Paper ID #18751

Lane Keeping System by Visual Technology

Mr. Tony Fan, Wayne State University
Dr. Gene Yeau-Jian Liao, Wayne State University

GENE LIAO is currently Director of the Electric-drive Vehicle Engineering and Alternative Energy Technology programs and Professor at Wayne State University. He received a M.S. in mechanical engineering from Columbia University, and a doctor of engineering from University of Michigan, Ann Arbor. He has over 17 years of industrial practices in the automotive sector prior to becoming a faculty member. Dr. Liao has research and teaching interests in the areas of hybrid vehicles, energy storage, and advanced manufacturing.

Prof. Chih-Ping Yeh, Wayne State University

Chih-Ping Yeh received his BS in Electrical and Electrnoic Engineering from TamKung University in Taiwan, MS and Ph.D. in Electrical Engineering from Texas A&M University in College Station, Texas. He is the Chair of the Division of Engineering Technology at Wayne State University in Detroit, Michigan.

Dr. Chung-Tse Michael Wu, Wayne State University

Dr. Chung-Tse Michael Wu is currently an Assistant Professor at the Department of Electrical and Computer Engineering at Wayne State University (WSU), Michigan, USA. His research interests include applied electromagnetics, antennas, passive/active microwave and millimeter-wave components, RF systems and metamaterials. He received his B.S. degree from National Taiwan University (NTU) in 2006. He then received his M.S. and Ph.D. degree in the Department of Electrical Engineering, University of California at Los Angeles (UCLA) in 2009 and 2014, respectively. From September 2008 to June 2014, he worked as a graduate student researcher at the Microwave Electronics Laboratory in UCLA. In 2009, He was a summer intern in Bell Labs, Alcatel-Lucent, Murray Hills, NJ. In 2012, he was a special-joint researcher at Japan Aerospace Exploration Agency (JAXA) in Kanagawa, Japan. In 2016, Dr. Wu received National Science Foundation (NSF) Faculty Early Career Development (CAREER) Award, as well as WSU College of Engineering Faculty Research Excellence Award. He was also a recipient of 2011 IEEE Asia Pacific Microwave Conference (APMC) Student Prize and a recipient of 2013 IEEE APMC Best Student Paper Award. In addition, he received the second place award in 2014 IEEE International Microwave Symposium (IMS) student design competition. He is a Member of IEEE, IEEE-MTTs, IEEE-APS and IEEE-ComSoc.

Dr. Jimmy Ching-Ming Chen, Wayne State University

Assistant Professor 2015-present Division of Engineering Technology Wayne State University. Ph.D 2006 Texas A&M University.

Lane Keeping System by Visual Technology

Abstract

The introduction of vehicle automation, autonomy and connectivity is fundamentally changing the concept of automotive transportation. Although many of these technologies are still in development in lab, some of these technologies are already available and demonstrated by the prototypes such as Google and Toyota self-driving cars. To prepare for the future workforce needs of autonomous vehicles in the automotive industry, we develop new, technologically progressive curricula and hands-on lab as well as student project materials. This proposed "Lane Keeping System by Visual Technology" is a research and concept-proving student project that will be studied and used to develop teaching materials for the subject of vehicle automation, autonomy and connectivity. Lane Keeping System (LKS) is an advanced active safety system, which uses a front-view camera to detect lane lines and distinguish lateral deviation. It will alert drivers when there is unintentional departure from the driving lane, and then actively steer the vehicle back into the driving lane. Vehicles not connected to the infrastructure do not have real time information of their lane position on the road. A visual identification system using a camera is therefore fundamental for a vehicle to obtain the traffic information. In this student project, a webcam and a microcomputer (Raspberry Pi) are connected and mounted on a previously developed modified RC toy car where the car movements are controlled by an Arduino. The obtained images in front of the car are processed by the OpenCV software. The current goal is to identify the horizon line, the roadside lines, and vertical roadside building lines from the images and then drive the car in the "middle" of the road. The test takes place in a long hall way and the RC toy car is able to stay in the middle when moving along the hallway automatically. Student working processes of design, hardware modification, as well as the algorithm and coding procedures are presented. The project activities, the testing results, and student's learning experiences and outcomes are present in this paper.

