On-Road Vehicle and Lane Detection

Chi-Shuen Lee
Department of Electrical Engineering
Stanford University
Stanford, CA
chishuen@stanford.edu

Yu-Po Wong
Department of Applied Physics
Stanford University
Stanford, CA
mkenwong@stanford.edu

Xuerong Xiao
Department of Electrical Engineering
Stanford University
Stanford, CA
xuerong@stanford.edu

***Abstract*— We implement lane detection using edge detection, Hough transforms, and vanishing point filtering in Hough space; the car detection is implemented by using histogram of oriented gradients feature descriptors and classified by linear support vector machines. Hard-negative mining is applied to alleviate detection of false positives; with the information of vanishing point along with prior knowledge such as the width of the lanes, we reconstruct the 3D ground plane and estimate the distance from the camera to the cars in the front from monocular vision.**

Keywords—Car detection, Lane detection, Hough transform, HOG, SVM, hard negative mining.

I. Introduction

On-road vehicle and lane detection is critical for the safety of a self-driving automobile system. When a vehicle changes lane, the location of the lanes, the vehicles on the lanes, and the distance from itself to other vehicles need to be accurately measured. Many algorithms for vehicle and lane detection have been proposed and will be briefly reviewed in Section II. Accurate distance measurement often rely on active detection systems such as Radar or Ladar. The distance estimation method proposed and implemented in this report is meant to give only a rough estimate of the distance from the object to the camera from monocular vision and is not the focus of this work. Therefore, we will not review the prior works on distance measurement in detail.

II. Prior Works

A. Lane Detection

King Hann Lim et al. [1] used the bottom region in an image to statistically find the pixel color range of the road surface to generate a map locating the lane region, performed an edge detection using Sobel filter, and then weighted-gradient Hough Transform was employed to identify the lane markings. Yue Wang et al. used Canny/Hough Estimation of Vanishing points and B-spline to fit the lane in the images. [2] And Qiang Chen et al. used hyperbola fitting for a real time lane detection system [3].

B. Car Detection

Many vision-based techniques have been developed to detect vehicles in various road scenes. A good review of vehicle detection has been described in [4], including: Tzomakas and Seelen [5] detected vehicles based on the shadows underneath them; Khammari et al. [6] applied a horizontal Sobel filter on the 3rd level of the Gaussian pyramid to obtain local gradient maxima where a vehicle candidate is located. Then a bounding box was extracted by verifying the horizontal symmetry; Claudio Caraffi et al. [7] used a WaldBoost [8] trained sequential classifier applied within a sliding window framework, which is an AdaBoost-based algorithm automatically builds a fine-grained detection cascade of the Viola and Jones type; motion-based methods such as optical flow are also commonly used for vehicle detection [9].

III. Algorithms

A. Lane Detection

For lane detection part of this project, we designed and implemented our own algorithm from scratch based on the knowledge we learned from the class. The algorithm we developed is based on two main operations: edge detection and Hough transform, where the 1D Prewitt gradient filter in horizontal direction is used in edge detection.

The algorithm is designed to detect straight lines in the image. Furthermore, based on the knowledge in 3D reconstruction, if we project parallel straight lines on a plane in 3D space to a 2D image with a view angle not perpendicular to the plane, the parallel straight lines will be projected into straight lines passing through one point on the 2D image, which is called vanishing point. An example of vanishing point is shown in Fig. 1. In our algorithm we extract vanishing point and use its position to detect lanes from many false positive in Hough transform.

In the first few frames of the algorithm, it is working on initializing the system by finding the vanishing point. In these few frames, the filter used is a hard filter just removing lines close to horizontal direction. Images from each step are shown in Fig. 2.

