

# ROBUST LANE DETECTION AND TRACKING WITH RANSAC AND KALMAN FILTER

*Amol Borkar, Monson H. Hayes, Mark T. Smith*

Address

## ABSTRACT

Abstract goes here

**Index Terms**— One, two, three, four, five

## 1. INTRODUCTION

Intro

## 2. PRIOR RESEARCH

Numerous techniques for vision-based lane detection have been developed in an attempt to robustly detect lanes. In the extraction of features for lane detection, one of the most commonly used approaches is to apply an edge detector to the data [1, 2]. With this approach, a Canny edge detector is typically used to generate a binary edge map. From the binary edge map, the classical Hough transform is then used to extract a set of lines as candidates for the lane markers. While this approach shows good results in general, the detected lanes are often skewed due to surface irregularities or navigational text markers on the road. Color segmentation to extract lane markers is another approach that is often used [3, 4]. Unfortunately, color segmentation is sensitive to ambient light and requires additional processing to avoid undesirable effects.

The majority of the approaches used for lane detection operate directly on the images that are captured by the camera without any geometrical correction or change in camera perspective [5, 1, 3, 6]. Although dealing with images from the camera perspective allows access to raw data values, defining the properties of the features of interest may be complicated. For example, a forward-looking camera will capture images that have lane markers that are not parallel and have line widths that vary as a function of the distance from the camera. These variations often necessitate processing each row of a captured image in a different manner.

Many of the systems described above perform well under certain driving conditions and often require that a certain set of assumptions are valid. Some of these assumptions include the presence of strong lane marker contrast and roads devoid of artifacts such as cracks, arrows, or similar markers. Unfortunately, these assumptions do not hold in many high traffic urban streets and highways. No existing publicly cited literature documents all around performance of a lane detection

system on all streets and highways around the world; hence, there is still scope for improvement as robust lane detection still remains unsolved. **(See Mark Smith's comment and make the appropriate corrects)**

## 3. METHODOLOGY

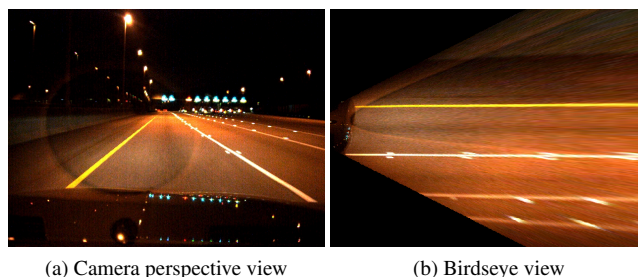
This paper extends the layered lane detection approach in [5] by (1) using an inverse perspective mapping, (2) applying a random sample consensus to help eliminate outliers, and (3) using a Kalman filter for prediction and smoothing. In the following sections, the various components of our lane detection system are described.

### 3.1. Image Enhancement

The acquired Bayer array images are reconstructed into their RGB representation **Isn't this done automatically by the camera?** and then converted to grayscale. The grayscale images are then temporally blurred by averaging  $N$  **(how many?)** successive frames. This smoothing helps connect dashed lane markers to form a near continuous line [5].

### 3.2. Inverse Perspective Mapping

The next step is to perform an inverse perspective mapping (IPM) on the images. This transformation is used to change the captured images from a camera perspective to a bird's eye view as illustrated in Fig. 1. [7, 8, 9]. With this transforma-



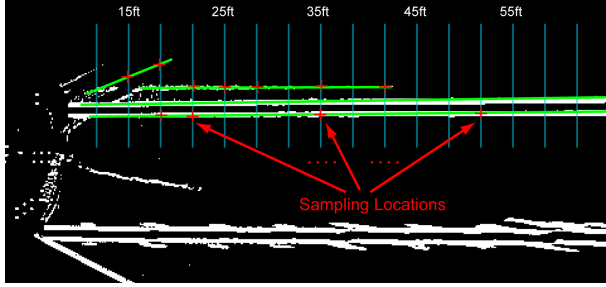
**Fig. 1:** Inverse perspective mapping transforms a camera perspective image into a birdseye view image

tion, lane detection now becomes a problem of detecting two parallel lines that are generally separated by a given, fixed

distance. In addition, this transformation enables a mapping between pixels in the image plane to their corresponding locations in the world with metric **What are metric coordinates? Is this important?** co-ordinates. Camera parameters such as height from the ground, inclination and horizontal/vertical viewing angles need to be determined ahead of time to ensure an accurate transformation.