Introduction

Vehicle automation is fundamentally changing the concept of automotive transportation. Autonomous vehicles are vehicles that are capable of sensing and navigating an environment without direct driver input. Over the past decade, fully autonomous vehicles have become more widespread, including early prototypes in the early 2010s, such as Google and Toyota's self-driving cars, to more recent developments like Tesla's Autopilot system and Uber's self-driving ridesharing service.

According to an analysis by the US National Highway Traffic Safety Administration in 2003, out of 539,000 lane departure related crashes, 11.5% or 62,000 crashes were due to one driver drifting into another driver's lane for no particular reason[1]. Since the lane departure in these crashes were unintentional, they could be reduced or prevented with the use of lane keeping systems which warn the driver with sufficient time to respond or actively keep a vehicle in lane.

Table 1. The Society of Automotive Engineers (SAE) defines six levels for vehicle automation[2]:

Automation Level	**Description**
Level 0 No Automation	The driver is in complete and sole control of the vehicle and all driving related tasks – steering, acceleration/deceleration, monitoring the driving environment, and driving tactics (responding to events, when to change lanes, turn, use signals, etc.) – at all times.
Level 1 Driver Assistance	The driver receives assistance in the form of steering or acceleration/deceleration, but still monitors the driving environment and performs dynamic driving tasks. Lane keep assist and cruise control fall under this category.
Level 2 Partial Automation	The driver assistance system assists in both steering and acceleration/deceleration tasks using information about the driving environment, while the driver remains in charge of the dynamic driving task. Lane keep assist and cruise control used in conjunction with each other fall under this category.
Level 3 Conditional Automation	The automated driving system performs all aspects of the dynamic driving tasks, with the expectation that the human driver will respond appropriately to a request to intervene.
Level 4 High Automation	The automated driving system performs all aspects of the dynamic driving tasks, even if a human does not respond appropriately to a request to intervene.
Level 5 Full Automation	The automated driving system performs all aspects of the dynamic driving tasks under all roadway and environmental conditions that can be managed by a human driver.

The curriculum of vehicle automation, autonomy, and connectivity will mainly cover functions from Level 1 to Level 3, with Level 4 as a target. For Level 1 to 3 work, students will setup modules of sensors, control and communication units, and integrate the modules into a human drivable car and model vehicles, and develop on-road control strategies and algorithms for self-driving testing. The lane keeping system is one step of the curriculum development, covering Level 1 and Level 2 vehicle automation.

In general, lane keeping systems use a video camera to detect road features such as lane markers, and calculate lateral vehicle position, velocity, and lane width in-respect to the lane lines[3]. This method allows the use of existing infrastructure and is easily adaptable to road changes such as construction. However, optical systems are prone to failure in conditions where road features do not exist or are obscured by low sun angles (sunrise, sunset), oncoming headlights, snow, fog, and other environmental factors. GPS based systems are another type of lane keeping system. GPS based systems can operate in all weather conditions, but can fail due to outdated map databases or GPS dropouts from bridges and other terrain.

In this paper, a modified RC car actively steering itself to the center of lane is demonstrated. The project is an application of mechatronics that integrate an optical sensor, actuators (DC motor and servo motor of the toy car), and the control unit (Raspberry Pi). An undergraduate student (now in the EVE graduate program) is assigned to work part time on the hardware modification, algorithm coding, and testing. The major objective of this project is to evaluate the work load and time frame for implementing a similar project on the topic of autonomous vehicles as a student senior project or graduate directed study topic. The working processes and time frame are recorded and evaluated for the development of curriculum. The project platform (toy car) will be used to develop teaching material of other functions of vehicle automation in the future.

