Building A Robust, Real-Time Vision-Based Lane Detection Algorithm

Derrick Yu '10

Advisors Alain Kornhauser, ORFE and Bede Liu, ELE

January 14, 2009

Submitted in partial fulfillment
of the requirements for the degree of
Bachelor of Science in Engineering
Department of Electrical Engineering
Princeton University

This paper represents my own work in accordance with University regulations

____

Derrick Yu

The Lane Detection Problem:

The lane detection project aims to improve a previously developed lane detection algorithm to be more robust to varied lighting conditions. The basic premise of the lane detection algorithm is to take an image of a road scene from a camera mounted on a vehicle and process it to detect all the lane markings visible on the road surface. In addition to being robust to lighting changes, the lane detection algorithm must run in real time, processing multiple frames per second. This project was developed as part of ongoing research of autonomous vehicle technologies at Princeton Autonomous Vehicle Engineering (***PAVE***), an undergraduate student organization investigating autonomous robotics. Eventually, the algorithm will be integrated with other systems on *Prospect 12*, ***PAVE***'s current vehicle research platform (Fig. 1). In addition to being applicable to *Prospect 12*, the lane detection algorithm has applications in the automotive industry. Further development of the algorithm may result in a system that is robust enough to be implemented in the near future as a driver aid to identify whether or not a car is staying within a lane and possibly to allow the vehicle to perform emergency maneuvers to prevent it from straying off the road.

Figure 1: *Prospect 12*, an autonomous vehicle used as a research platform by ***PAVE***.

The development of lane detection is motivated by the need to create technologies to increase the safety of automobiles. The National Highway Traffic Safety Administration (NHTSA) reports that in 2007, approximately 41,000 fatalities in the United States resulted from motor vehicle crashes. Of those fatalities, approximately 13,000 were the result of alcohol-related accidents.[1] These fatalities often result from vehicle straying across lanes as a result of erratic driving. Although systems which use sensors or special markers embedded in the road can be implemented to automatically guide specially-equipped vehicles, such a system would require a massive investment in road infrastructure and years to standardize the entire automobile industry. By using vision technology to detect lane markings which are part of pre-existing road infrastructure, it may be possible to produce a commercially viable autonomous vehicle or driver aid in the near future.

Preliminary review of the literature:

To date, there have already many attempts to build robust lane detection algorithms. In 2005 Kuo-Yu Chu presented a method for color segmentation for use in lane detection.[2] The algorithm relies on multiple threshold and histogram analysis to pick out the lane markings. Although the method works well in the scenarios that were tested, the algorithm does not perform well in areas where there are portions of the road covered by tree shadows, where there are areas of the image that may be over or underexposed. In addition, any pixels that are similar in color to the yellow markings are also marked as lane pixels. Another paper describes the use of image-preprocessing

---

1 NHTSA, “2007 Traffic Safety Annual Assessment – Highlights,” 2007, NHTSA’s National Center for Statistics and Analysis/Washington, DC, Aug. 2008 < http://www-nrd.nhtsa.dot.gov/Pubs/811017.PDF>

2 Kuo-Yu Chiu; Sheng-Fuu Lin, "Lane detection using color-based segmentation," Intelligent Vehicles Symposium, 2005. Proceedings. IEEE, vol., no., pp. 706-711, 6-8 June 2005 <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1505186&isnumber=32246>

techniques and uses a set intensity threshold to detect white lanes.[3] The frames are first enhanced to improve the contrast between the lane marking and the road, before a simple intensity threshold is applied to each channel (Red, Green, and Blue) of the color image. Shang-Jeng Tsai presents an interesting algorithm for shadow detection and “removal” for irregular tree shadows.[4] The algorithm scans the image horizontally and separates the pixels into groups of small and large color differences. By comparing the centers of the two clusters as pixels are separated, the algorithm determines the location where the lane boundaries exist. In the case of a shadow, the cluster centers vary from the previous set by a large error. When a shadow is encountered, a separate shadow removal algorithm is run that enhances the shadowed area. Although many of these algorithms have the ability to detect lane pixels, few of them actually represent lanes as a line or a curve. A recently published paper by Aly[5] uses a simple Hough transform to detect lane pixels but introduces novel spline fitting algorithm using RANSAC[6] to detect the lanes.

