ROBUST LANE DETECTION AND TRACKING WITH RANSAC AND KALMAN FILTER

Amol Borkar, Monson Hayes

Center for Signal and Image Processing (CSIP)
Georgia Institute of Technology
Atlanta, GA
{amol,mhh3}@gatech.edu

Mark T. Smith

Institut för Tillämpad Informationsteknik
Kungliga Tekniska Högskolan
Stockholm, Sweden
msmith@kth.se

ABSTRACT

In a previous paper, a simple approach to lane detection using the Hough transform and iterated matched filters was described [1]. This paper extends this work by incorporating an inverse perspective mapping to create a bird's-eye view of the road, applying random sample consensus to help eliminate outliers due to noise and artifacts in the road, and a Kalman filter to help smooth the output of the lane tracker.

Index Terms— Lane detection, Hough transform, Kalman filter.

1. INTRODUCTION

Driver safety on the highways has been an area of interest for many years. With the development of fast, cheap, low-power, and sophisticated electronics, automobiles with sensors, electronics, and warning systems are beginning to appear on the market.

One of the interesting areas of research and development is collision avoidance. An important component for effective collision avoidance is lane detection. The ability to detect sudden or unexpected lane changes when there is traffic in the lane a driver is moving into could help a driver to avoid collisions. Effective monitoring of the position of a car within a lane could be used to help avert a collision due to driver distractions, fatigue, or driving under the influence of a controlled substance. There are obvious difficulties and challenges in designing collision avoidance systems, and some of the challenges fall outside the realm of engineering and involve complicated issues related to law and liability.

In this paper, we address some of the image processing challenges is designing a lane detection system. It is organized as follows. After a brief survey of some previous research, we then describe the various components of the system. These include image pre-processing using temporal blurring, inverse projective mapping to create a bird's-eye view of the road, a Hough transform for detecting candidate lane markers, a random sample consensus algorithm to help deal with outliers in the image, and tracking of the lane parameters using a Kalman filter. Then, we briefly describe the hardware that was used to collect data, and then show the performance of the lane tracking system. It is shown that this system exhibits considerable improvement in performance compared to a system using only the Hough transform and matched filtering that was previously described [1].

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 it to apply an edge detector to the data [2, 3]. 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 [4, 5]. 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 [1, 2, 4, 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 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.

3. METHODOLOGY

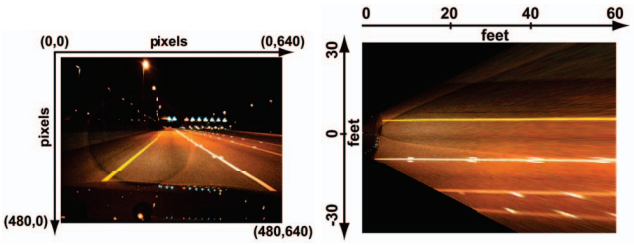
This paper extends the layered lane detection approach in [1] by (1) using an inverse perspective mapping, (2) applying a random sample consensus to help eliminate outliers, and (3) using a Kalman filter for prediction and smoothing. In the following sections, the various components of the lane detection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

3.1. Image Enhancement

The captured color images undergo a grayscale transformation and temporal blurring by averaging $N = 3$ successive frames. This smoothing helps connect dashed lane markers to form a near continuous line [1].

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 transformation, lane detection now becomes a problem of detecting a pair of 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 world co-ordinates (feet) as shown in Fig. 1b. The camera's intrinsic and extrinsic parameters are necessary to ensure an accurate transformation.



(a) Camera perspective view. (b) Bird's-eye view.

Fig. 1: Inverse perspective mapping transforms a camera perspective image into a bird's-eye view image.

3.3. Lane Candidate Location Detection

Next, an adaptive threshold is applied to the IPM image to generate a binary image [1]. Each binary image is then split into two two halves, each one presumably containing one lane marker. A low-resolution Hough transform is then computed on the binary images and the ten highest scoring lines are found for each half image [1]. Each line is then sampled along its length at a specified distance as illustrated by the red plus signs in Fig. 2. To find the approximate center of each line, a one-dimensional matched filter is applied at each sample point along each line. As described in [1], the matched filter is a Gaussian with a variance that is a function of the line width. Since the bird's-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 that exceeds a predetermined threshold is selected as the best estimate of the center of the lane marker as indicated by the green plus signs in Fig. 3. The minimum threshold helps in ignoring false positives like cracks, tar patches or cases where lane markers do not exist.