Current Lane Keeping Systems in the Market

Many automobile manufacturers provide optional lane keeping systems including Nissan, Toyota, Honda, General Motors, Ford, Tesla, and many more. However, these systems require human monitoring and acceleration/deceleration inputs are not completely automatic. Ford's system[4] uses a single camera mounted behind the windshield's rear-view mirror to monitor the road lane markings. The system can only be used when driving above 40 mph and is detecting at least one lane marking. When the system is active, it will alert the driver if they are drifting out of lane or provide some steering torque towards the lane center. If the system detects no steering activity for a short period, the system will alert the driver to put their hands on the steering wheel. The lane keeping system can also be temporarily suppressed by certain actions such as quick braking, fast acceleration, use of the turn signal indicator, or an evasive steering maneuver. Ford's system also allows the choice between alerting, assisting, or both when active. All these systems use similar strategies in aiding a human driver to stay in lane, but do not allow full autonomous driving[4-7]. GM, in particular, warns that their lane keeping system should not be used while towing a trailer or on slippery roads, as it could cause loss of control of the vehicle and a crash[5].

Project Description

There are two types of lane assist systems: Lane Departure Warning (LDW), which warns the driver of lane departure; and Lane Keeping Systems (LKS), which actively apply steering torque to direct the car back to the center of lane. This project is focused on the lane keeping system. A RC toy car was modified to stay in the center of a lane as it drove along the road by integrating an optical sensor with actuators controlled by a computer and strategy planning/coding. The vehicle platform was not built from scratch, but continued building upon a previously modified RC toy car used in an automatic parking project prior. It used an Arduino Mega board with a high current motor drive shield to control the car's forward, reverse, and turning movements to automatically parallel park the car[8]. The Raspberry Pi LKS controls the car via a USB cable connected to the Arduino board.

The modified RC toy car's LKS is expected to perform the following tasks while driving along the road:

1. Drive along a hallway, while continuously monitoring the driving environment.
2. Keep track of lane markers to determine the car's orientation and lateral position in lane.
3. Adjust its orientation and lateral position to remain in the center of lane.

The LKS has the following major components:

1. The RC toy car consisting of a 7V DC motor in the back and servo motor in the front. The length of this car is 35 cm and the width is 30 cm. The car was previously modified for an automatic parking project developed by another student.
2. Raspberry Pi Model B+. This mini-computer processes images received from the PlayStation Eye camera, and sends control signals to the Arduino Mega controller to steer the car. The lane keeping strategy and algorithm can be programmed and uploaded to the Raspberry Pi.
3. PlayStation Eye camera, shown in Figure 1. The camera is mounted on the car and provides optical data to the Raspberry Pi.
4. Arduino Mega Controller. The Arduino Mega replaces the car's original control board to control the DC drive motor and the turning servo motor. The Arduino receives signals from the Raspberry Pi to drive and steer the car.
5. L298N H-bridge high current motor drive shield, shown in Figure 1. The Arduino can handle a maximum of 200 mA from its VCC and GND pins. The shield provides up to 2 A to drive the car's DC motor.

Figure 1. PlayStation Eye camera (left) and L298N (right)

6. OpenCV software library for Python. This library provides prebuilt image-processing functions used in many computer vision applications.

Lane Keeping System Approach and Algorithms

Lane keeping systems developed by Hyundai and Mitsubishi use a video camera to capture data on lane detection[9-10]. The camera sensor data is composed of lateral deviation, heading angle, and road curvature[9]. Calculating the required steering angle with only the camera sensor data results in the angle for neutral steering of the vehicle[9-10]. Lateral deviation change caused by under steering amount in high acceleration/high banked road situations could not be effectively covered by the calculations for neutral steering. Thus, additional calculations are required to model under steering amount and overcome the lateral deviation change[9]. Mitsubishi's system also takes data from a vehicle sensor in addition to the camera sensor, which are used to calculate a reference steering wheel angle and effective coefficient based on the lane signal (from the camera) and vehicle sensor signal[10]. The steering controller then calculates the reference motor torque based on the reference steering wheel angle and effective coefficient calculated by the lane keeping assist controller, makes the electric power steering system motor operate and output reference motor torque, and assists the driver in manipulating the steering wheel[10]. For the sake of simplicity and time, the student's project only considers a straight road and input signal from a single camera sensor.

