בּ"ה

Visual Interfaces to Computers - COMS 4735
Professor John R. Kender
Columbia University
Spring 2015

Mimoun Cadosch Delmar (mpc2160)

Final Project Proposal

Lane Estimation for Autonomous Driving in a Highway Setting

1. Introduction

The advent of autonomous cars seems inevitable. After the 2004 DARPA challenge, many research institutions and businesses have invested substantial energy and funds in this area, and great progress has been made.

Autonomous cars need a high level of context awareness in order to operate safely and efficiently. In order to obtain situational awareness, autonomous cars are equipped with a wide range of sensors, including laser range finders and cameras. Moreover, these cars are endowed with powerful computers to process the data gathered from the sensors.

Context awareness is a broad problem: it involves knowing the location of other vehicles, pedestrians, cyclists, and even construction sites. For a car to be aware of its context, it also needs to estimate the location of the road and its lanes.

This project deals specifically with the subject of lane estimation. Lane estimation consists of (i) robustly detecting the lanes on the road, (ii) estimating the position of the car with respect to the lanes, and (iii) interpreting the messages conveyed by the painted markings (e.g. solid yellow and white lines separate traffic traveling in opposite and similar directions, respectively).

Many researchers in the computer vision and robotics communities have studied and developed systems of lane estimation for autonomous driving, both in the context of highway and urban driving. And many production systems (Google driverless car, Tesla highway assistance) use lane estimation software. The inspiration for this project is rooted in this research.

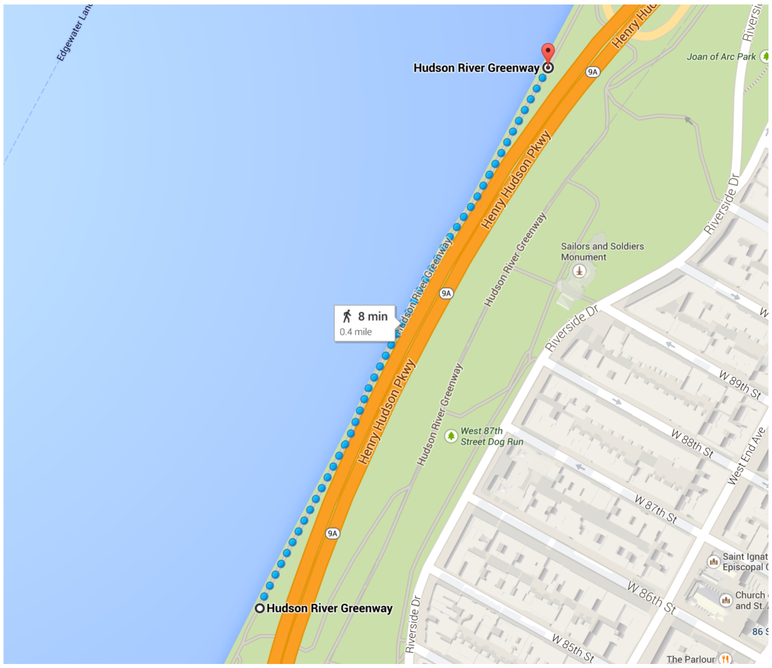
The goal of this project is to use lane estimation, along with a closed-control loop, in order to drive a car autonomously in a highway setting. For safety and logistical reasons, we will not be developing and testing this system for a real car. Rather, we will simulate a real-world system by driving an remote controlled (RC) car in a pedestrian road. Specifically, the pedestrian bridge in the Hudson River Greenway, running from 84th st. to 92nd st., is very suitable for experimentation as it is marked with lines that can be considered similar to those of a regular highway **(figure 1)**. The length of this bridge is approximately 0.4 miles.

Considerations

Instead of this simulation, which approximates highways with pedestrian roads, I considered using stock footage of real highways as the visual input for this program. This approach, however, makes it very difficult to evaluate the system performance.