Some recent research has focused on methods of tracking a lane hypothesis through time. These algorithms usually pair a simple algorithm that is not very robust with a tracker that filters the results of the detection step. McCall proposes a method using “steerable filters” applied to an

---

3 Calderon, J.; Obando, A.; Jaimes, D., "Road Detection Algorithm for an Autonomous UGV based on Monocular Vision," *Electronics, Robotics and Automotive Mechanics Conference*, 2007. CERMA 2007, vol., no., pp.253-259, 25-28 Sept. 2007 <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4367695&isnumber=4367642>

4 Shang-Jeng Tsai; Tsung-Ying Sun, "The robust and fast approach for vision-based shadowy road boundary detection," *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, vol., no., pp. 486-491, 13-15 Sept. 2005 <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1520026&isnumber=32528>

5 Aly, Mohamed, "Real time detection of lane markers in urban streets," *Intelligent Vehicles Symposium, 2008 IEEE*, vol., no., pp.7-12, 4-6 June 2008 <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4621152&isnumber=4621124>

6 Fischler, M. A.; Bolles, R. C., “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,” Comm. of the ACM, Vol. 24, pp 381-395, 1981.

image to quickly filter out lanes from the image.[7] Lines are fitted to the lane pixels and the results are compared with previously detected lanes. The lanes are matched by comparing the variance in the lateral offsets of the lanes. A Kalman filter is applied to the road and vehicle state variables to update the lane estimates. Vacek, et. al. propose a method that uses particle filters to detect and track lane hypotheses.[8]

However, even the latest implementations are designed mostly for use on highways and cannot handle a variety of corner cases that are present in smaller, more local roads. Most current lane detection algorithms function abysmally when presented with roads that have sharp turns, areas where markings are occluded, dappled shadow and rapidly changing or diminished lighting conditions. For such an algorithm to be robust, it also needs to function under situations where lane markings suddenly disappear or shift, which dictates the development of a method to generate a smoothly tracked lane hypothesis based on the current position of the lane. This portion of development focuses on improving the current algorithm such that it can provide a better raw detection rate under such conditions.

Developing the New Algorithm:

The previous implementation of the lane detection algorithm was used as a base from which the new algorithm was to be implemented. The framework of the previous algorithm already provided the capability to save images from the camera, and run the algorithm on the saved

---

[7] McCall, J.C.; Trivedi, M.M., "An integrated, robust approach to lane marking detection and lane tracking," *Intelligent Vehicles Symposium, 2004 IEEE*, vol., no., pp. 533-537, 14-17 June 2004 <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1336440&isnumber=29469>

[8] Vacek, S.; Schimmel, C.; Dillmann, R., "Road-marking analysis for autonomous vehicle guidance," *European Conference on Mobile Robots*, 2007.

frames. Therefore, the use of the original framework simplified the data collection and testing process, greatly reducing development time. Because *Prospect 12* is constantly undergoing modification and upgrades, it would be impossible to conduct thorough live testing of the algorithm. Instead, several data runs were made with the vehicle and the saved images were played back through the framework to test the new algorithm.

For data collection, we used a color monocular camera with a resolution of 640 by 480 pixels mounted to the roof of *Prospect 12*. During the data collection, the images were captured with a frame rate of ~10 Hz so that live test conditions could be approximated in the lab. The vehicle was run between 25-30 miles per hour to simulate the actual target speed of the vehicle under autonomous operation.

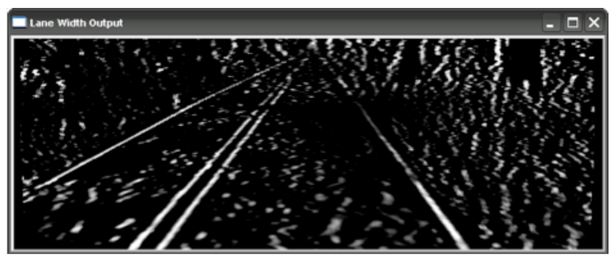
While the algorithm was designed to be robust, several assumptions about the conditions under which it would be run were made:

- The algorithm would always be run during the day. Although some implementations of lane detection algorithms were shown to perform quite well at night, time constraints would not allow enough time to gather the necessary data sets, or tune the algorithm for good performance under these conditions.
- The algorithm would not be run under inclement weather, such as snow or rain. Since the camera needs to be placed high above the vehicle, it needs to be mounted externally. Snow or rain falling on the lens of the camera would distort the image and possibly obscure areas of the image entirely, making detection of lanes almost impossible.