1. Processing the image from the PlayStation Eye camera.
   The Raspberry Pi receives images from the PlayStation Eye camera in real-time with a resolution of 640 pixels wide by 480 pixels tall. Each image is converted from color to grayscale. Then the Canny edge detection function is applied to the grayscale image to produce a binary image containing only pixels detected as edges. The Canny edge detection algorithm finds edges based on the intensity gradient, and checking if the gradient exceeds a specified threshold[11]. The top quarter of the resulting binary image is ignored, since the sky does not provide relevant lane marker information and to reduce the processing load on the Raspberry Pi. This binary image is then run through the Hough line detection algorithm to find potential lane markers, which are stored in a vector array in polar coordinate format[12]. The coordinates of each lane marker represent the shortest distance from the origin, top left corner, to the closest point on the line representing the lane marker. An example of the coded process is shown in Figure 2 below.

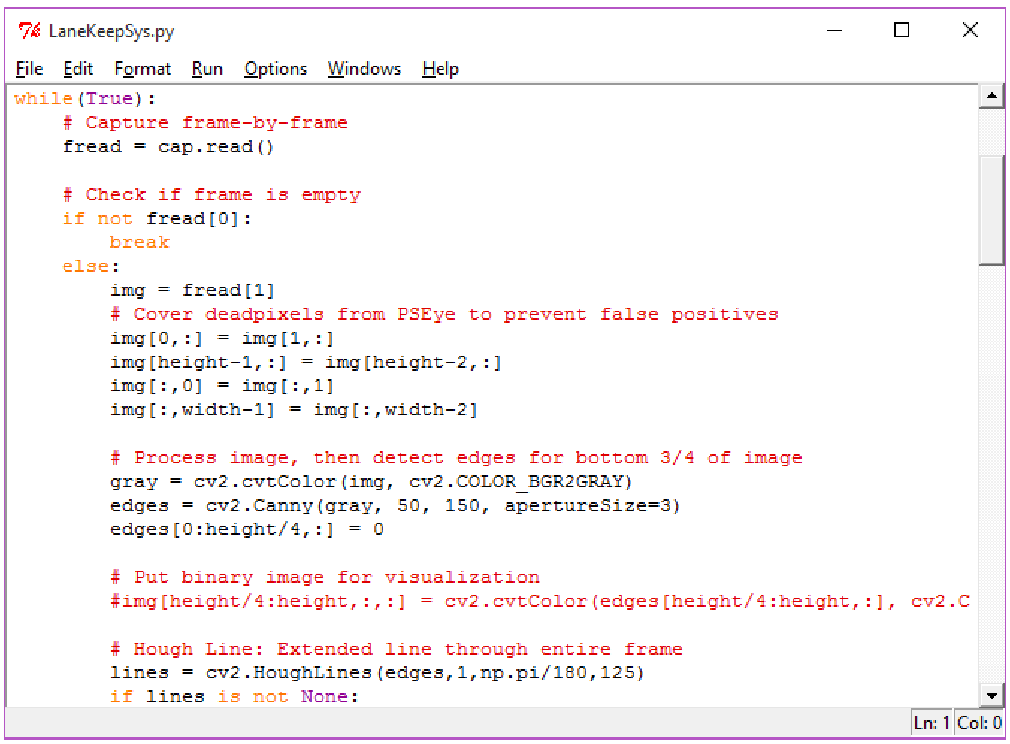

Figure 2. A sample of the coding process used to detect lane markers from the PlayStation Eye.

2. Based on the lane markers detected, find the car's orientation and lateral lane position.
   After the lane markers are detected, they must be filtered to find the two lines that best represent the lane the car is driving in. The filtering algorithm first searches for lines tilted right to represent the left side lane marker, then looks for the line closest to the center of the car. The algorithm then does the same, but using lines tilted left to represent the right side lane marker. An example image after filtering lane markers is shown in Figure 3 below. Once

both lane boundary lines are found, the intersection of the lines is found using two coordinates on each line. With any arbitrary points A and B on one line, and points C and D on the other line, the coordinates of the intersection are found as follows[13]

$$x_{intersection} = \left(\frac{(x_A y_B - y_A x_B)(x_C - x_D) - (x_A - x_B)(x_C y_D - y_C x_D)}{(x_A - x_B)(y_C - y_D) - (y_A - y_B)(x_C - x_D)}\right) \quad (1)$$

$$y_{intersection} = \left(\frac{(x_A y_B - y_A x_B)(y_C - y_D) - (y_A - y_B)(x_C y_D - y_C x_D)}{(x_A - x_B)(y_C - y_D) - (y_A - y_B)(x_C - x_D)}\right) \quad (2)$$