### 3.3. Lane Candidate Location Detection

Next, an adaptive threshold is applied to each IPM image to generate a binary image [5]. Each binary image is then split into two halves, each one presumably containing one lane marker. A low resolution Hough transform is then performed on the binary images and a set of  $X$  ??? highest scoring lines are found for each half image [5]. Each line is then sampled along its length at a specified distance as indicated by the red “+” in each line in Fig. 2. To find the approximate center

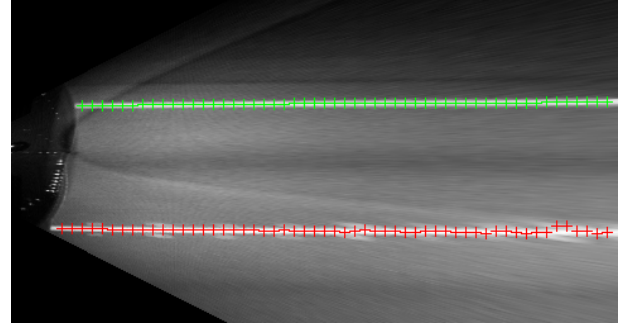


**Fig. 2:** Green lines indicate the Hough transforms’  $X$  highest scoring lines, and the +’s show some of the locations where the lines are to be sampled.

of each line, a one-dimensional matched filter is applied at each sample point along each of the  $X$  lines. As described in [5], the matched filter is a Gaussian with a variance that is a function of the line width. Since the birds eye view created with the IPM produces lines of approximately constant width, a fixed variance Gaussian kernel may be used for the matched filter. After the matched filtering, the pixel with the largest correlation coefficient at each sample point is selected as best estimate of the center of the lane marker as shown in Fig. 3.

### 3.4. Outlier Elimination and Data Modeling

Once a candidate lane marker position is identified at each sample point for each candidate line, Random Sample Consensus (RANSAC) is applied to the data points. The generic RANSAC algorithm robustly fits a model through the most probable data set or inliers while rejecting outliers [10, 11]. Linear Least Squares Estimation (LSE) is then used to fit to a line on the inliers. The orientation of line is specified by  $\rho$  (vertical distance to the line from the origin, which is the top left corner pixel) and  $\theta$  (the angle of the line, which generally is close to zero degrees).



**Fig. 3:** The best estimates for lane marker candidates in each half image

### 3.5. Tracking

The parameters of each line is predicted using a Kalman filter. The state vector  $x(n)$  and observation vector  $y(n)$  are defined as

$$\mathbf{x}(n) = \mathbf{y}(n) = \begin{bmatrix} \rho(n) & \dot{\rho}(n) & \theta(n) & \dot{\theta}(n) \end{bmatrix}^T \quad (1)$$

where  $\rho$  and  $\theta$  define the line orientation and  $\dot{\rho}$  and  $\dot{\theta}$  are the derivatives of  $\rho$  and  $\theta$  that are found by forming the difference in  $\rho$  and  $\theta$  between the current and previous frame. **Do not understand the following sentence** Piece-wise linearity is assumed between the frames allowing use of the Kalman filter [12, 13]. The state transition matrix  $A$  is

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

and the  $C$  matrix in the measurement equation is the identity. The independence between the variables in  $x(n)$  and  $y(n)$  allows creation of simple covariance matrices  $Q_w$  and  $Q_v$ .  $Q_w$  and  $Q_v$  represent the process and observation noise respectively [12]. The covariance matrices are defined as identity and multiplied with non-uniform weights along the diagonal, these weights correspond to the variances of the parameters in  $x(n)$  and  $y(n)$ . The Kalman filter recursively predicts the parameters in the state vector from the previously available information [12, 13].

**Not sure about the following ... why would you not just set  $C$  to zero - no measurements? How much is "significant"?** In the case of a lane marker sequence not being detected, the values in  $Q_v$  are increased significantly and  $\hat{x}(n|n)$  is modified as follows

$$\hat{x}(n|n) = \hat{x}(n|n-1) \quad (3)$$

to force the Kalman filter to rely purely on prediction.

Still do not see how filter does smoothing, and I am lost on the next two sentences. Finally, after extraction from  $\hat{x}(n|n)$ ,  $\rho$  and  $\theta$  are transformed back to the image plane and used to model the estimated orientation of the lane marker sequence. The series of iterated matched filtering and tracking is similarly performed on the other half image.

## 4. EXPERIMENTAL ANALYSIS

### 4.1. Hardware

The hardware used to test and evaluate this new lane detection system is built around an Intel based computer. A forward facing Firewire color camera is installed below the rear-view mirror so that it has a clear view of the road ahead. The Firewire allows video to be captured at VGA resolution and 30fps. Video from the camera is recorded onto hard disks.

### 4.2. Results

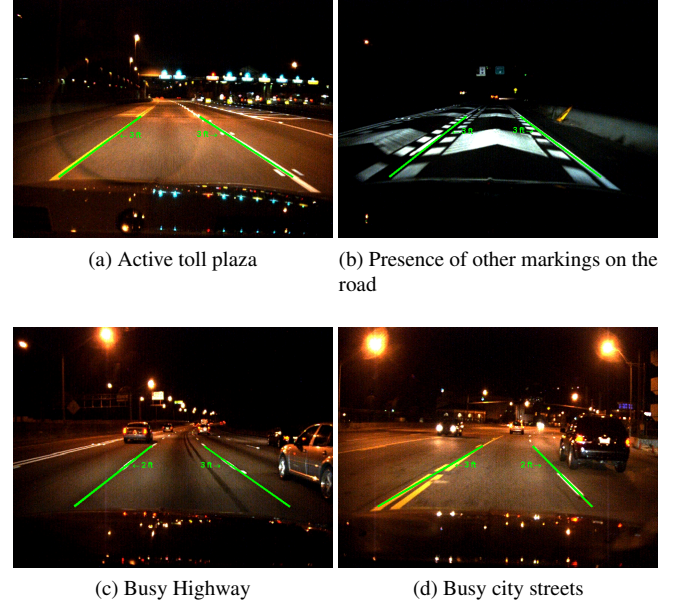
The lane detection algorithm was implemented in Matlab and requires approximately 0.8 seconds to process each frame. The results in Table 1 illustrate the performance of the proposed system when applied to over 10 hours of captured video. The results also show improvement in accuracy over the system developed in [5]. **you may get hammered by not comparing this to other systems and approaches! I would nail you for this!**

**Table 1:** Comparing accuracies of lane detection system

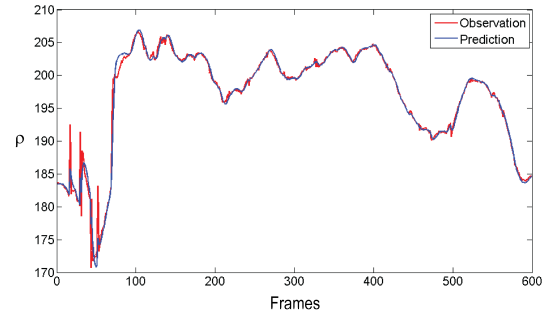
Road type	Traffic type	Average Accuracy Per Minute					
		Current System			Previous System [5]		
		Correct	Incorrect	Misses	Correct	Incorrect	Misses
Isolated Highway	Light	0%	0%	0%	0%	0%	0%
	Moderate	0%	0%	0%	0%	0%	0%
Metro Highway	Light	0%	0%	0%	0%	0%	0%
	Moderate	0%	0%	0%	0%	0%	0%
City Streets	Variable	0%	0%	0%	0%	0%	0%

The captured videos contain scenes with a variety of traffic and illumination conditions that depict environments a driver might encounter [5]. Fig. 4 shows a few of instances of detected lane markers. The metric used to test the quality of the lane detections is accuracy per minute (**what does this mean ???**). This metric allows portability and consistency when testing videos with different frame rates. Since defining a ground truth for the data is extremely tedious, it is commonly avoided. Detections are generally quantified based on visual inspection. Fig. 5 shows a comparison between the observed and predicted value of  $\rho$ . As pure measurements tends to be noisy, the Kalman filter in this context serves as a low-pass filter by smoothing the observed values. (**Sure?**)

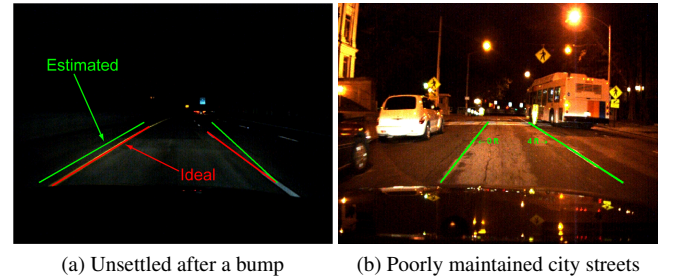
A few instances of incorrect lane detections are shown in Fig. 6. Fortunately in Fig. 6a, the Kalman filter is able to settle within a couple of milliseconds after passing the bump on the road. However, in Fig. 6b, the absence of lane markers



**Fig. 4:** Successful lane detections in various environments



**Fig. 5:** Comparison between observed and predicted values of  $\rho$  over a range of frames



**Fig. 6:** Inaccurate lane detections in a few scenes

due to aging and wears leads to detection and tracking false positives like cracks.

## 5. CONCLUSIONS

The work presented in this paper is a significant improvement over the layered lane detection system presented in [5]. The addition of features such as i) Inverse Perspective Mapping (IPM) ii) Random Sample Consensus (RANSAC), and iii) Kalman filtering has added to the novelty and extension over the previous system. IPM aided in simplifying the process of finding candidate lane markers, while RANSAC helped in rejecting outliers within the estimations. Finally, the Kalman filter ignored minor perturbations and kept the lane marker sequence on its track.

The data set used to test the accuracy of the proposed system was recorded on Interstate highways and city streets in and around Atlanta, GA. Despite the variety in traffic conditions and road quality encountered, the proposed system still yielded good performance as reflected in Table 1.

## 6. FUTURE WORK

Lane Departure Warning (LDW) will be implemented in the not too distant future as the proposed lane detection system is able to accurately determine the distance to the lane markers on either side. By analyzing the velocity and acceleration of lane marker movement, the driver can be notified of an upcoming lane change. Further investigation is needed to enable day time lane detection as current assumptions lead to less than satisfactory results. In addition, the implemented algorithms will be ported to C++ with the help of existing libraries like OpenCV and VXL to facilitate a real-time system.

## 7. REFERENCES

- [1] A.A.M. Assidiq, O.O. Khalifa, R. Islam, and S. Khan, "Real time lane detection for autonomous vehicles," in *International Conference on Computer and Communication Engineering*, 2008, pp. 82–88.
- [2] C. C. Wang, S. S. Huang, and L. C. Fu, "Driver assistance system for lane detection and vehicle recognition with night vision," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, 2005, pp. 3530–3535.
- [3] T. Y. Sun, S. J. Tsai, and V. Chan, "HSI color model based Lane-Marking detection," in *IEEE Intelligent Transportation Systems Conference*, 2006, pp. 1168–1172.
- [4] K. Y. Chin and S. F. Lin, "Lane detection using color-based segmentation," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2005, pp. 706–711.
- [5] A. Borkar, M. H. Hayes, M. T. Smith, and S. Pankanti, "A layered approach to robust lane detection at night," in *2009 IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems*, 2009.
- [6] H. Wang and Q. Chen, "Real-time lane detection in various conditions and night cases," in *Proceedings of the IEEE Intelligent Transportation Systems Conference*, 2006, pp. 17–20.
- [7] S. Sehestedt, S. Kodagoda, A. Alempijevic, and G. Dissanayake, "Robust lane detection in urban environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 123–128.
- [8] Y. Shu and Z. Tan, "Vision based lane detection in autonomous vehicle," in *Fifth World Congress on Intelligent Control and Automation*, 2004, vol. 6, pp. 5258–5260.
- [9] M Bertozzi and A Broggi, "GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Transactions on Image Processing*, vol. 7, no. 1, pp. 62–81, 1998.
- [10] Richard Hartley and Andrew Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2 edition, Apr. 2004.
- [11] Martin A. Fischler and Robert C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [12] Monson H. Hayes, *Statistical Digital Signal Processing and Modeling*, Wiley, Mar. 1996.
- [13] Eli Brookner, *Tracking and Kalman Filtering Made Easy*, Wiley-Interscience, Apr. 1998.