The Lane Detection Algorithm:

The lane detection algorithm consists of three separate filters and one lane-fitting step. Before the images are filtered, they are cropped to a constant size to remove areas of the image above the horizon and areas that are blocked by the hood of the vehicle. This reduces the time it takes for each of the filters to run. A width filter is run on the image to detect areas of the image that correspond to the proper width of a lane. Yellow and white color filters are also run on the original input image. Each color filter is combined separately with the width filter to generate heat maps for yellow and white lanes. Then a RANSAC parabola-fitting algorithm is run 600 iterations for each lane detected in the image. Figure 2 shows a series of images showing the lane detection steps.

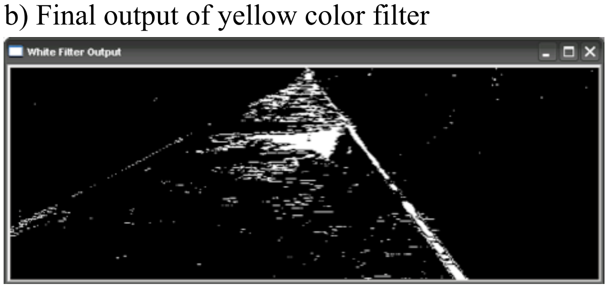
a) Original input image



d) Final output of lane width filter

b) Final output of yellow color filter

e) Detected lanes overlaid on original image



c) Final output of white color filter

Figure 2: Original image with filter results and final fitted lane output.

Yellow Color Filter:

The yellow color filter is designed to filter out all pixels that are not yellow in the image. The filter was devised by transforming the image into the HSV (Hue, Saturation, Value) space and performing various filtering experiments on each channel. The experiments showed that the most prominent feature of the yellow lane was its hue. The steps of the algorithm are as follows:

- Calculate the histogram of the hue channel and find the maximum value within the range of yellow hues (50 – 64). Since the hue of the yellow lane can vary at different times of day, a range of histogram values was chosen. Due to time constraints, these constants have not been verified experimentally. By finding the maximum among these values, we are able to determine approximately the mean yellow hue from the image, assuming a Gaussian distribution of the hues of yellow pixels.
- Threshold image at a constant interval above and below the maximum value. Although the max value is a good indication of the mean of the yellow hue value, the majority of the yellow pixels do not occur at this value. Therefore, a range of yellow values is considered for thresholding. The constant was tuned by subjecting the filter to small data sets.
- Combine with saturation channel and apply a threshold to remove areas of low saturation. The saturation of a pixel is a measure of how intense the color is, ranging from white-yellow to pure yellow in this case. This step removes any pixels that have a yellow hue but may be part of the white lane because they have low saturation values.

Figure 2b illustrates a typical example of the results achieved by the yellow filter. Some pixels are detected that are not part of the lane marking, which is unavoidable since yellow grass and foliage on the side of the road has similar color characteristics to the yellow lane markings.

White Color Filter:

A similar method of experimentation was used to develop the white filter. The white filter is slightly more complex because it was discovered through experimentation that there are different characteristics needed to detect white pixels in the shadow versus white pixels illuminated by sunlight. For this filter, two different images are produced, one based on high saturation values and one on low saturation values, and the result is the sum of the two images.

- A constant saturation threshold is applied to split the saturation channel between areas of low saturation and high saturation. The bright portions of the lane have low saturation values, while the rest of the lane pixels have surprisingly high saturation. Therefore, this filter separates the two types of white lane pixel

For areas of high saturation:

- Combine with hue channel and filter out hues through the green range (180). It was determined experimentally that the white pixels in shadowed areas of the image have surprisingly high hue values, usually in the blue range of the spectrum. Therefore, the hue threshold removes pixels that are most likely not lane pixels in shadow.
- Combine with value channel and threshold at a constant value. Since the image still contains many road pixels that are darker than the white markings, it is necessary to filter them out from the image. Since in some cases, portions of the road may actually be the same brightness or brighter than the lane markings in the darkest shadow, a threshold of 40 (out of values between 0-255) was chosen to balance the amount of lane pixels retained with the amount of road threshold away.