Performance evaluation requires a ground truth to compare the results of the lane estimation algorithms. Researchers have done the painstaking job of manually labeling lane geometries for kilometers of highway. Dr. Huang's doctoral thesis at M.I.T. [1] goes great lengths to explain how this data was obtained. This data, unfortunately, is not publicly available.

By testing the system in a real environment, on the other hand, we can gather the ground truth data necessary to evaluate our performance. This data consists of road and lane width, as well as vehicle location.



(a)

(b)

Figure 1: (a) Map of pedestrian bridge in the Hudson River Greenway, between 84th and 92nd streets, which spans approx. 0.4 miles (b) Picture of the painted markings on the road

2. Investigation

The problem to be investigated is lane estimation for autonomous driving. As mentioned in the introduction, lane estimation entails: detecting lanes, estimating the position of the car with respect to the lanes, and interpreting the messages conveyed by the road lines. A control loop then processes this information to provide the car with driving instructions.

This program is a visual interface because it takes visual human data (road and lanes) to make clear decisions. These decisions are (i) whether the car is safely driving on the lane, or whether it is drifting to the side. Additionally, (ii) were the car to move out of its lane, the system needs to decide whether this is safe or legal, based on road markings.

3. Project Limits

System Decisions
When it comes to lane changing, this system is restricted to deciding whether it is *legal* to change lanes. Evidently, a more robust system would also determine whether it is *safe* to move to a different lane. For this purpose, however, the system would need to estimate the position and velocity of neighboring cars. This requires technology (laser range finders, cameras) that is outside the resources of this project. Moreover, there are no other simulated cars in our experimental settings, as we are driving in a pedestrian road.

Time of Tests
Most self-driving cars are equipped with laser range finders, which will work at any time of the day. Given that this project uses using low-cost cameras, our system is only meant to function in daylight.

Road Marks
We are using a pedestrian road, that is conveniently marked with lane lines. This road is marked with a solid, yellow line in the middle, and solid, white lines on the sides to mark the limits of the road.

These markings, naturally, are not the same as those found on a real highway. For this reason, our system will modify the meaning of the road lines according to the needs of the different scenarios we plan to test.

For instance, in some experiments the middle yellow line will separate traffic traveling in opposite directions, similar to the traditional meaning of such marking. In other scenarios, this line will be considered a traditional broken, white line. In other words, it will separate traffic traveling in similar directions.

4. Methods and Results Expected

Step 1: Feature Detection

Method

The first step consists of detecting the features for our program. Here, the features consist of painted markings on the road, which correspond to lane boundaries. Our system makes the assumption that the colors of the road and road markings are within a known range. For instance, the road color can be approximated by the color of the gray pavement, and the markings are either white or yellow. Using prior knowledge of object colors, our system can filter the image to isolate the regions of interest (ROI), in this case the painted markings.

Expected Results

Given the particular colors of the pavement and painted markings, we expect the color-based filter to be robust. I understand that colors may vary with the amount of sunlight or road conditions. But the system will attempt to engineer the domain by performing the trials in similar weather and sunlight conditions.

Step 2: Eliminating False Positives

Method

There is a risk of detecting false positives, namely other objects that are not lane boundaries, with color similar to that of painted markings. Fortunately, knowledge of lane geometry can prove very useful in discarding false positives. Our system will apply a horizontal linear filter to images, such that the width of the filter is the expected width of the painted markings on the road. This will ensure that the only features detected are lane boundaries.

Expected Results

We can manually label lanes in a sequence of images, and compute the maximum and minimum values for lane width in the image frame. These values can then be used to determine the width of the horizontal filter. This will enable precise filtering, so that elimination of false positives is robust.

Step 3: Feature Extraction

Method

Feature extraction consists of representing the features detected in an image, so that they can be compared and related to other features. This system will model lanes in a similar way as described in [2]. Specifically, each lane is represented by a straight line $l = [x_l, y_l, \Psi_l]$, where (x_l, y_l) represents the centroid of the shape and Ψ_l its orientation in the image frame (figure 3).