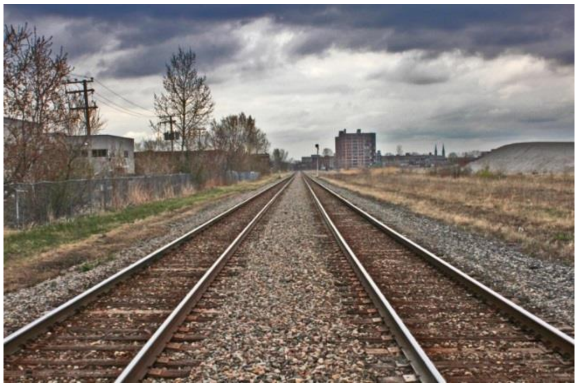

Fig. 1 Example of vanishing point. Here the four parallel lines from a pair of train track is imaged as four lines passing through the vanishing point (Source: http://www.vertice.ca/index.php/2012/sonic-vanishing-points/)

The algorithm in initialization steps check the variance of the intersections of the detected lines (more than 2 lines). The vanishing point is being set as the mean of the intersections once the variance of the intersection is smaller than 50 pixels.

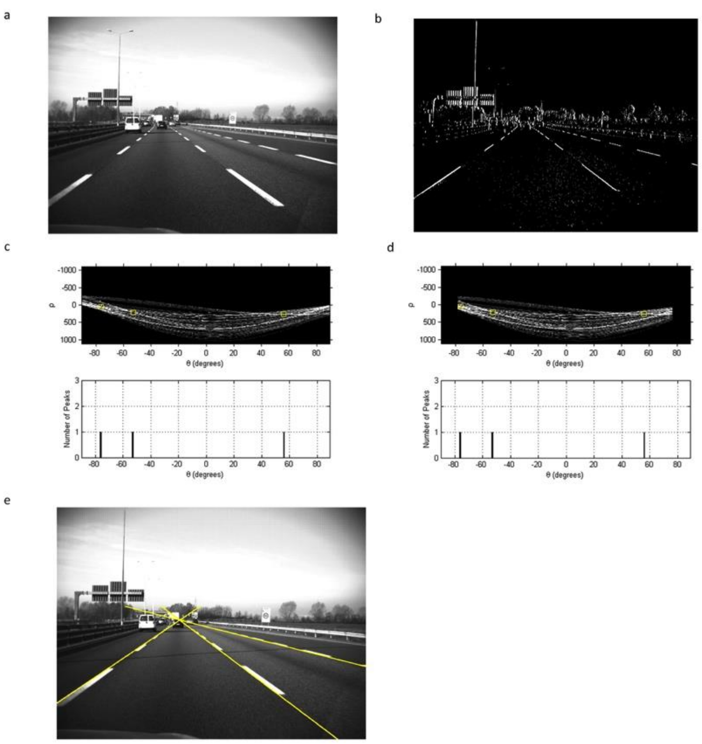

Fig 2. Image from each step a) Original image from one frame, b) Edge detection with Prewitt gradient filter in horizontal direction, c) Hough transform of detected edges, d) Hough space after removing horizontal lines by hard filter, e) result.

After knowing the vanishing point, we used vanishing point filter in the Hough space, which removes all the line that not pass through a 20-pixel circle around the vanishing point. The filter Hough space after applying vanishing point filter is shown in Figure 3.

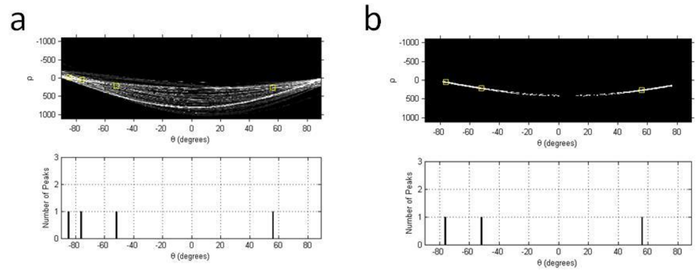

Figure 3. Vanishing point filtering in Hough space. a) before filtering, b) after filtering.

