2014 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)

ROBUST LANE DETECTION & TRACKING BASED ON NOVEL FEATURE EXTRACTION AND LANE CATEGORIZATION

Umar Ozgunalp and Naim Dahnoun

Department of Electrical and Electronic Engineering
University of Bristol
Merchant Venturers Building
Woodland Road, Bristol
BS8 1UB, United Kingdom

ABSTRACT

In this paper, we introduce a robust lane detection and tracking algorithm to cope with complex scenarios and to decrease the effect of thresholds. For lane feature extraction, an extension to the symmetrical local threshold (SLT) is proposed to improve the feature map and obtain orientation information. Then, while creating a Hough accumulator, obtained orientation information is used to decrease computational complexity ($\approx$ 60 times) and acquire a clearer accumulator. The left and right lanes are categorized by applying a mask on the Hough accumulator, which leads to low computational complexity and reduced sensitivity to thresholding. To quantify the new feature map, we used ground truth lane markings from the RoMa Datasets and the optimum true positive (TP) to positive (P) ratio increased from 69% to 86% on average, compared to the SLT. The successful lane detection rate calculated from more than 10K frames is, 96.2%, demonstrating the robustness of the system.

Index Terms— Lane feature extraction, Lane detection, Hough transform, Kalman filter

1. INTRODUCTION

Lane detection is one of the key elements of the Driver assistance systems (DAS) [1] and it is necessary for lane departure warning systems or fully autonomous ground vehicles. Lane detection algorithms should perform robustly for a wide variety of environments in real-time. However, due to changing environments, lane detection can be a difficult task in some cases. For example, changing light conditions, shadows on the road or lack of consistent painting can affect lane detection performance significantly. Thus, to improve the performance, many lane feature extractors have been developed to supply less noisy feature maps to the optimization stage. Although there are many feature extractors in the literature, most of them are not quantitatively evaluated. One significant work that compares their performance has been detailed by [2]. The authors evaluated the most common feature extractors in the literature, including edge detectors [3], top hat filters [4], steerable filters [5], global threshold [6], local threshold [7] and SLT [2]. Among the tested feature extractors, the authors [2] concluded that, despite its low computational complexity, the symmetrical local threshold gave the best results among the evaluated feature extractors. The problem with the symmetrical threshold is that it does not use any orientation information and it does not supply any orientation information to the optimization stage. However, using orientation information along with the feature points location would decrease the noise in the optimization stage, such as when creating the lane likelihood function using the orientation [8] or while using the Hough transform. Using the orientation information during the Hough transform not only decreases the noise in the accumulator but also decreases the computational complexity by voting only to the angle range around the feature point angle. Due to the global nature of the Hough transform [9], along with other methods, the Hough transform [10–15] is a popular and an effective method for lane detection.

2. LANE DETECTION

2.1. Symmetrical Local Threshold

The SLT [2] algorithm processes each row of the image independently. It relies on the Dark-Light-Dark (DLD) transition property of the lanes to distinguish the painted lanes from the noise, such as shadows and cracks on the road. The algorithm checks all points on the grey scale image. In the first step, for each input point (I_p), the algorithm calculates the average intensity value of all the points which appear in the same row and on the left hand side ($Average_L$) of the input point (within the range) and, then, repeats for the right hand side ($Average_R$) of the input point. If the intensity of the input

Algorithm 1 Proposed feature extraction algorithm

for $I_p \in image\ size$ **do**
  if $I_p > Average_R + T_h$ and $I_p > Average_L + T_h$ **then**
    Calculate $\theta_{\max(I_{areaL})}$ and $\theta_{\max(I_{areaR})}$
    if $\left|\theta_{\max(I_{areaL})} - \theta_{\max(I_{areaR})}\right| < \theta_{TH}$ **then**
      $\theta_{I_p} = \frac{1}{2}\left(\theta_{\max(I_{areaL})} + \theta_{\max(I_{areaR})}\right)$
    end if
  end if
end for

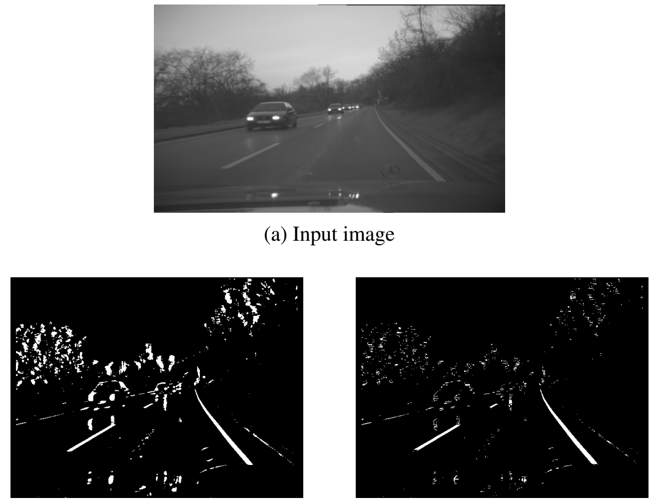
point minus the threshold (T_h) is larger than both the left and right averages, then the point is considered as the lane feature point and labelled accordingly in the feature map. In the final stage, a one dimensional connected component analysis is applied to remove further noise such as salt and pepper noise.