For areas of low saturation:

- Combine with hue channel and filter out all hues in the yellow range (above 60). Since these areas are illuminated by sunlight, these pixels are actually yellow pixels with very low saturation values. Therefore, to eliminate extraneous data, we threshold out everything with hue values greater than yellow.
- Combine with the value channel and subtract yellow filter. Since the range of hues for the white filter is similar to that of the yellow filter, we use the result of the yellow filter to remove the possibility of a yellow lane pixel also being detected as a white lane pixel.
- Calculate the image histogram, find its max value, and threshold the image at that value. At this point only a small portion of the image pixels remain, and the white pixels consist of the majority of the image. The threshold removes any bright road pixels that may remain. The threshold is calculated such that it never exceeds 254. Otherwise, all the bright lane pixels would be eliminated.

Finally, the two separate images are added together to produce the composite white filter image, shown in figure 2c. Although some of the road has been detected as part of the white lane, and some of the shadowed lane pixels to the left of the image were undetected, the majority of the pixels have been filtered out correctly.

Width Filter:

Brendan Collins '08 is the original author of the lane width filter that was developed for *Prospect 12* during the 2007 DARPA Urban Challenge.[9] After an extensive review of the literature and a

[9] More information about the DARPA Urban Challenge can be obtained at http://www.darpa.mil/grandchallenge/index.asp

thorough series of tests with the edge filter, it was determined that the width filter continues to be a good filter to use for the following reasons:

- The filter is quite robust to changes in lighting conditions.
- The filter performed well in extensive tests using gathered data sets.
- In order to reduce the development time for the improved algorithm, the filter was re-used in this implementation.

The width filter works by analyzing the image using a rectangular region of one row in height and a width scaled linearly according to the vertical location of the row in the image. Contrasting areas within the search rectangle that match the lane-width characteristics are marked as lane pixels. In the case of images with dappled shadows such as the original image in figure 2a, you can see that the width filter output, shown in figure 2d, picks out some of the bright patches on the road as possibly part of a lane.

RANSAC Parabola-Fitting Algorithm:

In order to actually detect the lanes, each color filter is independently merged with the lane width filter. The RANSAC fitting algorithm followed by a greedy search are run to find the lanes in the image. Again, this algorithm was developed in the original framework, and was re-used due to time constraints and because experiments verified that the algorithm met many of the design criteria of the new algorithm:

- RANSAC is a very fast algorithm, and running it through multiple iterations allows it to converge on a good fit for the lane.
- The greedy search allows lane pixels that don't exactly fit the parabola to be included in the final detection.

Figure 3 shows the pseudocode for the RANSAC algorithm used to find the parabola fit. Previous experimental results showed that running the algorithm approximately 600 times provides a good trade-off between the accuracy of the detected parabola and the amount of time consumed in running the algorithm.

```
bestParabolaFitness = -1
bestParabola = null
for a set number of iterations
   rows[SIZE] = set of random row values
   cols[SIZE] = set of random column values

   for SIZE points
      find random points to fill row/column arrays

   parabola = a parabola segment fitted to the set of points.

   parabolaFitness = 0;
   for the number of rows spanned by the fitted parabola segment
      parabolaFitness += value of pixels underneath parabola

   if parabolaFitness is greater than bestParabolaFitness
      bestParabola = parabola
      bestParabolaFitness = parabolaFitness

return bestParabola
```

Figure 3: Pseudocode describing the RANSAC parabola-fitting algorithm used to partially detect lanes in the composite filter images.

After the best-fit parabolas are picked out, the ends of each parabola is extended by searching in the rows beyond the end of the parabola fit. Each row is searched near the current end of the lane, and any additional lane pixels that are found are added to the current lane.

After the detection step, the lanes are displayed on an overlay image, shown in Figure 2e. The overlay image is composed of a grayscale version of the original, with the detected yellow lanes overlaid in yellow and the detected white lanes overlaid in green.

