November, 2014

Visual Lane Detection Algorithm Using the Perspective Transform

Kevin McFall, *Kennesaw State University*
D. Tran

VISUAL LANE DETECTION ALGORITHM USING THE PERSPECTIVE TRANSFORM

Dr. Kevin S. McFall and David J. Tran
Southern Polytechnic State University
Marietta, GA USA

ABSTRACT

This manuscript develops a visual lane detection algorithm using the Hough transform to detect strong lines in a road image as candidates for lane boundaries. The search space in the Hough transform is reduced by searching for lane boundaries where they were detected in the previous video frame. The perspective transform is applied to determine the position and orientation of candidate lines, which are trusted as true boundaries if the detected lane width falls within a specified tolerance of the actual width. Results from a nearly 8-minute long video of highway driving in rain indicate that lane boundaries are correctly identified in 95% of the images. Detection errors occur primarily during lane changes and poor lighting when entering underpasses. Including data from inertial measurements, location on digital maps, and steering direction would help to reduce or eliminate the instances of incorrectly detected lane location.

INTRODUCTION

Within the past decade the development of autonomous automobiles has grown significantly. Semi-autonomous vehicles are becoming more common in the commercial car industry. Audi, for example is reinventing their vehicles with many semi-autonomous features such as adaptive cruise control, side assist, and lane assist [1]. However, fully autonomous vehicles are still not available to the public. One factor of autonomous vehicles that requires refinement and improvement is the road detection system. The most common detection method is through a vision-based system due to the ability to collect data in a nonintrusive manner [2]. Typically a vision-based system uses a camera to capture footage and a sensor or laser to scan the preceding road area. Several methods are used to approach a vision-based road detection system. One concept analyzes road textures, and segments the road image based on road and non-road regions [3]. A system using visual memory stores guiding images to determine a drivable path for current road images [4]. A detection method for urban areas was proposed to identify curbs by using a Markov chain to detect and link curb points [5]. Issues involving these methods include the lack of precision and reliability when functioning on a variety of roads and the inability to identify lane boundaries on multi-lane roads.

Lane detection can be accomplished using a variety of techniques. One system used color extraction to identify lane markings on roads. However size, shape, and motion was also considered in the detection process in order to differentiate lanes and cars of similar color [6]. A fusion-based method of laser scanners and video footage was developed to locate the drivable region and then detect any necessary lanes on roads [7]. A vision-based system incorporated vehicle localization on a digital map to detect and predict lanes [8]. Another approach develops modeled spatial context information shared by lanes, using an algorithm applied learns to address issues with shadows [9].

This manuscript introduces a lane-detection approach using the perspective transform, including an approximate distance from both the left and right lane boundaries. A detection algorithm is applied by searching for lines with the largest Hough transform value within the detection range where boundary lines are expected. The algorithm then determines whether the lines are to be trusted as the true lane boundaries based on their physical distances as computed from the perspective transform. In order to successfully navigate autonomously, only one line boundary need be trusted at any time as the position of other unknown boundary can be accurately approximated.

PERSPECTIVE TRANSFORM

Image generation in cameras follows the central imaging model in Figure 1 [10] where a camera with focal length f is physically placed at the origin of the XYZ axes. Using similar triangles, a point P in space is projected to position p on the xy image plane according to

$$x = f\frac{X}{Z} \text{ and } y = f\frac{Y}{Z} \tag{1}$$

A lane boundary line

$$Z = MX + B \tag{2}$$

with slope M and intercept B in space would be located a distance L to the left of the driver at an angle ϕ as in Figure 2 where

$$M = \cot\phi \text{ and } B = L\csc\phi \tag{3}$$

Assuming all objects to be detected, i.e. lane boundary lines, are located at $Y = H$ where H is the height of the camera above the ground, and substituting Equation (2) into Equation (1), the lane boundary projects to the line

$$y = -\frac{HM}{B}x + \frac{Hf}{B} = mx + b \tag{4}$$

in the image plane. The detection algorithm supplies the image line parameters m and b which combined with Equation (3) simplify to a lane boundary positioned at

$$L = \frac{H}{\sqrt{b^2/f^2 + m^2}} \text{ and } \phi = -\tan^{-1}\frac{b}{mf} \tag{5}$$

In general, a line in the image plane could correspond to an infinite number of lines in real space, but Equation (5) shows that restricting lines to a horizontal plane distance H below the camera achieves a one-to-one correspondence between lines in the image plane and real space.