2.2. Extension of Symmetrical Local Threshold

Despite its low computational complexity, compared to many lane feature extractors, the SLT gives better results [2]. As stated before, the SLT uses only the DLD transition property of lanes and ignores the fact that the left and right boundaries of a lane marking should be parallel to each other in the world coordinate system. In the image coordinate system, this property is not valid due to the perspective mapping effect of the imaging. However, depending on the camera parameters and the actual lane width, the orientation difference between the left and right boundaries of a lane should not be more than a few degrees. Using this information, along with the DLD transition property of a lane, can improve the resultant feature map.

In this paper, to improve the feature map, first the SLT is applied to the grey scale image (excluding the connected component analysis). As a second step for each initially estimated feature point, both its left hand side and right hand side are searched for the lane borders (a point from each side with the highest intensity change within the search range). If the angles of the resultant lane borders are close to each other, the orientation of the feature point is calculated by averaging the angles of the detected lane borders. Otherwise, the feature point is eliminated from the feature map. The proposed feature extraction algorithim is demonstrated in pseudo code 1. In Fig. 1, feature maps for an image using the SLT and the proposed algorithm are demonstrated. Fig. 1(a) is the input image, Fig. 1(b) is the feature map of the SLT without the connected component analysis and Fig. 1(c) is the proposed feature map without the connected component analysis.

To quantify our results and compare them with the original SLT, ground truth lane markings supplied by RoMa datasets [2] are used. True positive (T_p) over all positive (P) feature points are calculated for a large range of threshold values for both of the algorithms for all the images in the dataset and averaged. As seen in Fig. 2, the proposed algorithm outperforms the SLT for almost all ranges of thresholds and the optimum average T_p/P ratio increases from 69% to 86%.



(a) Input image

(b) Output of SLT without connected component analysis

(c) Output of Proposed algorithm without connected component analysis

Fig. 1: Example lane detection results from different video sequences

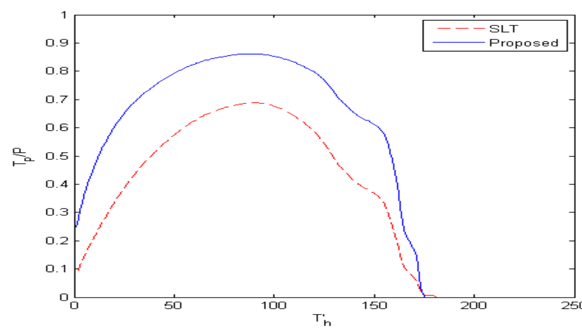


Fig. 2: T_p/P ratio for range of thresholds

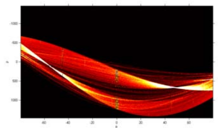
2.3. Distance Transform

The last stage of the SLT is to apply the connected component analysis to the feature map to eliminate isolated noise. For this purpose, there are many tools developed in the literature, such as the low pass filter, median filter or morphological operators. In this paper, to reduce the noise in the feature map, the distance transform has been used. For each feature point, the distance transform calculates the distance between the feature point and the nearest non feature-point in the feature map. In the resultant feature map, each point has a weight and is no longer binary. This process has two advantages. Firstly, the isolated pixels have less weight than the unisolated ones. Thus, they have less effect on the optimization

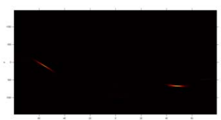
(a) Input image

(b) Section of a extracted feature map

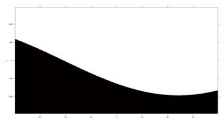
(c) Output of the distance transform



(d) Hough accumulator when voted for all range of orientations



(e) Hough accumulator when voted for only around feature point orientations



(f) Mask for Hough accumulator

Fig. 3: Example Hough Transform

stage. Secondly, in the weighted feature map, feature points which appear on the centre of the lane have more weight than the others (this helps to detect the centre of the lanes). This is illustrated in Fig. 3. Fig. 3(a) is the input coloured image, in Fig. 3(b) a section of an extracted feature map (bottom left of the image) is shown and in Fig. 3(c) the output of the distance transform for the same section is illustrated.