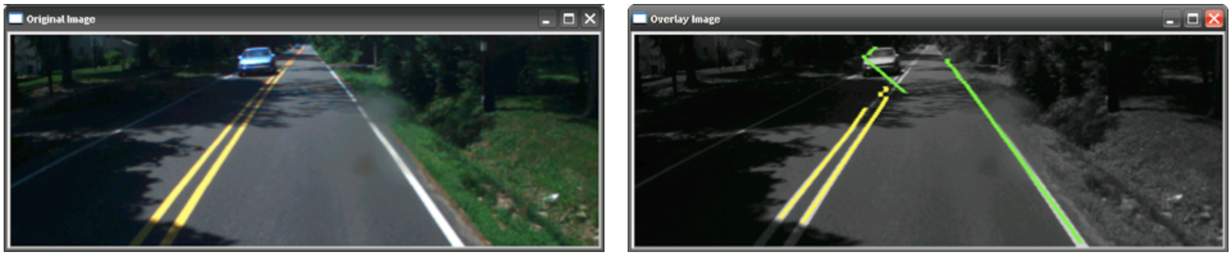
Limitations of the Current Algorithm:

There are very few instances where the algorithm has trouble detecting lanes. One such instance is when the algorithm is presented with a frame where the image is very washed out. As illustrated in figure 4 below, the algorithm only detects a small portion of the yellow lane, and one of the right lanes is detected improperly. One weakness of the algorithm is that it is unable to determine whether an extremely washed out yellow pixel is part of a yellow or white lane. In this case, the algorithm detects a portion of the right yellow line correctly, but the rest of the sections of that lane are misinterpreted as white lane marking.

a) Original input image with overexposed areas.   b) Results of lane detection on image

Figure 4: Scenario where lane detection does not yield good results—in this case the washed out areas of the image are unusable.

One other scenario, which illustrates the limitations of the current algorithm, is illustrated in figure 5. Since the lane detection algorithm has no way of telling whether an object is occluding the lane, it filters it through the color filter as part of the white lane. In addition, the edge detector does really well at detecting the edge of the car as part of the lane. Since the edges of the car are brighter and sharper than those on the shadowed portion of the road, the best fit for the left lane is around the contour of the vehicle.



a) Original input image with vehicle occluding lanes

b) Result of lane detection includes vehicle edges

Figure 5: A second scenario where lane detection yields bad results, in which a car is mistaken for part of the left white lane.

A third limitation can be seen in both of these situations; the algorithm has no way to track lanes in time and estimate their position if the tracking fails. While the lane of travel is clearly detected in the case where the car obscures the road markings, in the first case where the white marking is detected in the wrong position, there is a potential for the algorithms downstream from lane detection to misinterpret that and cause the vehicle to veer into the opposing lane.

Although extensive test data was gathered for the project, due to time constraints, there was not enough time to set up a scheme to run automated tests on all of the data. Therefore, only small-scale data sets were tested with the algorithm. Even with the limited testing done on the algorithm, it is reliable during most daylight hours on roads with a variety of shadow types. However, the algorithm will only run at a maximum of about 10 Hz, or 10 frames per second, meaning that it is not applicable to real-time highway operation.

Conclusions and Future Work:

Through the small-scale tests that were conducted using portions of the sample data, the results in figure 6 were achieved. Although about 1% of the frames were unusable in the sense that lane detection could not detect any lanes, the most important section is the 9% where only a portion of the lanes were detected. This means that the algorithm has a 90% detection rate in its current form, which is still not robust enough for use with an autonomous system—a rate of 98% is necessary to be able to trust the algorithm.