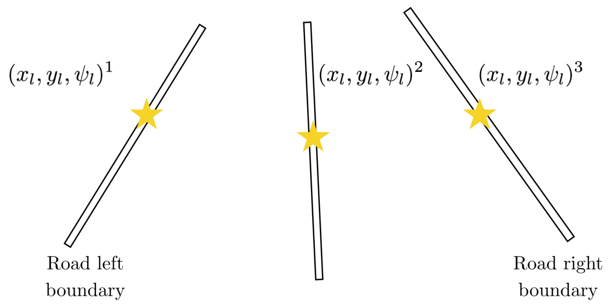


Figure 3: Each lane is described by a line l = [x_l, y_l, Ψ_l], parameters corresponding to the centroid and orientation of the line. The stars denote the centroids of each line.

Expected Results

The road where this system will be tested is, for the most part, straight. There are, however, a few sections with slight curves. This geometry is suitable for our system, since we are simulating a highway setting.

An issue may arise when the vehicle approaches a curve. Then, lanes will no longer appear as straight lines, and representation as such can seem ill-suited. As a solution to this problem, we propose the following approach. (1) First, model lanes as parabolic curves. (2) Then, we compute the derivative to the curve at different points, in increasing order of distance from the vehicle. In figure 4, the derivatives to the curve are computed at the orange points, and are depicted by red lines perpendicular to the lane, shown as a gray curve. (3) As long as the slope of the derivative falls within a given range [m_{min}, m_{max}], the curve can be considered straight. Once the derivative of a point in the curve exceeds this range, the curve is no longer considered straight, and the points onward are not considered part of the feature l. Thus, our system ensures that only the straight portion of the lanes ahead are extracted as features, and so our representation of features as straight lines is suitable again (figure 4).

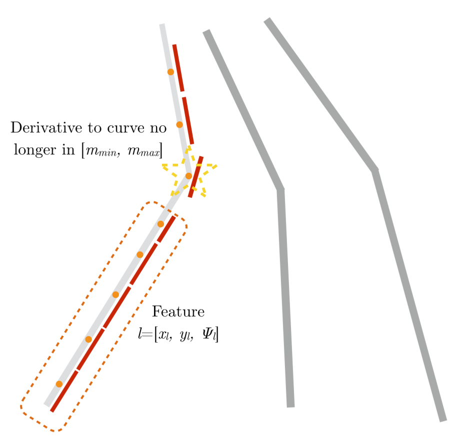


Figure 4: Here we show the derivative of the curve at different points (only for the leftmost lane). When the vehicle approaches a curve, the lanes will no longer appear as straight lines but

rather as curves. Only the segment of the curve with derivatives in a given range will be considered the feature l, as it corresponds to the segment of the lane that is straight.

Step 4: Lane Interpretation

Method

The meaning of a lane will be inferred from its color. Since there are only two types of lanes (white and yellow, solid), our system will modify the meaning of the road lines according to the needs of the different scenarios we plan to test.

Expected Results

The system expects that it will be able to differentiate between white and yellow based on the known pixel range for these colors.

Step 5: Position Estimation

Method

In this step, we must determine which lane the vehicle is driving on. Our system will model the road in a way similar to the one described in [2]. Specifically, lanes will be indexed from 1 to n, from left to right. Evidently, in our case $n = 2$, as there are only two lanes in the pedestrian road. But our system will be prepared to handle greater numbers of lanes.

Our system begins by counting the number of lanes extracted from the images. Next, our system performs perspective projection to compute the transformation between the camera on the vehicle and the left road boundary. The norm of this transformation is denoted by t_{norm} (figure 5). Experimentally, we will find the ranges of t_{norm} corresponding to each lane. Then, by continuously computing this transformation, our system will be able to autonomously determine which lane the vehicle is driving on.

Expected Results

The results of position estimation depend heavily on the precision of the perspective projection. This, in turn, is the product of proper camera calibration. At the moment, I do not know how accurate the perspective projection will be, as I have never used it to estimate camera pose with features that are so distant from the camera. I do know, however, how accurate it needs to be for the system to work (see section 5 on project evaluation).

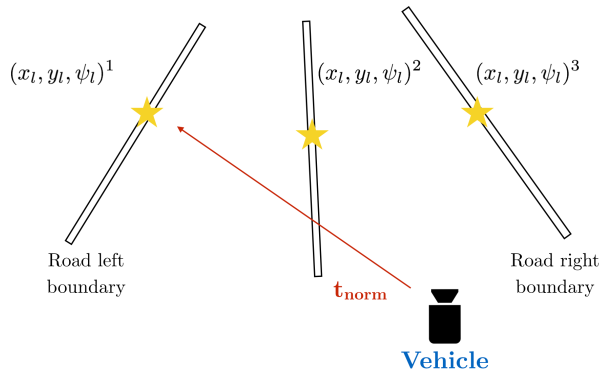


Figure 5: The distance from the vehicle to the left boundary of the road is denoted by t_{norm}.

Step 6: Control Loop

Method

Finally, our system will implement a closed-control loop to keep the car within its lane. Specifically, the system will use a PID controller that it will tune manually to best match the environment.

In this controller, we define the following variables:

- *Process variable:* this is the variable we measure to assess the state of the system. Here, we set it equal to t_{norm}, the norm of the translation from the left boundary of the road to the car.
- *Setpoint (desired outcome):* this is the value of t_{norm} we seek to reach and maintain. This value will be found experimentally, by manually placing the car in the middle of its lane, taking an image of the scene, and computing the corresponding value of t_{norm}.
- *Control variable:* this corresponds to the parameter we tune until the system reaches the desired outcome, the setpoint. In our system, the control variable is the angular velocity ω of the car.

5. Project Evaluation

Evaluation metric: t_{norm}

This variable is the norm of these transformation from the camera reference frame to the left boundary of the road. It is calculated by means of perspective projection, and its accuracy its essential for the performance of our system. Indeed, this value is used to estimate which lane the vehicle is driving on, and the position of the vehicle within its lane. Moreover, the system's control loop depends on this variable to assess the state of the system.

Measuring the metric

This value will be compared against the ground truth, which will be measured manually. For this purpose, we propose the following procedure. (1) First, we place visual marker on the road in front of the vehicle, (2) we move the position of the vehicle until the visual marker is at the same height as the centroid of the leftmost lane (figure 6), (3) and finally, we manually measure the distance from the vehicle to the visual marker. This distance corresponds to t_{norm}.

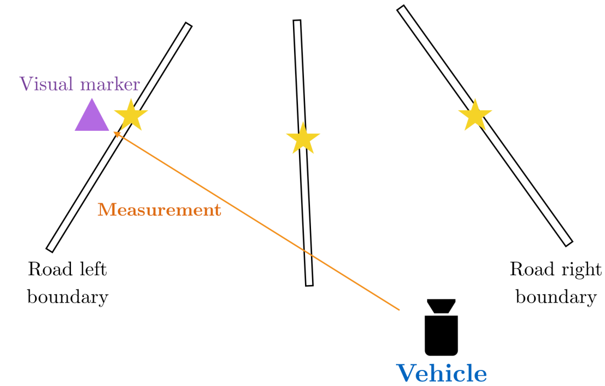


Figure 6: The visual marker is placed, and the position of the car is adjusted so that the marker lies at the centroid of the leftmost lane. Then we measure the distance from the car to the visual marker.

Acceptable performance

Each lane is approximately 1.5 meters wide, and we assume that the RC car is 0.3m wide. Given these values, the vehicle needs to be 0.6m approx. from each side of its lane (figure 7). So, for the vehicle to drift from its lane, it would have to move 0.6m to either side. The variable t_{norm} corresponds to the distance from the left boundary of the road to the vehicle. Its maximum value is 3.0m (1.5m for each lane). Therefore, the margin of error for t_{norm} cannot exceed 0.6/(3.0) = 20%. If it does, the vehicle could drift out of its lane, while according to t_{norm} the car is still driving safely in its lane.

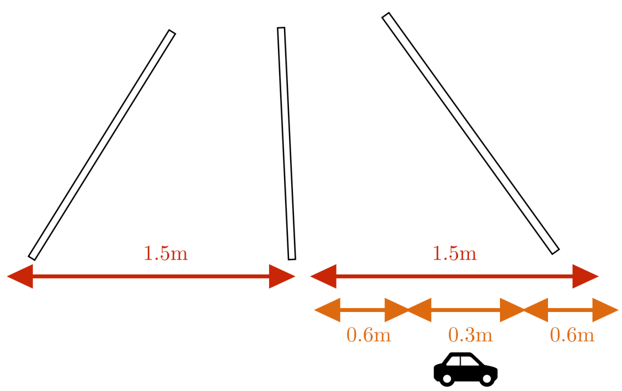


Figure 7: For the vehicle to remain centered in the middle of its lane, it needs to be 0.6m away from either side of the lane. Therefore, the distance t_{norm} needs to have accuracy greater than 0.6/3.0 = 20%.

Trials

My plan is to run three trials. In the first trial, I will test that the vehicle is properly detecting and extracting the features, as well as properly interpreting their meaning. Thus, the trial will steps 1, 2, 3 and 4. In the second trial, I will test the calculation of t_{norm}. Successful calculation of this variable is key to the success of my project, so in reality I will leave time for two trials, in order to ensure the value of t_{norm} is as accurate as necessary. Finally, I plan on running one trial for the entire system, testing whether the car is capable of driving autonomously for the entire length of the pedestrian bridge.

What I expect to learn from any failures

An important skill that I would like to develop is the ability to make plans that I can realistically execute within a proposed timeframe. Failure to complete this project will teach me how to develop plans that can be completed well and on time.

I also expect to learn about the use of perspective projection to estimate camera pose in situations where the features detected are at considerable distances (greater than 2m). Perspective projection is widely used in augmented reality, as it provides the transformations indicating where in the image to overlay the virtual objects so that they appear “real” in the scene. In these cases, however, the objects used to estimate the camera position are much closer to the camera than the centroid of the lanes will be to the vehicle.

Final test

As a final test, our system will be considered successful if the car can drive autonomously within its lane from the beginning to the end of the marked road, which is approximately 0.4 miles.

6. References

[1] “Albert Huang- Lane Estimation for Autonomous Vehicles Using Vision and LIDAR." Albert Huang- Lane Estimation for Autonomous Vehicles Using Vision and LIDAR. N.p., n.d. Web. 27 Mar. 2015. <http://people.csail.mit.edu/albert/pubs/2009-huang-phd-thesis.pdf>.

[2] Seo, Young-Woo. "Tracking and Estimation of Ego-Vehicle’s States for Lateral Localization." (2014): n. pag. The Robotics Institute, CMU. Web. <https://www.ri.cmu.edu/pub_files/2014/5/tech-report-ri-tr-14-08.pdf>.

[3] Huang, Albert. "Lane Estimation for Autonomous Vehicles Using Vision and LIDAR." Web. 24 Mar. 2015. <http://rvsn.csail.mit.edu/Pubs/phd_ashuang_2010feb_laneestimation.pdf>.

[4] Huang, Albert. "Probabilistic Lane Estimation Using Basis Curves." Web. 24 Mar. 2015. <http://www.roboticsproceedings.org/rss06/p04.pdf>.

[5] A Prototype Of Vision Based Driver Assistance System, and By Sebastian Korczak*. Department of Mechanics.PROCEEDINGS OF THE INSTITUTE OF VEHICLES 4(90)/2012 <http://myinventions.pl/035/Sebastian_Korczak-A_prototype_of_vision_based_driver_assistance_system.pdf>

[6] Ioan Bucio, Alexandru Gacsadi, and Cristian Grava. "Vision Based Approaches for Driver Assistance Systems." N.p., n.d. Web. <http://www.wseas.us/e-library/conferences/2010/Iasi/ICAI/ICAI-15.pdf>.

[7] Department of Transportation, Intelligent Transportation Systems. "Connected Vehicle Insights - Trends in Computer Vision." Technology Scan and Assessment Final Final Report. Web. 24 Mar. 2015. <http://www.its.dot.gov/research/pdf/TechScan computervision ITSA.pdf>.

[8] Cutilli, Benjamin. "Computer Vision and Its Application in Self-Driving Cars." Web. 26 Mar. 2015. <http://thesis.haverford.edu/dspace/bitstream/handle/10066/9003/2012CutilliB_thesis.pdf?sequence=1>.

[9] Huang, Albert S., and Seth Teller. "Probabilistic Lane Estimation for Autonomous Driving Using Basis Curves." N.p., n.d. Web. <http://people.csail.mit.edu/albert/pubs/2011-huang-teller-auro-preprint.pdf>.

[10] Apostoloff, Nicholas, and Alexander Zelinsky. "Robust Vision Based Lane Tracking Using Multiple Cues and Particle Filtering." Web. 26 Mar. 2015. <http://www.robots.ox.ac.uk/~nema/publications/Apostoloff03.pdf>.

[11] Mccall, J.c., and M.m. Trivedi. "Video-Based Lane Estimation and Tracking for Driver Assistance: Survey, System, and Evaluation." IEEE Transactions on Intelligent Transportation Systems: 20-37. Print.

[12] Stückler, Jörg, Hannes Schulz, and Sven Behnke. "In-Lane Localization in Road Networks Using Curbs Detected in Omnidirectional Height Images." Web. 26 Mar. 2015. <http://www.ais.uni-bonn.de/papers/ROBOTIK_2008.pdf>.

[13] “Robust Lane Detection in Urban Environments." IEEE Xplore. N.p., n.d. Web. 27 Mar. 2015. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4399388>.

[14] “New Lane Detection Algorithm for Autonomous Vehicles Using Computer Vision." IEEE Xplore. N.p., n.d. Web. 27 Mar. 2015. <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4694332&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D4694332>.

[15] Proceedings Of The Croatian Computer Vision Workshop, Year 1, and September 19 2013 Zagreb Croatia. Computer Vision Systems in Road Vehicles: A Review (n.d.): n. pag. Web. <http://arxiv.org/pdf/1310.0315.pdf>.

[16] “CiteSeerX — Related by Co-Citation: Driver Assistance Systems Based on Vision in and out of Vehicles." CiteSeerX — Related by Co-Citation: Driver Assistance Systems Based on Vision in and out of Vehicles. N.p., n.d. Web. 27 Mar. 2015. < http://people.csail.mit.edu/lukesf/publications/fletcherIVS03.pdf>.

[17] “Miao, X.D., Li, S.M. and Shen, H. (2012) On-Board Lane Detection System for Intelligent Vehicle Based On Monocular Vision. International Journal on Smart Sensing and Intelligent Systems, 5 < http://www.s2is.org/issues/v5/n4/papers/paper13.pdf >.

[18] “Vision Based Lane Keeping Assistance Control Triggered by a Driver Inattention Monitor." IEEE Xplore. N.p., n.d. Web. 27 Mar. 2015. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5642254>.

[19] “CiteSeerX — Citation Query Lane Change Detection and Tracking for a Safe-lane Approach in Real Time Vision Based Navigation Systems." CiteSeerX — Citation Query Lane Change Detection and Tracking for a Safe-lane Approach in Real Time Vision Based Navigation Systems. N.p., n.d. Web. 27 Mar. 2015. <http://airccj.org/CSCP/vol1/csit1231.pdf>.

[20] “Robust Road Marking Extraction in Urban Environments Using Stereo Images." IEEE Xplore. N.p., n.d. Web. 27 Mar. 2015. < http://perso.lcpc.fr/tarel.jean-philippe/publis/jpt-iv10b.pdf >.