Furthermore, lane tracking is applied to remove noise that only appears less than 5 frames and the tracked lane is labeled as detected if it's missing less than 5 frames. The lane tracking code is modified from MATLAB example [10].

B. *Car Detection*

Car detection is achieved using histogram of oriented gradients (HOG) descriptor in conjunction with a linear support vector machine. Both positive and negative data are needed for training. Hard-negative mining (HNM) explicitly includes false positives into negative training data, based on probabilities. The false positives are retrained after evaluation, and the process of HNM is repeated several times. The model is then applied to test images. Information from lane detection is used to remove false detections at regions outside lanes. The following diagram in Fig. 4 summarizes the algorithm [11].

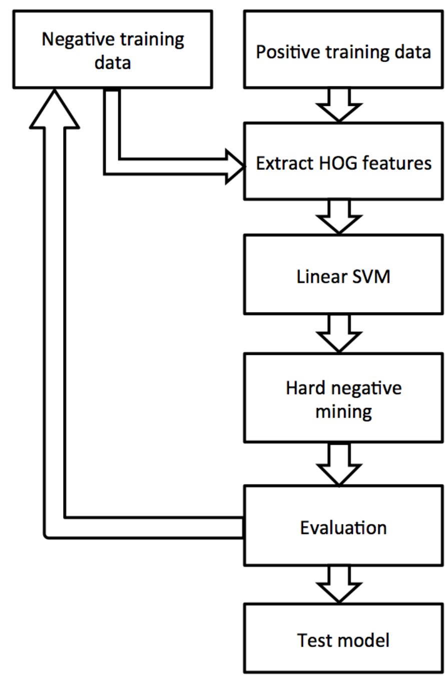

Fig. 4. Process flow of training and testing the detection of cars

The positive training data are from [12], and they are mostly rear views of cars that occupy a substantial portion of the image. There are also front views of cars included. All positive data images contain information of background scenes that later proved to degrade the performance of the detection. The initial negative training data are from [13]. The negative data are mostly street scenes that do not contain cars. They are dynamically increased after each iteration of the HNM process. Some examples of positive and negative data are shown in Fig. 5 below.

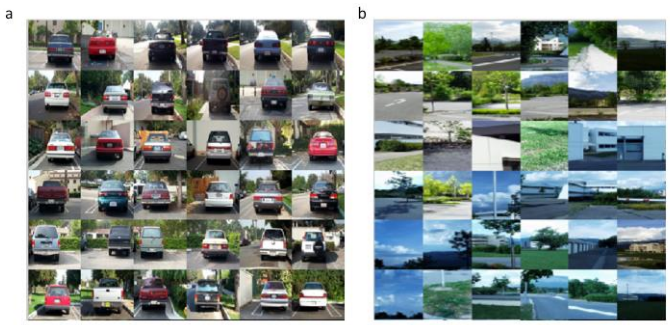

Fig. 5. Examples of (a) positive and (b) negative training data ([12, 13])

HOG descriptors of both positive and negative samples are then extracted. The algorithm of HOG feature extraction is as follows [14]. A detection window is scanned across the image at different scales. Each window is divided into smaller cells. For each cell, a local histogram of gradient directions is computed over the pixels. The cells are then grouped into blocks for contrast normalization, which would improve robustness against illumination. The normalized blocks are then vectorized and referred to as the HOG descriptors. We implemented HOG descriptor using VLFEAT vl_hog function with 8×8 cell size. An example image and its HOG features are shown in Fig. 6 below.

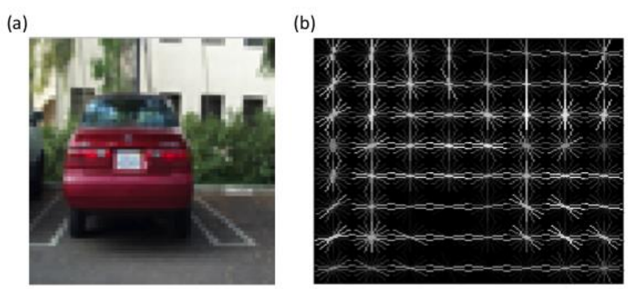

Fig. 6. An example resized training image (a) and its HOG features (b)

The HOG descriptors are then fed to a linear SVM for classification. If the classifier incorrectly classifies a non-car object as a car in a sliding window, the feature vector of that false-positive patch is recorded. This method is called hard-negative mining [11]. The false positives are evaluated according to their probabilities and are added to the negative training data to go through the training process again. We found that a single iteration of HNM is usually not enough to generate satisfying results. The car detection results shown in this report use 5 iterations.

After training, the classifier is applied to test images. When an objected is detected to be a car with a probability above a threshold, a bounding box is drawn. Several bounding boxes may appear at nearby locations of a detected object. We use non-maximum suppression to eliminate the redundant detector responses. Only a few boxes with top scores are kept considering multiple occurrences of cars in an image. A test image with detected cars is shown in Fig. 7.

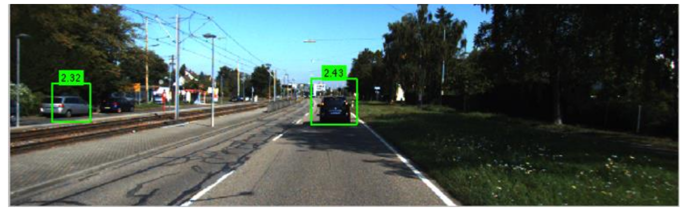

Fig. 7. Detected cars with bounding boxes and scores in a test image

C. Distance Estimation by 2D Projection

In this section, an algorithm mapping 2D image from camera to 2D plane on the ground is developed. Real 3D reconstruction requires stereo-camera. Here we estimate the distance from a 2D monocular vision with an assumption that the camera matrix (K) has been calibrated beforehand. The algorithm is illustrated in Fig. 8 and the procedure goes as follows: (i) From the vanishing point (v) calculated in the lane detection, the direction of the lanes in 3D space can be derived from $\overrightarrow{d_{3D}} = K^{-1}v / \left\|K^{-1}v\right\|$. (ii) Mark 2 points p1 and p2 on each of the lane boundaries, then calculate the two corresponding lines in 3D space ℓ_1 and ℓ_2. (iii) By imposing assumptions that the distance from the camera to the ground plane (Zcam = 1.6m) and the width of the lane (Dlane = 3.6m), we can calculate P1 and P2 in 3D space. (iv) With P1 (or P2) and $\overrightarrow{d_{3D}}$, the ground plane can be reconstructed and any point in the 2D image on the ground can be used to estimate the distance in the 3D space. From Fig. 9, the lanes are quite straight and parallel to each other in the ground plane projection. This shows that by knowing the camera matrix and direction of ground plane, a single view 2D image to ground plane projection is possible. After constructing this projection, the bottom lines of the bounding boxes of cars are projected to the ground plane and distance information can be extracted from there.

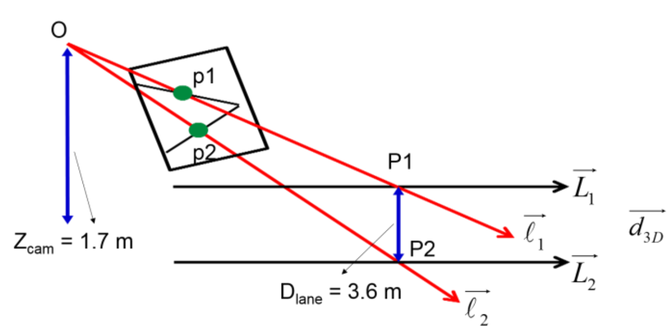

Fig. 8. Illustration of the algorithm and assumptions of estimating the distance from a detected object to the camera.

a

b

Fig. 9. Projection of 2D image to ground plane: a) original image, b) projection to ground plane (vertical axis not to scale for illustration).

IV. Results

A. Training performance:

The precision and recall curve shown in Fig. 10 quantitatively illustrates the performance of the hard-negative mining process. Threshold used in PASCAL VOC [15] is set to be 0.1. Precision is defined as TP/(TP+FP); recall is defined as TP/(TP+FN), and accuracy TP/(TP+FP+FN), where TP is true positive, FP is false positive, and FN is false negative.

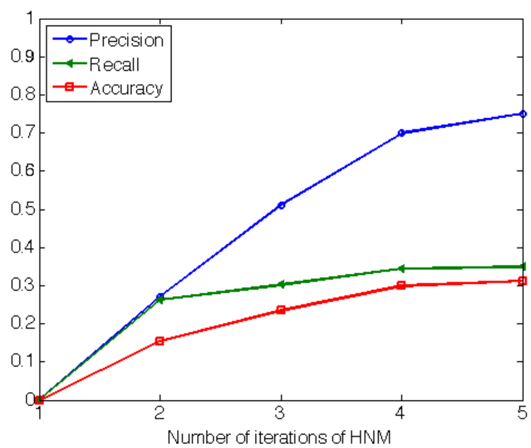

Fig. 10. Precision, recall, and accuracy as a function of number of iterations of HNM in the training process

As the number of HNM process increases, all three parameters increases, meaning the HNM contributes constructively to the training process. The performance shows the trend of saturating when the number of iterations reaches 5, which is the parameter that we used in this project.

B. Good results:

Qualitative analysis of our algorithm running on several datasets is shown below with screenshots of the demo video.

Good results are shown in Fig. 11. We can see there that lane and car detections both generate satisfying results. The yellow lines match the lanes, and the bounding boxes enclose the detected cars. The distance estimation is not very accurate due to the variation of size of bounding box between frames, but we can get a rough idea of the distance of detected car from our car. Besides, the algorithm generally underestimates the distance, which is a safe result for self-driving automobile applications.

a

b

c

d

Fig. 11. Four screenshots of well-detected objects using dataset from KITTI [16]

C. Bad results:

In Fig. 12, two bad frames of detection are shown. In frame a, the algorithm is detecting a false positive on the divider right next to the outer lane. This false positive cannot be removed by the vanishing point filter, because the divider is parallel to the lane and the height of it is close to the ground, which makes the line to be too close to the vanishing point. The front view of the incoming car is not detected (false negative) due to the fact that there are more rear views in the positive training data than front views.

In frame b, there's a false negative in lane detection. The false negative is from the suppression neighborhood used in the algorithm searching peaks in Hough spaces. We set constant spacing for the suppression neighborhood, causing one of the lanes being neglected in frame b. The algorithm can be improved by considering the Hough space spacing from a 3D construction point of view. There are also overlapping bounding boxs in frame b in car detection, resulting from an inappropriate threshold set in the non-maximum suppression and can be remedied easily.

Fig 12. Two screenshots of our algorithm running poorly on dataset from KITTI for lane detection.

Fig. 13 shows the distance estimation error compared to one of the video, with the ground truth measured by Velodyne HDL-64E Laserscanner. While the mean error is about -40 %, the correlation between the estimation and the ground truth is reasonably linear.

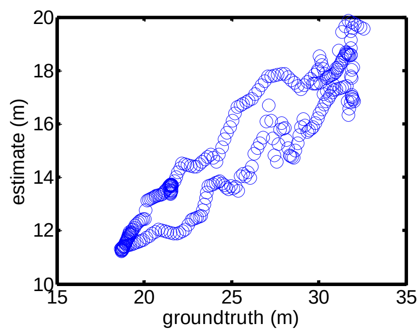

Fig. 13. Distance estimated from 2D monocular-vision image vs. the ground truth.

V. Discussions

In this section we discuss the difficulties faced in our detectors and propose potential techniques to resolve these issues.

- The current lane detector does not consider the detection of curved lanes. Here we briefly discuss one potential method to do so. Assuming the curvature of the curved lanes are small, then the lane markings close to the bottom of the image (i.e. close to the camera) will appear nearly straight in the image. Therefore, we can still detect the near lanes and calculate the vanishing point. With the information of the near lanes color and location, we can gradually move the window upward and detect the lane marking within the local window, which should have similar color and smooth transition to those appear close to the bottom. Occlusion and lighting effect may affect the detection performance so some sophisticated tricks will be needed.

- The current lane detector still suffers from occlusion and variance of lighting effect. Although pieces of the lane marking can be detected as the peaks in the Hough space even with occlusion, there are usually several line-like features in the image which may have signals stronger than the real lane boundaries resulting in false detections. To improve the lane detector, we can utilize the knowledge of some lane markings detected in the previous frames that consistently show strong signals. For example, the two lane boundaries right beside the car are in general easy to detect and have stronger signals across the frames. With the information of the lane marking color, we can raise the weight of the pixels that have similar color; in addition, with the vanishing point and the prior knowledge of the approximate width of the lanes, we can predict where the next lane boundaries should be and narrow down the searching window.

- There is a big room to improve our car detector. One major problem is the difficulty of dealing with occlusion: when part of a car is blocked in the image, e.g. by another car, the detector usually fails. This problem can be mitigated by using object tracking as proposed in [17]. In a sequence of frames, there is a good chance to detect a car without being occluded; once the car is detected, the program needs to keep tracking this car so that it can be detected even with occlusion. Another benefit of tracking the cars is for speed-up because the searching does not need to cover the entire image.

Another simple trick that should improve the performance is to find a better training data set. The data currently used for training is not "clean" enough, i.e. there are many other unnecessary information in the image besides the cars. We extract the ground truth bounding boxes form 5 different data sets labeled as the "Clean" data, as illustrated in Fig. 14, to train the SVM model and test the model on another testing set different from the 5 data sets. In Fig. 15 we plot different detection performance metrics versus the matching threshold (i.e. the bounding box overlap with the ground truth should > threshold in order to be claimed as a correct detection). As a reference, a threshold of 0.5 is used in the PASCAL VOC [15] as the detection criterion. Clearly the "Clean" data greatly improves the performance, especially the recall, and the detected bounding box will also be tighter, which in turn improves the distance estimation.

Fig. 14. Upper row: “Clean” training data with tight bounding box; lower row: “noisy” training data with loose bounding box.

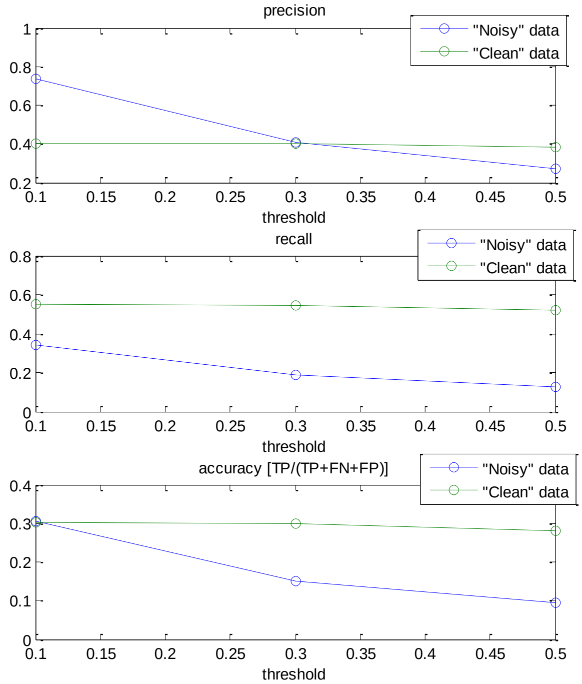

Fig. 15. Detection performance metrics versus the matching threshold (i.e. the bounding box overlap with the ground truth should > threshold in order to be claimed as a correct detection).

In addition, our current detector does not take the advantage of the RGB information, which should be useful because a car usually has a uniform color.

- For all kinds of detectors, parameter optimization is of great importance because various scenarios need to be considered. For example, in the lane detector, having a higher threshold in the selection of the Hough peaks can be helpful when there are many weak lines (e.g. utility pole) in the frames; while in a clean frame but with lower resolution or darker lighting, a lower threshold is desirable. One major drawback of this work is that we do not establish a thorough evaluation platform to enable efficient parameter optimization, which is something we should implement in the future.

References

[1] K. Lim, K. Seng; L. Ang, and S. Chin, "Lane Detection and Kalman-Based Linear-Parabolic Lane Tracking," Intelligent Human-Machine Systems and Cybernetics, 2009. IHMSC '09. International Conference on , vol.2, no., pp.351,354, 26-27 Aug. 2009

[2] Y. Wang, E. Teoh, and D. Shen, “Lane detection and tracking using B-Snake,” Image and Vision Computing, Volume 22, Issue 4, 1 April 2004, Pages 269-280

[3] Q. Chen and H. Wang, "A Real-time Lane Detection Algorithm Based on a Hyperbola-Pair Model," Intelligent Vehicles Symposium, 2006 IEEE , vol., no., pp.510,515, 0-0 0

[4] S, Sivaraman and M. Trivedi, "A review of recent developments in vision-based vehicle detection," Intelligent Vehicles Symposium (IV), 2013 IEEE , vol., no., pp.310,315, 23-26 June 2013

[5] C. Tzomakas and W. Seelen, "Vehicle detection in traffic scenes using shadows," Technical Report 98-06, Institut Fur Neuroinformatik, Ruht-Universitat, Bochum, Germany, 1998

[6] A. Khammari, F. Nashashibi, Y. Abramson, and C. Laurgeau, “Vehicle detection combining gradient analysis and AdaBoost classification,” in Proc. IEEE Intell. Transp. Syst., Sep. 2005, pp. 66–71.

[7] C. Caraffi, T. Vojii, J. Trefny, J. Sochman, and J. Matas, “A system for real-time detection and tracking of vehicles from a single car-mounted camera,” in Proc. 15th Int. IEEE Conf. ITSC, 2012, pp. 975–982.

[8] J. Sochman and J. Matas, "WaldBoost - Learning for Time Constrained Sequential Detection," in CVPR, vol. 2, 2005, pp. 150-157.

[9] J. Arrospide, L. Salgado, M. Nieto, and F. Jaureguizar, “On-board robust vehicle detection and tracking using adaptive quality evaluation,” in Proc. 15th IEEE ICIP, Oct. 2008, pp. 2008–2011.

[10] http://www.mathworks.com/help/vision/examples/lane-departure-warning-system.html

[11] A. Vedaldi and A. Zisserman, “Object category detection practical,” http://www.robots.ox.ac.uk/~vgg/practicals/category-detection/index.html#step-33-evaluation-on-multiple-images

[12] http://www.vision.caltech.edu/html-files/archive.html

[13] http://lear.inrialpes.fr

[14] N. Dalal and B. Triggs, “Histogram of oriented gradients for human detection,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2005.

[15] http://www.computervisiononline.com/dataset/pascal-voc-datasets

[16] J. Fritsch, T. Kuehnl and A. Geiger, “A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms,” International Conference on Intelligent Transportation Systems (ITSC), 2013.

[17] T. Voj´ıˇr and J. Matas, “Robustifying the Flock of Trackers,” in Computer Vision Winter Workshop, 2011.

Appendix 1: Contributions:

Algorithm design and evaluation: All three members

Lane Detection: Yu-Po Wong

Car Detection: Xuerong Xiao

Distance estimation and integration: Chi-Shuen Lee