2.4. Hough Transform

Initially, the algorithm averages the last few frames to extract longer lane segments from the dashed lanes and, then, extracts features followed by the distance transform. To estimate the lane parameters, the algorithm forms a 2D accumulator for the Hough transform in which its axes are ρ (the distance between the origin of the Hough transform and the lane) and θ (the angle). In the Hough transform, each cell of the accumulator represents a line in the image domain and this line has the following equation:

$$\rho = x.cos(\theta) + y.sin(\theta) \tag{1}$$

Conventionally, each feature point needs to calculate a ρ value for all possible θ (between -90^0 to 90^0 since the SLT does not supply any orientation information). However, with a known feature point orientation information, the algorithm can vote for only around feature points θ value (i.e $\theta \pm 1^0$). This process will decrease the computational complexity of the Hough transform by a factor of 60 times and outputs a cleaner Hough accumulator. In Fig. 3(d) the Hough accumulator is illustrated when the Hough transform is applied for the complete orientation range for the input image and in Fig. 3(e) the Hough accumulator is illustrated when the Hough transform is applied only for the orientations estimated from the feature map for the input image.

2.5. Optimized Lane Categorization

At this stage, lane parameters need to be estimated using the Hough accumulator. Simple thresholding applied to the Hough accumulator would not be a robust enough solution since, along with actual lanes, many false lanes can be detected due to noise. Thus, it is better to benefit from the road structure information to minimize false lane departure warnings. Both left and right lane markings must be detected. Therefore, lanes are categorized into two domains, left and right domains in the Hough accumulator. Lane categorization is achieved by finding the intersection of the line and the bottom row of the image for each cell in the Hough accumulator. To do so, for each cell, an intersection point should be calculated by using Equation 2.

$$P_{down} = (\rho - H \times sin(\theta))/cos(\theta) \tag{2}$$

where H is the image height and P_{down} is the intersection point between the lane and bottom row of the image. P_{down} should be calculated for each cell of the Hough accumulator to seperate the left and right lanes. However, the equation to estimate the intersection point is only dependent on the image resolution. Thus, the algorithm creates a binary mask for the Hough accumulator using the image resolution (Resolution is same for the whole video sequence) and, by simply multiplying the Hough accumulator with the labelled mask, the algorithm categorizes the lanes in an optimized way. The algorithm needs to detect both left and right lanes. Therefore, from each domain, one lane that has the highest votes is selected. In Fig. 3(f), the calculated mask for the Hough accumulator is illustrated. Lanes closer to the centre of the image generally tend to have higher peaks in the Hough accumulator. However, if the closer lane is not well painted or blocked by an obstacle, this lane can have a lower likelihood than the next lane. In these cases, the proposed algorithm assumes the lanes are parallel to each other. Therefore, it uses the vanishing point and road width cues to define a new region of interest (ROI). Then, the skipped lane is detected by searching the Hough accumulator cells where their corresponding lanes appear on the ROI and cross the vanishing point.

3. TRACKING

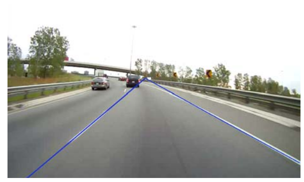
After detecting the left and right lanes in a frame, these lanes are tracked. Initially, a large ROI for the image domain is created, depending on how well the lanes are detected (the number of votes they get from the HT), to remove further noise from the feature map. This ROI is defined by the detected lane positions and vanishing point since anything above the horizon line is noise for lane detection. Then, four parameters, including the intersection between the left lane and the bottom row of the image, the intersection between the right lane and the bottom row of the image and the vanishing point (V_x,V_y), are tracked. In the following frames, the created ROI changes position according to the predicted positions of the lanes. Also, the ROI shrinks and expands according to the standard deviations of the predicted parameters. In the case of a lane change, when one of the predicted lanes crosses from the centre, both the predicted lane positions and the ROI are updated by using the lane width cue. Although tracking works robustly, it can occasionally fail. To drop tracking, two parameters are used. The first parameter is the average of votes each lane gets from the Hough accumulator for the last 30 frames. The second, parameter is the road width. If a lane splits into two and the tracking algorithm follows the lane which is further away than the closer lane, then the algorithm will keep consistently getting a high number of votes. However, the tracked lane would not be the desired lane. Thus, the algorithm uses the road width to drop tracking and restart detection to avoid such cases.