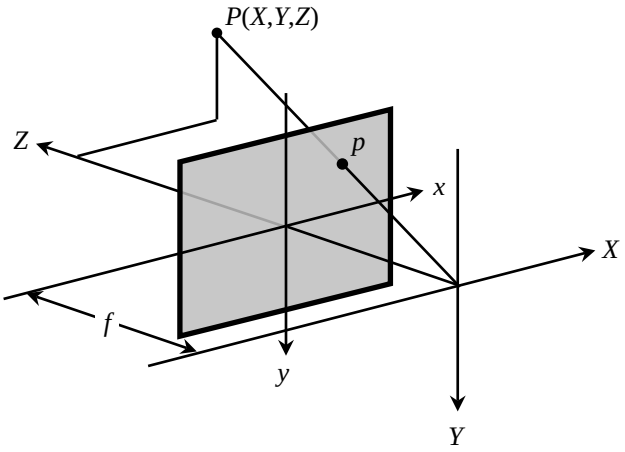


Figure 1: Geometry for projecting a point P in XYZ space to its location p on the xy camera image plane.

The diagonal angle of view ψ is the angle between lines from opposite corners of the image to the XYZ origin. This angle, along with focal length f and image aspect ratio R, define the width of the image as

$$w = \frac{2f\tan\left(\frac{1}{2}\psi\right)}{\sqrt{1+1/R^2}} \tag{6}$$

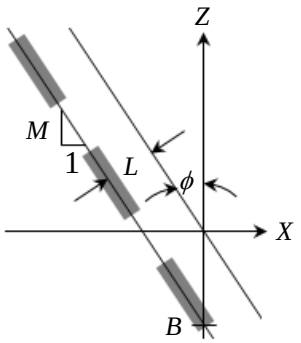


Figure 2: Geometry of a lane boundary line in real space for a car travelling at a distance L from the line directed at an angle ϕ away from line.

Note that b is measured in length units and can be converted to a pixel position with

$$b_{\text{pix}} = \frac{N_c}{w}b \tag{7}$$

where N_c is the horizontal pixel resolution. Combining Equations (5)-(7) rewrite the lane boundary position and orientation as

$$L = \frac{H}{\sqrt{\left[2b_{\text{pix}}\tan\left(\frac{1}{2}\psi\right)/N_c\sqrt{1+1/R^2}\right]^2 + m^2}} \text{ and}$$
$$\phi = -\tan^{-1}\frac{2b_{\text{pix}}\tan\left(\frac{1}{2}\psi\right)}{mN_c\sqrt{1+1/R^2}} \tag{8}$$

Locating the lane line boundary in relation to the driver therefore requires the detection algorithm to supply the equation of the line in the image plane via m and b_{pix}, the camera height H, horizontal image resolution N_c, aspect ratio R, and angle of view ϕ. Note that the angle of view is the only required specification of the camera hardware.

LINE DETECTION USING HOUGH TRANSFORM

The strategy for the line detection algorithm is to first apply a Canny edge filter to create a binary image in which to search for lines. A Hough transform [10] then tallies the number of pixels in the binary image falling on any given line in the image. In order to reduce the complexity of the Hough transform, the top half of the image is cropped out since it falls above the vanishing point for all road boundary lines. Additionally, the left and right lane boundaries will always be located in the left and right sides of the image, respectively. For this reason the bottom left and bottom right quarters of the road image are analyzed separately.

The Hough transform discretizes all possible lines using parameters ρ and θ rather than m and b as illustrated in Figure 4 to avoid discontinuities in the value of slope. The quantity ρ, measured in pixels, represents the perpendicular distance from the line to the top left corner of the image, which is different depending on whether the left or right bottom quarter image is being analyzed as reflected in parts a) and b) of Figure 4. The detection algorithm returns the distance ρ and angle θ of a detected line, which is converted to slope using

$$m = \tan\theta \tag{9}$$

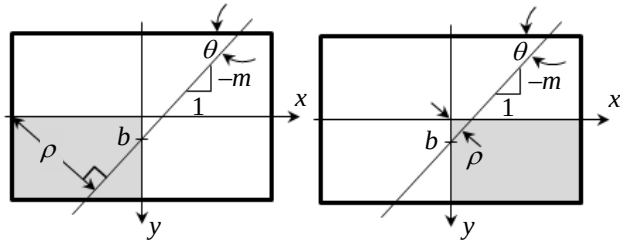
and pixel intercept with

$$b_{\text{pix}} = -\left(\tfrac{1}{2}N_c - \left|\frac{\rho}{\sin\theta}\right|\right)\tan\theta \tag{10}$$

for a line to the left in Figure 4a and

$$b_{\text{pix}} = \frac{\rho}{\cos\theta} \tag{11}$$

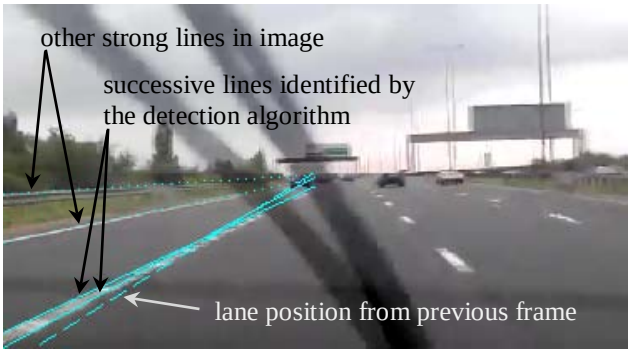
for a line to the right in Figure 4b.



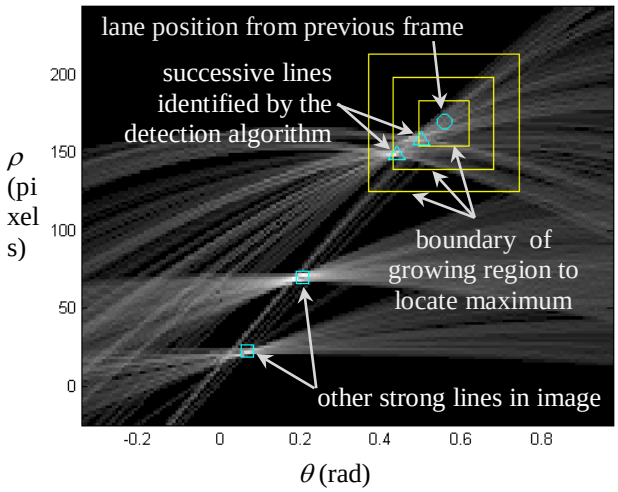
a) b)

Figure 4: Defining a line in the image space with either m and b or ρ and θ for the a) lower left quarter image and the b) lower right quarter image.

Results of the Hough transform appear as an image such as that in Figure 5b where the pixel intensity for a line with any given ρ and θ represents the number of pixels in the binary edge image which fall on the line. The guiding principle for the detection algorithm is to find local maximum in the Hough image in the vicinity of where the lane boundary was identified in the image stream's previous picture frame. The road scene in Figure 5a produces the Hough image in Figure 5b where the dashed line on the road and circle in the Hough image represent where the lane boundary was in the previous frame. A first estimate for the new lane boundary is determined by finding the maximum Hough value within a limited range bounded by the smallest square area surrounding the circle in Figure 5b. This square area is successively enlarged until both the average and maximum Hough value in the bounded area stop increasing. The ρ and θ line with the largest Hough value within the final bounded area is selected as a potential lane boundary line. Notice how the two sides of the lane stripe are identified by the detection algorithm, either of which are acceptable, and other strong lines such as the road edge and guardrail are avoided.



a)



b)

Figure 5: Road image a) and corresponding Hough transform b) used in the algorithm for detecting lane boundaries.

EVALUATING THE ACCURACY OF DETECTED LINES

The line detection algorithm described above may fail for any number of reasons such as poor lighting, occluded camera view, nondescript lane boundary lines, etc. The perspective transformation is used to identify false positives returned by the detection algorithm. The ρ and θ values for the left and right potential lane boundaries are used with Equations (9)-(11) and Equation (8) to determine the distance from and angle each line makes with the camera position. The position of both boundary lines are trusted if the distance between them is within 10% of the known lane width for the road. An example of such a situation appears in Figure 6a where both trusted lane boundaries are marked with solid lines. Also indicated are the

distances from the camera of both detected lines (left and right), and the difference between them (center) which is a reflection of the detected lane width. If the distance between detected lines is not within tolerance, one of the detected lines may still be trusted according to:

- Trust the detected left (or right) line if it was trusted in the previous frame and its distance L has not changed more than 1/6 of a lane width and the angle ϕ has not changed more than 5° from the previous frame.
- Trust the closer of the two detected lines if the distance between them is within 10% of the known double lane width (this covers the case where the detection algorithm correctly identifies a two lane span instead of one)

If only one of the detected lines is trusted, the position of the other one is guessed using the perspective transform to define where the boundary should be at one lane width away from the trusted line. Figure 6b illustrates an example of this situation where the detected lane width was not acceptable and the left lane boundary is trusted but not the right. Notice that the perspective transform correctly identifies the position of the right lane boundary by assuming it is one lane width from the trusted left line. If neither of the detected lines is trusted, it is guessed that the car is driving straight ahead in the center of the lane when searching for lines in the next frame. Detection is considered successful if one or both boundary lines is trusted and all trusted lines do in fact reflect the correct position of the lane boundary.

a) b)

Figure 6: Road images where a) both detected lane boundaries are trusted and b) where one boundary is trusted and the other is approximated.

RESULTS

In order to evaluate the robustness of the detection algorithm, the first viable dashcam video found on YouTube was selected, which happened to be a recording on a British highway. The video was recorded at 30 frames per second (FPS) but it was analyzed at 3 FPS to account for the lower frame rate which would be expected if an embedded system were to perform the analysis in real time in the vehicle. The lane width was assumed to be 12 ft, camera height 3 ft, and camera angle of view to be 50°. Despite uncertainties in these parameters, the algorithm was successful in appropriately detecting a lane boundary 95% of the time. For example, in Figure 7 at least one lane boundary is trusted during lane change as is also the case in Figure 8 while raining. For the nearly 8 minute long video of highway driving, a breakdown of outcomes for the 1400 frames analyzed is:

1) 1330 frames: Successful
2) 9 frames: Both lines trusted, one correctly placed and the other slightly off
3) 6 frames: Only one line trusted, which is significantly misplaced
4) 55 frames: No line trusted