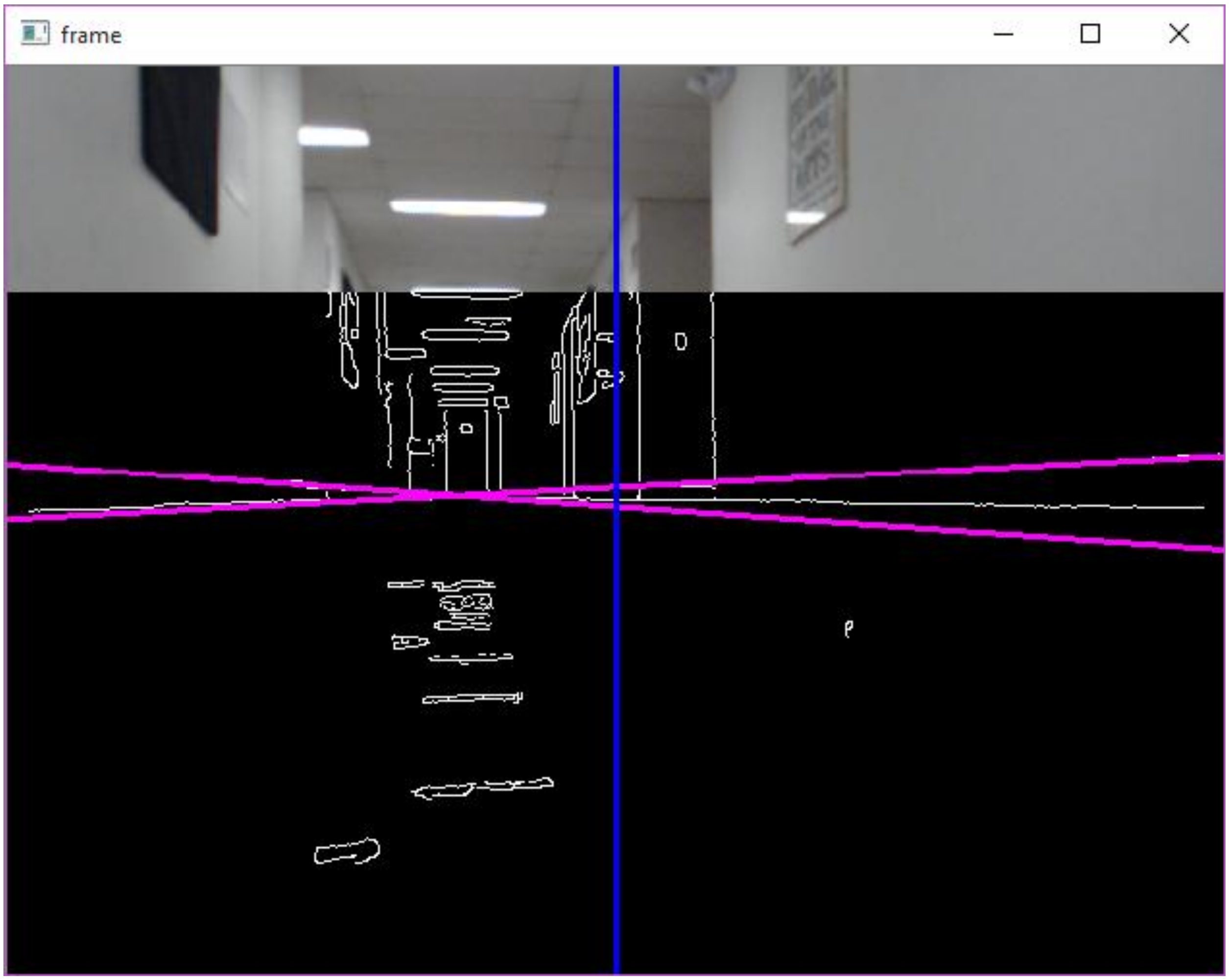

Figure 3. Processed image from the PlayStation eye with the bottom three-quarters representing a binary image of edges detected, magenta lines representing the detected lane boundaries, and vertical blue line representing the center of the car.

The x-coordinate of the intersection from Eq. (1) is then compared to the center of the car to determine whether the car is facing left, right or straight. Using Figure 3 as an example, it is clear the car is facing right as the intersection point is on the left of the car's blue reference line. The car's lateral position is found by bisecting the angle between both lane boundary lines, then extrapolating the bisecting line, representing the center of lane, from the intersection point (from Eq. (1) and (2)) to the y-coordinate (y=480 pixels) at the bottom of the image and comparing the extrapolated x-coordinate to the car's blue reference line. If the center lane line is to the right of the blue reference line, then the car is on the left side of the lane.

3. Finally, use state-based logic to determine how to apply steering torque.

Table 2. Turn recommendation based on the RC toy car's lateral lane position and orientation.

Position \ Orientation	**Facing Left**	**Facing Straight**	**Facing Right**
Left-side of Lane	Steer Right	Steer Right	No change
Center of Lane	Steer Right	No change	Steer Left
Right-side of Lane	No change	Steer Left	Steer Left

While *Table 2* shows the basic logic and accounts of the most likely cases, there are some cases that are not covered. Such as when the car is on the far right-side of lane and barely facing left in orientation. A case like this may require the car steer left to return to the center of lane faster. In order to cover those cases, additional conditions considering these edge cases need to be added separately.

Experimental Results

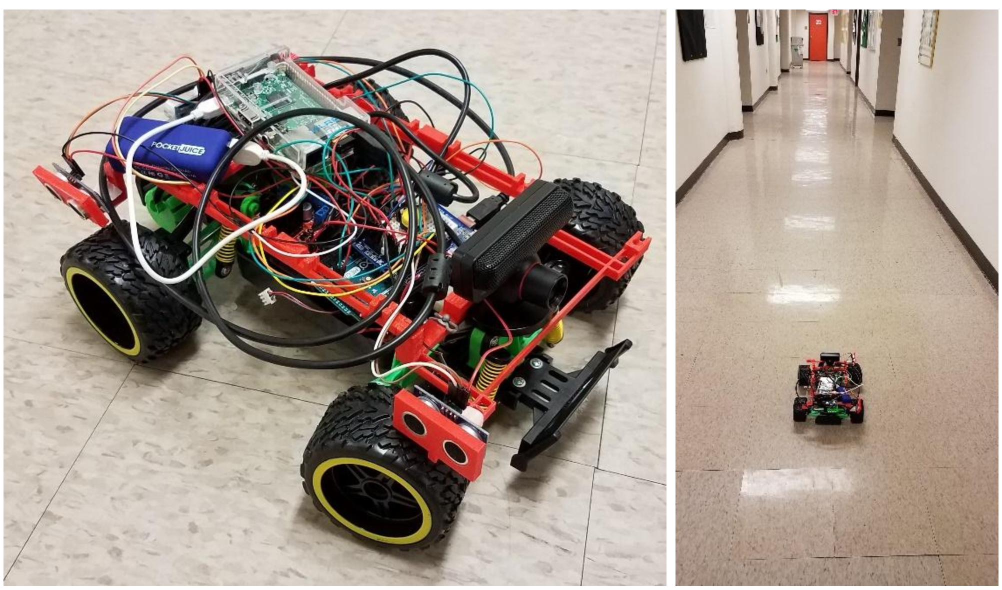

Figure 4. Picture of the modified RC toy car on the left, and picture of the hallway used for testing on the right.

The modified RC toy car and hallway used for testing are shown in Figure 4. During the test, the car is placed in various lateral starting positions and orientations on one end of the hallway. The hallway used for testing is 7.5 feet wide by 43 feet long. When the car started with a clear view of both sides of the hallway, it was able to stay close to the center of the hallway most of the time. However, sometimes the car would travel in an S or zig-zag pattern, by overcorrecting in one direction and then overcorrecting again in the opposite direction about the center of lane. The zig-zagging resulted in lateral deviations of up to 1 foot on either side of the center of lane, and crossing the center of lane up to 13 times while traveling down the hallway. When the car was only able to see one wall of the hallway, it did not always steer correctly back to the center of the hallway, as it would detect both lane boundary lines on the same wall as shown in Figure 5 below. Other times, the car would stop before reaching the end of the hallway due to not detecting any lane boundary lines.

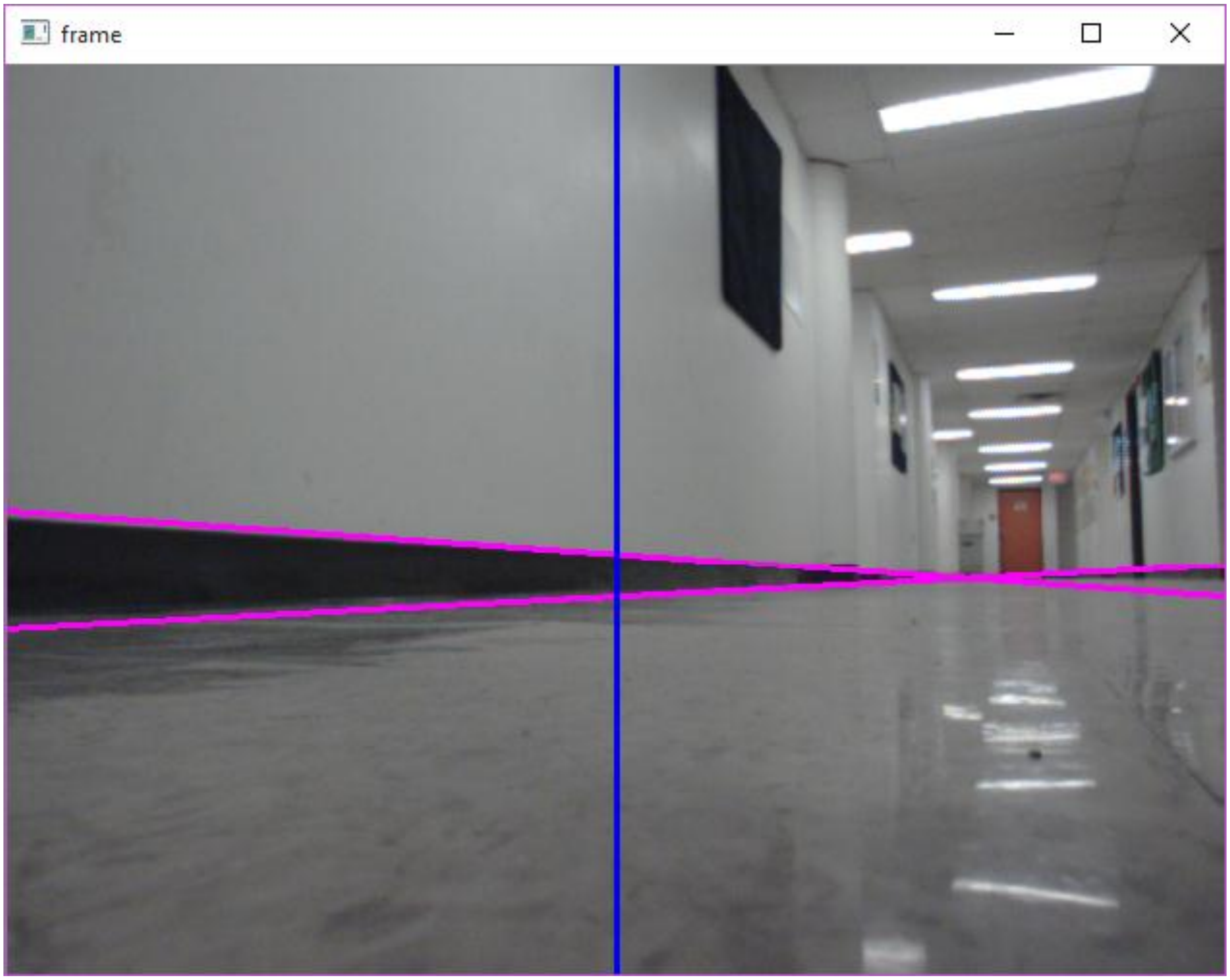

Figure 5. Example of detecting both lane boundary lines on the same wall.