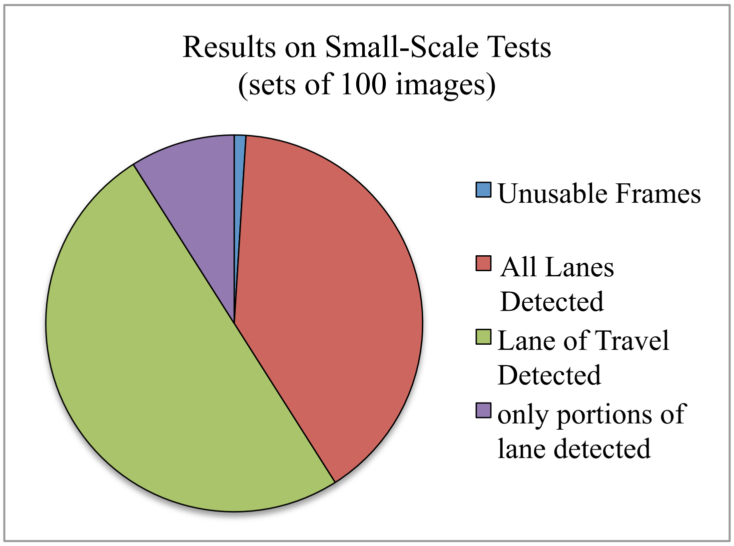


Figure 6: Lane detection statistics from small scale tests

It is important to note that the implementation of the algorithm is quite elegant. The premise of each of the components is very simple, and each has limitations that would not be acceptable if they were used alone to filter lanes, but by combining them, a relatively robust algorithm can be produced. It is also apparent from analyzing the results from detection that a tracking algorithm would greatly improve the performance of the algorithm by being able to estimate the position of the lane in the cases where the lanes are detected incorrectly or incompletely.

A future extension of the lane detection algorithm could involve experimenting with image preprocessing to enhance washed out images before detection, so better results could be achieved. In addition, experimentation with newer camera technologies could greatly improve the quality of the raw images. One camera technology that was researched in the beginning stages of algorithm implementation was PIXIM camera technology[10], which promises to produce images that do not become washed out due to glare or changing light conditions.

---

[10] more information about PIXIM cameras is available at http://www.pixim.com

Finally, probably one of the more important areas of future work would be to add a tracker to the algorithm. This would be made possible by comparing the lateral position of the detected lanes with the previously detected lanes, and rejecting those that deviate by a certain error. In the case that a tracked lane disappears, a Kalman filter can be used to estimate the position of the missing lane(s). The Kalman filter would be updated whenever a good tracked result is recovered, and several state variables such as lateral position, heading, curvature, etc. can be tracked to ensure accurate estimation.

Bibliography:

Aly, Mohamed, "Real time detection of lane markers in urban streets," *Intelligent Vehicles Symposium, 2008 IEEE*, vol., no., pp.7-12, 4-6 June 2008
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4621152&isnumber=4621124

Banggui Zheng, Bingxiang Tian, Jianmin Duan, Dezhi Gao, "Automatic detection technique of preceding lane and vehicle," *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on*, vol., no., pp.1370-1375, 1-3 Sept. 2008
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4636367&isnumber=4636108

Calderon, J.; Obando, A.; Jaimes, D., "Road Detection Algorithm for an Autonomous UGV based on Monocular Vision," *Electronics, Robotics and Automotive Mechanics Conference, 2007. CERMA 2007*, vol., no., pp.253-259, 25-28 Sept. 2007
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4367695&isnumber=4367642

Fischler, M. A.; Bolles, R. C., “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,” Comm. of the ACM, Vol. 24, pp 381-395, 1981.

Kuo-Yu Chiu; Sheng-Fuu Lin, "Lane detection using color-based segmentation," *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, vol., no., pp. 706-711, 6-8 June 2005
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1505186&isnumber=32246

Luo-Wei Tsai; Jun-Wei Hsieh; Chi-Hung Chuang; Kuo-Chin Fan, "Lane detection using directional random walks," *Intelligent Vehicles Symposium, 2008 IEEE*, vol., no., pp.303-306, 4-6 June 2008
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4621271&isnumber=4621124

McCall, J.C.; Trivedi, M.M., "An integrated, robust approach to lane marking detection and lane tracking," *Intelligent Vehicles Symposium, 2004 IEEE*, vol., no., pp. 533-537, 14-17 June 2004
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1336440&isnumber=29469

NHTSA, “2007 Traffic Safety Annual Assessment – Highlights,” 2007, NHTSA’s National Center for Statistics and Analysis/Washington, DC, Aug. 2008
URL: http://www-nrd.nhtsa.dot.gov/Pubs/811017.PDF

Shang-Jeng Tsai, Tsung-Ying Sun, "The robust and fast approach for vision-based shadowy road boundary detection," *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, vol., no., pp. 486-491, 13-15 Sept. 2005
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1520026&isnumber=32528

Vacek, S.; Schimmel, C.; Dillmann, R., “Road-marking analysis for autonomous vehicle guidance,” *European Conference on Mobile Robots, 2007*.

Welch, G.; Bishop G., “An Introduction to the Kalman Filter”
URL: http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf