performing abnormally [4]. The other way is by observing the driver's "behavioral information," such as "blink rate, yawning, and head movements," to measure the driver's current level of alertness [4]. If the driver is confirmed to be physically unfit to drive, then the vehicle would alert the driver of his drowsiness in an attempt to revitalize the driver [4]. If the driver persists in not awakening from his stupor, then the vehicle's "Cooperative Collision Warning System" (CCWS) [10] would proceed to alert other nearby vehicles of the driver's hazardous exhaustion-induced behavior [4].

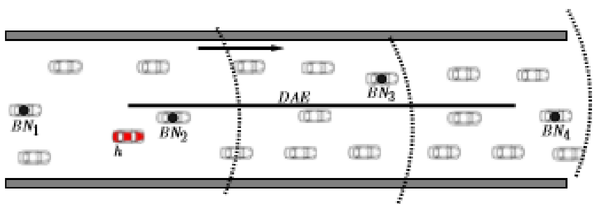

Figure 14. A four-link broadcast chain (from Vasilis [10])

2) Inter-Vehicle Communication: The CCWS allows vehicles to be able to detect hazardous events in their environment by means of inter-vehicle communication made possible by vehicle area networks [10]. The CCWS does this in two ways. First, it has the vehicle responsible for the hazardous behavior broadcast a warning to other nearby endangered vehicles [10]. Second, the vehicle's system should always be on alert for hazardous behavior from other vehicles [10]. Once a hazardous event is detected, a "broadcast chain" is assembled from the vehicles surrounding the hazardous event to warn other drivers in the proximity [10]. The broadcast chain also limits the number of vehicles sending alerts. As a result, endangered drivers quickly receive the alert from the broadcasting vehicle closest to them and reduces the number of overlapping alerts sent from other vehicles [10]. Intelligent vehicle systems that use a CCWS or another similar "co-operative system" possess a "more reliable and robust" vehicle detection system than one that does not share information [8]. A vehicle detection system that has access to the speed and acceleration of close-by vehicles could more accurately predict their location [8].

B. Platooning

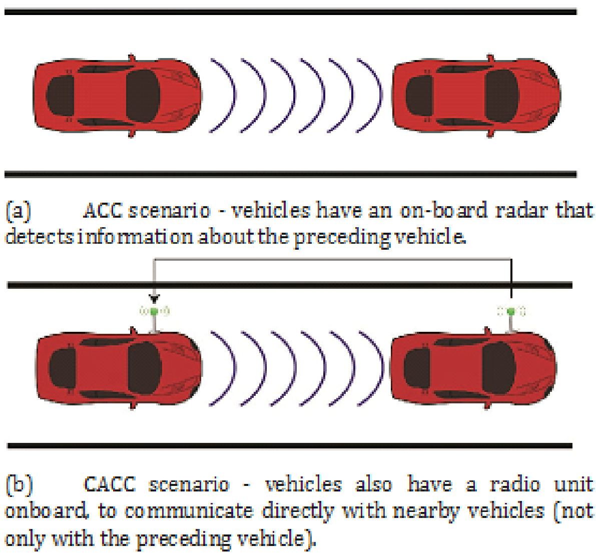

Figure 15. Comparison of ACC- and CACC-equipped vehicles (from Willigen [11])

Research has also been invested into Cooperative Adaptive Cruise Control (CACC), also known as "platooning," to increase vehicle efficiency and traffic safety. Platooning research originated as an attempt to devise a technology capable of "smooth merging and splitting" on dedicated highways [7]. To form a platoon, a vehicle uses its CACC to send radio signals to the vehicle directly in front of it to ascertain its " current velocity, position and acceleration" [11]. Figure 15 displays the current adaptation of this technology in the form of Adaptive Cruise Control (ACC), which uses a radar sensor to track the same information as CACC but in an indirect manner [11]. Meanwhile, CACC requires both vehicles to be equipped with a radio system so they can directly synchronize their platoon's velocity and acceleration [11]. As a result, CACC is more accurate than ACC at acquiring this information due to the direct communication between vehicles. Willem van Willigen claims traffic stability and throughput will increase when over 60% of all vehicles possess CACC technology due to platooning vehicles being in constant motion [11]. Similarly, vehicles in a platoon can safely drive within shorter distances of each other since they maintain similar speeds and acceleration [11]. Therefore, vehicles in a platoon are less likely to produce traffic jams or accidents caused by sudden shifts in speed, and conserve fuel wasted on spontaneous braking and acceleration.

VI. Concluding Statements

To this day, intelligent transportation systems continue to evolve to protect and assist drivers. Early adaptations of intelligent transportation systems focused on the development of rudimentary autonomous vehicles. These vehicles were capable of detecting obstacles and lane markings in a road environment, but struggled in certain environmental conditions such as heavy traffic, extreme weather, and low light conditions. Current systems are more resistant to environmental conditions through the use of multiple sensors, faster processors, and more accurate obstacle detection algorithms. The next generation of intelligent transportation systems will allow for greater synergy between the driver and their vehicle. For example, drivers can use vehicle area networks to gain access to useful driving information including the location of nearby environmental hazards. As a result, intelligent transportation systems improve traffic safety by informing drivers of potential threats in time for the driver to make an informed decision rather than an uninformed split-second decision.

References

[1] M. Bertozzi and A. Broggi. Gold: A parallel real-time stereo vision system for generic obstacle and lane detection. *Ieee Transactions on Image Processing*, 7(1):62–81, 1998. Bertozzi, M Broggi, A.

[2] M. Bertozzi, A. Broggi, and A. Fascioli. Vision-based intelligent vehicles: State of the art and perspectives. *Robotics and Autonomous Systems*, 32(1):1–16, 2000. Bertozzi, M Broggi, A Fascioli, A.

[3] Massimo Bertozzi, Alberto Broggi, and Alessandra Fascioli. Autonomous vehicles. *Linux J.*, 1999(59es):3, 1999.

[4] Miad Faezipour, Mehrdad Nourani, Adnan Saeed, and Sateesh Addepalli. Progress and challenges in intelligent vehicle area networks. *Commun. ACM*, 55(2):90–100, 2012.

[5] Leslie Pack Kaelbling. Technical perspective: New bar set for intelligent vehicles. *Commun. ACM*, 53(4):98–98, 2010.

[6] J. C. McCall and M. M. Trivedi. Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation. *Ieee Transactions on Intelligent Transportation Systems*, 7(1):20–37, 2006. McCall, JC Trivedi, MM.

[7] Michele Segata, Falko Dressler, Renato Lo Cigno, and Mario Gerla. A simulation tool for automated platooning in mixed highway scenarios. *SIGMOBILE Mob. Comput. Commun. Rev.*, 16(4):46–49, feb 2013.

[8] Z. H. Sun, G. Bebis, and R. Miller. On-road vehicle detection: A review. *Ieee Transactions on Pattern Analysis and Machine Intelligence*, 28(5):694–711, 2006. Sun, ZH Bebis, G Miller, R.

[9] Sebastian Thrun. Toward robotic cars. *Commun. ACM*, 53(4):99–106, 2010.

[10] Vasilis Verroios, Carmen Ruiz Vicente, and Alex Delis. Alerting for vehicles demonstrating hazardous driving behavior, 2012.

[11] Willem van Willigen, Evert Haasdijk, and Leon Kester. A multi-objective approach to evolving platooning strategies in intelligent transportation systems, 2013.

[12] www.lejos.org. Java for lego mindstorms.

[13] S. Yenikaya, G. Yenikaya, and E. Duven. Keeping the vehicle on the road: A survey on on-road lane detection systems. *ACM Comput. Surv.*, 46(1):1–43, 2013.

Appendix

I will now acknowledge all of the people who assisted me in creating this project. I would first like to thank Professor Heroux for answering all of my questions and providing me feedback that allowed me to evolve my project overtime to its current status. Similarly, I would like to thank Professor Miller for lending me the LEGO Mindstorms NXT kit that I used to complete my demonstration. Professor Miller was always available to answer my questions about the functionalities of the NXT system and also provided me with a tutorial on how to operate its system. I also thank Professor Holey for teaching me the fundamentals of Java programming and object-oriented design which were necessary to complete the project.

I will also reflect upon how Saint John's University's (SJU) Computer Science curriculum gave me the experience necessary to complete my project. Before coming to SJU, I knew nothing about the Java programming language or object-oriented design. However, I gained a great amount of Java programming experience after attending classes like CS200 Data Structures and CS230 Software Development. I put this Java programming experience to use in my project while writing the LeJOS program for my NXT robot. It would have been difficult for me to complete the project without the Java experience I had received from SJU. I expanded upon my previous knowledge of the Java programming language while developing my project by incorporating new proprietor-based API in the form of LeJOS to write my program. As a result, I increased my knowledge of the Java programming language.

I also learned how to collect and incorporate the scholarly information needed to write my paper through my senior research project coursework and Professor Heroux. He taught me how to use websites like LaTex, Endnote, and the ACM library to compile scholarly information into a professional document worthy of the project at hand. I will be able to take this technical writing knowledge with me as I enter the business world and apply it to the workplace. I am truly thankful for all the technical knowledge that I learned from SJU's CSCI curriculum and its professors.