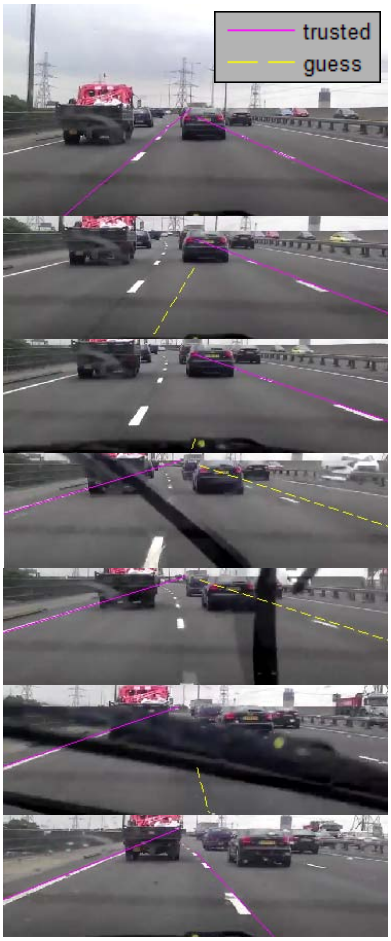


Figure 7: Continued lane boundary detection during lane change.

Obviously outcome 1) is the most common where 95% of the frames successfully located at least one lane boundary line. The 9 frames in outcome 2) are not especially troubling since each of the 9 incidents were momentary glitches where both the previous and following frames were successful. One such instance appears in Figure 9 where the detected right line is slightly out of place. More problematic are the false positives in outcome 3). The windshield wiper in Figure 10 is momentarily mistaken for a boundary line because another strong line in the image just happens to appear two lane widths away from the

wiper blade. Detection quickly recovers after one frame of outcome 4) where no boundaries are trusted, representing a brief 1 s of time between successful boundary detection. While troubling, a 1 s lapse is not as critical when driving straight in the middle of the lane. However, the other 5 frames of outcome 3) all occur during the same lane-changing incident in Figure 11. The incident is set off by reduced contrast from rain causing a loss of detection as a lane change is initiated. First, 3 frames of outcome 4) with no trusted line cause the algorithm to search for lane boundaries in the middle of the lane where none actually exists. A road streak is then mistaken for an actual line. After 5 frames of trusting a streak that is not an actual boundary, detection is lost for 7 more frames of outcome 4) before successful detection resumes. This one incident alone is responsible for 15 of the 70 unsuccessful frames. The primary reason failure continues for so long, over 5 s, is that after failing to detect a line, the algorithm searches by assuming the car is in the middle of the lane which is obviously not the case

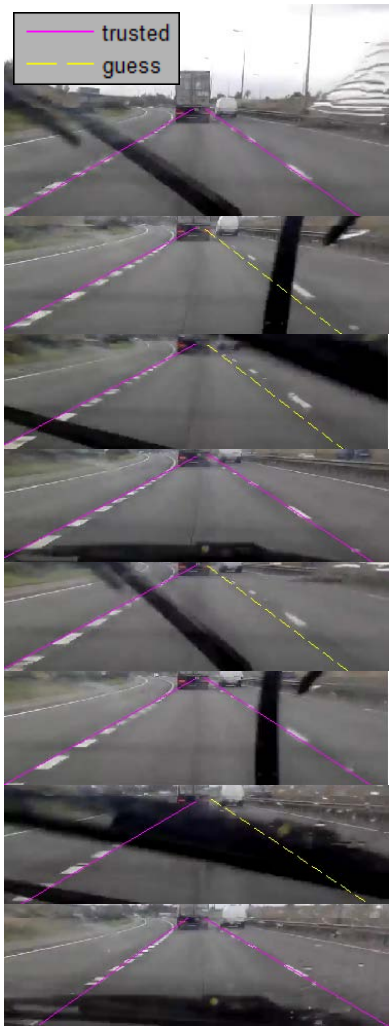


Figure 8: Successful lane boundary detection while raining.

Figure 9: Example of outcome 2) where both detected boundary lines are trusted but one (the left) is correct while the other (the right) is slightly off.

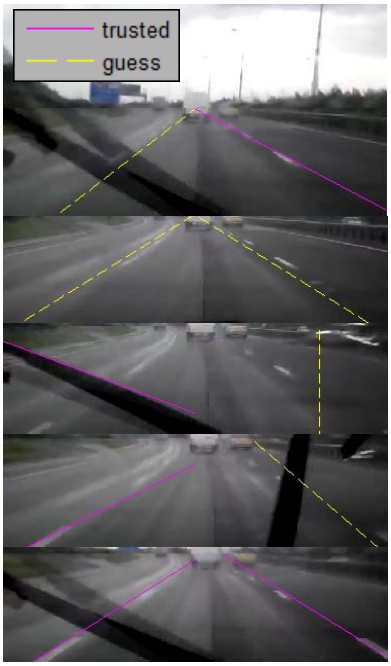


Figure 10: Example of false positive outcome 3) where a detected line is believed but significantly off.

when changing lanes. In an actual autonomous application, other data such as steering direction could help inform the algorithm where to look for lane boundaries.

Of the 44 outcome 4) frames not appearing in Figures 10 or 11, 7 involve a single frame before recovering, three cases (6 frames) span 2 unsuccessful frames in a row, one case spans 3 frames, and seven cases span 4 frame. One of the situations spanning 4 frames appears in Figure 12 where the momentary darkness of passing through an underpass hinders detection.

SUMMARY

In total, detection failed continuously in only 8 instances of more than 1 s during the almost 8 minute period explored. An actual autonomous vehicle would obviously require reducing such instances to essentially none. Rules for when to trust the detected lines can be expanded, and selection of design parameters fine-tuned, although care should be taken to avoid micromanaging to deal with specific situations in any given set of images. For reference, the number of outcome 4) failures is 80, 55, 42, and 35 for lane width tolerances of 5, 10, 15, and 20% of the known width, respectively. Note that reducing outcome 4) failures from 55 to 35 by increasing the tolerance results in approximately a doubling of both outcome 3) and 4) failures. Since the optimal set of design parameters will be specific to each set of road images, it is recommended to resist adjusting parameters, while using a simple set of rules for detecting boundary lines in order to succeed over a wide variety of different road conditions. However, developing a more sophisticated local maxima search for analyzing the Hough transform image is likely to result in improved performance for all road conditions. Perhaps the most promising improvement involves fusing image data with other sources of information such as speed/direction/location from an inertial measurement unit, steering direction, or road curvature and lane configuration by locating the vehicle on digital maps using GPS.

REFERENCES

[1] Audi Driver Assistance Systems. *Audi USA News*, [online] January 10, 2012, http://www.audiusanews.com/pressrelease/2757//driver-assistance-systems (Accessed: August 2014)

[2] Agunbiade, O. Y., Zuva, T., Johnson, A. O., and Zuva, K.., 2013, "Enhancement Performance of Road Recognition system of Autonomous Robots in Shadow Scenario," Signal & Image Processing : An International Journal., 4(6), pp. 1-12.

[3] Graovac, S., and Goma, A., 2012, "Detection of Road Image Borders Based on Texture Classification," International Journal of Advanced Robotic Systems., 9, pp. 242-254.

[4] Courbon, J., Mezouar, Y., and Martinet, P., 2009, "Autonomous Navigation of Vehicles from a Visual Memory Using a Generic Camera Model," IEEE Transactions on Intelligent Transportation Systems., 10(3), pp. 392-402.

[5] Tan, J., Li, J., An, X., and He, H., 2014, "Robust Curb Detection with Fusion of 3D-lidar and Camera Data," Sensors., 14, pp. 9046-9073.

[6] Cheng, H., Jeng, B., Tseng, P., and Fan, K., 2006, "Lane Detection with Moving Vehicles in the Traffic Scenes," IEEE Transactions on Intelligent Transportation Systems., 7(4), pp. 571-581.

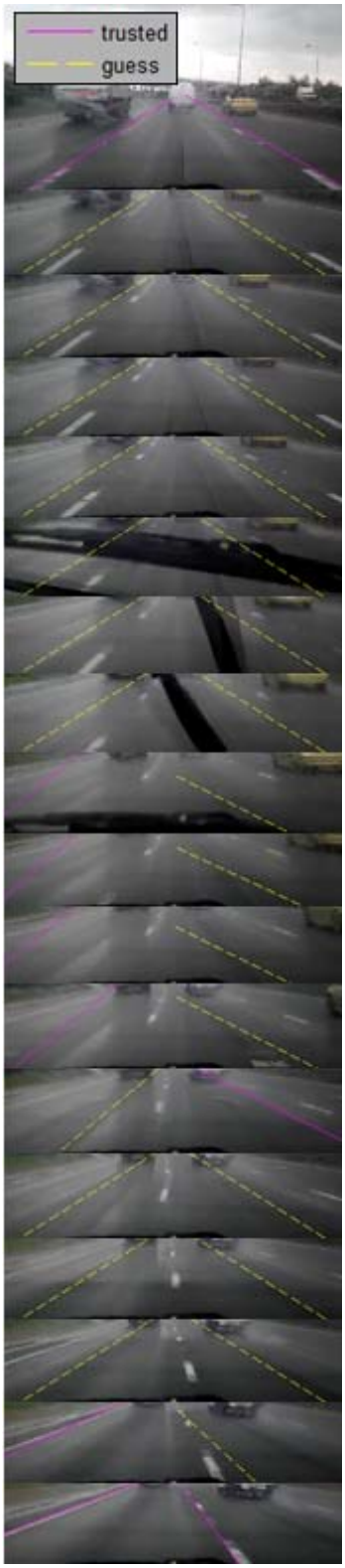


Figure 11: The other 5 frames of outcome 3) occurring during a lane change, where the preceding and following frames fail to detect boundary lines as in outcome 4).

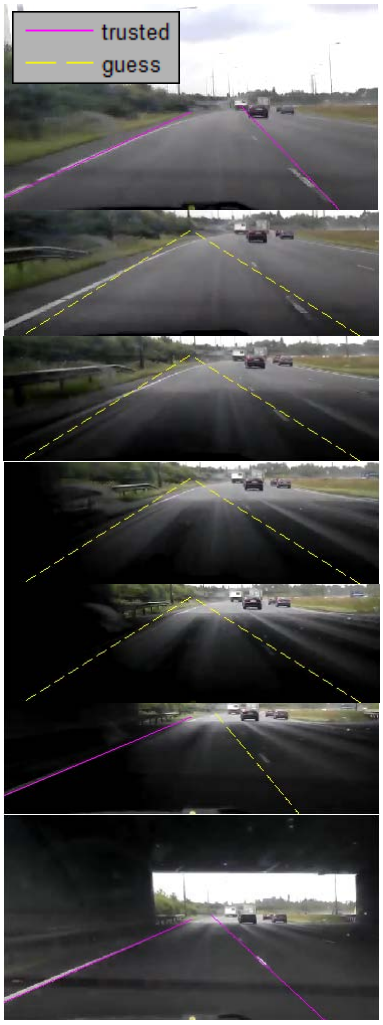


Figure 12: Changing light conditions entering an underpass and the resulting loss of lane boundary detection.

[7] Li, Q., Chen, L., Li, M., Shaw, S., and Nuchter, A., 2014, "A Sensor-Fusion Drivable-Region and Lane Detection System for Autonomous Vehicle Navigation in Challenging Road Scenarios," IEEE Transactions on Vehicular Technology., 63(2), pp. 540-555.
[8] Wang, C., Hu, Z., and Chapuis, R., 2012, "Predictive Lane Detection by Interaction with Digital Road Map," Journal of Information Processing., 20(1), pp. 287-296.
[9] Gopalan, R., Hong, T., Shneier, M., and Chellappa, R., 2012, "A Learning Approach Towards Detection and Tracking of Lane Markings," IEEE Transactions on Intelligent Transportation Systems, 13(3), pp. 1088-1098.
[10] Corke, P., 2011, Robotics, Vision, and Control, Springer, Berlin.