Fig. 2: The green lines represent high-scoring lines from the Hough transform, and the plus signs indicate the points where each line is to be sampled.

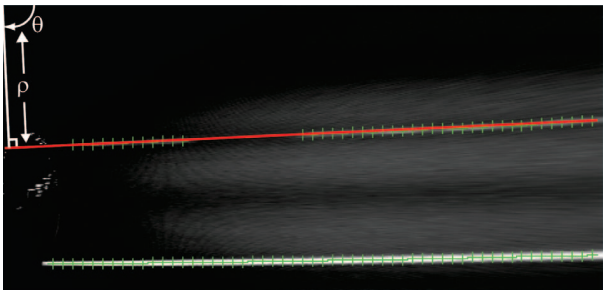


Fig. 3: Line fitted through one set of candidate points and parameterized with ρ and θ.

3.4. Outlier Elimination and Data Modeling

Once the center of each candidate line at each sample point has been estimated, 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. Fig. 3 illustrates the parametrization of the fitted line in terms of ρ and θ where ρ is the distance from the origin (top left corner pixel) to the line and θ is the angle as indicated in Fig. 3 (generally is close to $90°$).

3.5. Tracking

The parameters of each line is predicted using a Kalman filter. The state vector $\mathbf{x}(n)$ and observation vector $\mathbf{y}(n)$ 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 ρ and θ define the line orientation and $\dot{\rho}$ and $\dot{\theta}$ are the derivatives of ρ and θ that are estimated using the difference in ρ and θ between the current and previous frame. 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 matrix C in the measurement equation is the identity matrix. The noise in the state and measurement equations is assumed to be white and each process is assumed to be uncorrelated with the others. Therefore, the covariance matrices for these vector random processes are constant and diagonal. The variance of each noise process is estimated off-line using frames in which accurate estimates of the lanes were being produced. In the case of a lane markers not being detected, the matrix C is set to zero forcing the Kalman filter to rely purely on prediction. Finally, the estimated line is mapped back to the camera perspective to portray the lane detection results.

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. Video is captured in VGA resolution at 30fps.

4.2. Results

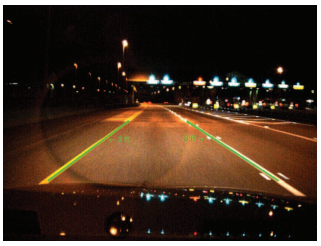
The lane detection algorithm was implemented in Matlab and requires approximately 0.8 seconds to process each frame. Table 1 and Table 2 illustrate the performance of the current and previous lane detection systems when applied to over 10 hours of captured video. The results in Table 1 show an improvement in accuracy over the system described in [1] when tested with similar data sets. The lack of accessibility to other lane detection algorithms and turnkey software systems makes it extremely difficult to compare results. In addition, defining a ground truth for the data is extremely tedious; hence, it is commonly avoided. Consequently, detections were qualitative and based purely on visual inspection by single user. The following rules were used to quantify the results into the different categories: i) a correct detection occurs when more than 50% the of lane marker estimate is overlaid on a lane marker in the scene, ii) an incorrect detection occurs when the estimate is overlaid on something else other than a lane marker, and iii) a missed detection occurs when no estimate is presented despite a relevant lane marker being visible. The detection rates of left and right markers are averaged to produce the numbers in the tables. Fig. 4 shows a few instances of correct lane detections. The results are presented in terms of detection rate per minute. This metric allows to normalize the results when data is captured using cameras with different frame rates.

Table 1: Accuracy of the current lane detection system

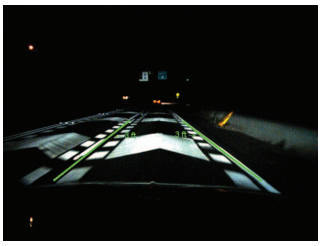
Road Type	Traffic	Avg. Detection Rate Per Minute		
		Correct	Incorrect	Misses
Isolated Highway	Light	99.08%	0.99%	0%
	Moderate	98.34%	1.65%	0%
Metro Highway	Light	98.37%	1.65%	0%
	Moderate	96.34%	3.65%	0%
City	Variable	86.39%	12.71%	0.78%