Issues that need to be solved include (1) failure to detect lane boundaries due to the gradient between the baseboard and its reflection on the floor being below the detection threshold; (2) incorrect steering actions when only one lane boundary is detected; and (3) S pattern or zig-zag steering actions when the PlayStation Eye camera is off-center or misaligned. Some fine tuning to the algorithms is needed to improve detection of the lane boundaries. We are planning to implement a fuzzy control system to reduce the zig-zag steering of the car.

Conclusion

The major objective of this project is to evaluate the work load and time frame for implementing a similar project on the topic of autonomous vehicles as a student senior project and graduate directed study topic. It took seven months for the student to modify the vehicle and develop the algorithms to achieve the Lane Keeping System mentioned above, including several weeks of planning and discussion throughout the project. This project is designed for senior or graduate level students who have taken courses such as Instrumentation, Micro and Programmable Controllers, Electrical Machines and Power Systems, Control Systems, and Image Processing as prerequisites. The expected student outcomes, in terms of the capabilities defined by ABET[14], include

General engineering technology (Bachelor):

a. an ability to select and apply the knowledge, techniques, skills, and modern tools of the discipline to broadly- defined engineering technology activities;
b. an ability to select and apply a knowledge of mathematics, science, engineering, and technology to engineering technology problems that require the application of principles and applied procedures or methodologies;
d. an ability to design systems, components, or processes for broadly-defined engineering technology problems appropriate to program educational objectives;
e. an ability to function effectively as a member or leader on a technical team;
f. an ability to identify, analyze, and solve broadly-defined engineering technology problems;

Electrical Engineering Technology:

a. the application of circuit analysis and design, computer programming, associated software, analog and digital electronics, and microcomputers, and engineering standards to the building, testing, operation, and maintenance of electrical/electronic(s) systems;
c. the ability to analyze, design, and implement control systems, instrumentation systems, communications systems, computer systems, or power systems;
d. the ability to apply project management techniques to electrical/electronic(s) systems.

In addition, students will benefit from hands-on experience with strategic analysis and coding, while also preparing them for the potential career in the future automotive industry.

References

1. National Highway Traffic Safety Administration. "Analysis of Lane Change Crashes." Retrieved February 4, 2017.
2. SAE International. "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles." SAE International, September 2016.
3. National Highway Traffic Safety Administration. "Lane Departure Warning Systems." Retrieved February 4,

2017
4. https://owner.ford.com/how-tos/vehicle-features/safety/lane-keeping-system.html
5. http://gmauthority.com/blog/2014/11/gms-lane-departure-warning-and-lane-keep-assist-tech-feature-spotlight/
6. http://www.toyota-global.com/innovation/safety_technology/safety_technology/technology_file/active/lka.html
7. http://owners.honda.com/vehicles/information/2015/CR-V/features/Lane-Keeping-Assist-System/2
8. Liu, H., & Liao, G. Y., & Yeh, C., & Chen, J. C. (2016, June), "Automatic Parking Vehicle System." Paper presented at 2016 ASEE Annual Conference & Exposition, New Orleans, Louisiana. 10.18260/p.26359
9. JaeHee Kim, Sang Min Lee, Sung Kwang Shin, and Sang Ho Jeong. "Development of Active Lane Keeping Assist System." SAE International, April, 2013.
10. Takayuki Tanaka, Shunsuke Nakajima, Takahiro Urabe, and Hideyuki Tanaka. "Development of Lane Keeping Assist System Using Lateral-Position-Error Control at Forward Gaze Point." SAE International, April 2016.
11. http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html
12. http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/hough_lines/hough_lines.html
13. http://mathworld.wolfram.com/Line-LineIntersection.html
14. http://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-engineering-technology-programs-2017-2018/#studentoutcomes