Table 2: Accuracy of the previous lane detection system [1]

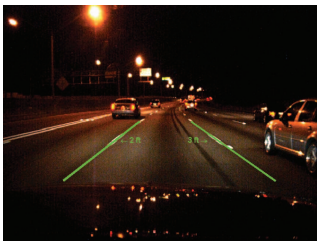
Road Type	Traffic	Avg. Detection Rate Per Minute		
		Correct	Incorrect	Misses
Isolated Highway	Light	89.69%	5.31%	5.19%
	Moderate	89.69%	5.31%	5.19%
Metro Highway	Light	91.47%	6.65%	1.87%
	Moderate	84.97%	10.33%	4.68%
City	Variable	76.55%	11.94%	11.95%



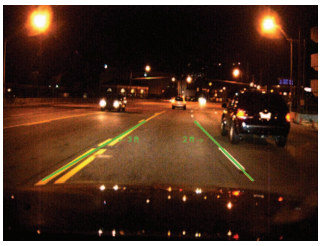
(a) Active toll plaza.



(b) Presence of other markings on the road.



(c) Busy Highway.



(d) Busy city streets.

Fig. 4: Examples of accurate lane detection.

Despite noisy measurements, the Kalman filter recursively estimates the dynamics of the state vector. Fig. 5 shows a comparison between the observed and predicted value of ρ.

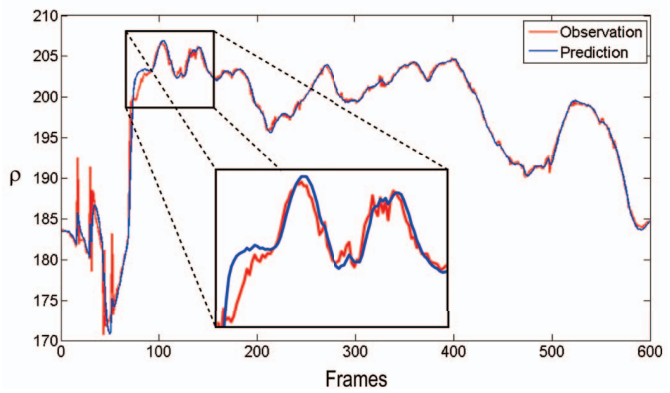


Fig. 5: Comparison between observed and predicted values of ρ over a range of frames. A blow up shows the Kalman filter smoothing the noisy measurements.

(a) Unsettled after a bump. (b) Poorly maintained city streets.

Fig. 6: Examples of incorrect lane detection.

A few instances of incorrect lane detections are also shown in Fig. 6. Fortunately in Fig. 6a, the Kalman filter is able to settle within a few milliseconds after passing the bump on the road. However, in Fig. 6b, the absence of lane markers due to road aging and wear leads to the detection and tracking of false signals such as cracks.

5. CONCLUSIONS

The work presented in this paper is a significant improvement over the layered lane detection system presented in [1]. The addition of features such as (1) Inverse Perspective Mapping (IPM), (2) Random Sample Consensus (RANSAC), and (3) Kalman filtering has added to the novelty and extension over the previous system. IPM aids in simplifying the process of finding candidate lane markers, while RANSAC helps in rejecting outliers within the estimations. Finally, the Kalman filter ignores minor perturbations and keeps the lane marker sequence on its track.

The data set used to test the accuracy of the proposed system was recorded on highways and 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 future. It will leverage from the lane detection system's ability to accurately determine the distance to the lane markers as shown in Fig. 4. In addition, the implemented algorithms will be ported to C# and C++ to facilitate a real-time system. Future data sets will also include ground truth information to allow accurate error calculation. Additional users will also perform visual inspection. Finally, the image enhancement stage will be made adaptive by computing N as a function of the speed of the vehicle.

7. REFERENCES

[1] A. Borkar, M. Hayes, M. 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, pp. 51–57.

[2] A. Assidiq, 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.

[3] 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.

[4] T. Sun, S. Tsai, and V. Chan, "HSI color model based Lane-Marking detection," in *IEEE Intelligent Transportation Systems Conference*, 2006, pp. 1168–1172.

[5] K. Chin and S. Lin, "Lane detection using color-based segmentation," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2005, pp. 706–711.

[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] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2 edition, Apr. 2004.

[11] M. Fischler and R. 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.