4. EXPERIMENTAL RESULTS

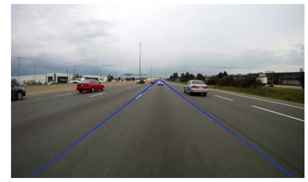
The algorithm has been tested on video sequences which are taken from both urban areas and challenging rural areas. To quantify the proposed feature extraction algorithm, 116 ground truth images taken from different scenes have been tested and the optimum average true positive to average positive ratio increased from 69% to 86%. A complete lane detection system was also tested using a total number of 10689 frames recorded by three different cameras with different parameters (aspect ratio, frame rate, resolution, field of view, etc.). The detection rate is estimated as 96.2% on average. Within these three video sequences, tracking parameters are only selected for sequence one and the same parameters are used for sequence two and three. To minimize the effect of different camera parameters, all the images are resized to the same resolution. While sequence one was taken with the camera fixed to the dashboard, sequences two and three were taken by a hand held cameras without fixing them to the car. Apart from the difficulties mentioned above, sequence two was taken when there were many shadows on the road and sequence three was taken during the night. While all the sequences showed robust results, sequence one had a higher detection ratio than the other two, due to more complex scenarios and difficulties that were intentionally introduced to the other two algorithms. The detailed detection results are illustrated in the Table 1 and sample video results are also available at http://rwnlabs.co.uk/umar/.

Table 1: Detection results

Sequence	Total frames	Correct detection	Incorrect detection	Mis-Detectin
Highway	5677	5645	30	3
shadow	2090	2029	0	61
Night	2920	2606	314	0



(a) A result from sequence 1



(b) A result from sequence 1

(c) A result from sequence 2

(d) A result from sequence 3

Fig. 4: Example lane detection results from different video sequences

5. CONCLUSION

We have presented a lane detection algorithm which performs robustly for a wide variety of environments which includes poorly maintained lanes, dashed lanes, vertical road curvature, horizontal road curvature, illumination changes and night scenes with a detection ratio of 96.2%. For the feature map extraction, an extension to the SLT is also proposed. To quantify improvement on feature extraction, ground truth lane markings have been used from the RoMa datasets. At almost all the threshold values (all usable thresholds), an improvement is observed and, at the optimum threshold, T_P/P ratio increased from 69% to 86%. The estimated feature map is then coupled with the distance transform to give a lower weighting to the isolated points and at the same time to be able to detect the centre of the lane. The Hough transform was also applied at a cost of a much decreased computational complexity ($\approx$ 60 times), thanks to the supplied orientation information from the feature map. Using the orientation information also decreased the noise in the resultant Hough accumulator. Furthermore, the proposed algorithm detected lanes using the road structure and this is achieved by categorization of the lanes in the Hough accumulator in an optimized way (by using mask for the Hough accumulator).

6. REFERENCES

[1] Aharon Bar Hillel, Ronen Lerner, Dan Levi, and Guy Raz, "Recent progress in road and lane detection: a survey," *Machine Vision and Applications*, pp. 1–19, 2012.

[2] Thomas Veit, J-P Tarel, Philippe Nicolle, and Pierre Charbonnier, "Evaluation of road marking feature extraction," in *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*. IEEE, 2008, pp. 174–181.

[3] Yue Wang, Eam Khwang Teoh, and Dinggang Shen, "Lane detection and tracking using b-snake," *Image and Vision computing*, vol. 22, no. 4, pp. 269–280, 2004.

[4] Didier Aubert, Karl Kluge, and Chuck Thorpe, "Autonomous navigation of structured city roads," *Proceedings of SPIE Mobile Robot V*, 1990.

[5] Joel C McCall and Mohan M Trivedi, "An integrated, robust approach to lane marking detection and lane tracking," in *Intelligent Vehicles Symposium, 2004 IEEE*. IEEE, 2004, pp. 533–537.

[6] MPN Burrow, HT Evdorides, and MS Snaith, "Segmentation algorithms for road marking digital image analysis," *Proceedings of the ICE-Transport*, vol. 156, no. 1, pp. 17–28, 2003.

[7] Alberto Broggi, Andrea Cappalunga, Claudio Caraffi, Stefano Cattani, Stefano Ghidoni, Paolo Grisleri, Pier Paolo Porta, Matteo Posterli, and Paolo Zani, "Terramax vision at the urban challenge 2007," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 1, pp. 194–205, 2010.

[8] Chris Kreucher, Sridhar Lakshmanan, and Karl Kluge, "A driver warning system based on the lois lane detection algorithm," in *Proceedings of IEEE International Conference on Intelligent Vehicles*. Stuttgart, Germany, 1998, pp. 17–22.

[9] R Guerreiro and P Aguiar, "Connectivity-enforcing hough transform for the robust extraction of line segments," 2011.

[10] Richard O Duda and Peter E Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[11] Joel C McCall and Mohan M Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 7, no. 1, pp. 20–37, 2006.

[12] Amol Borkar, Monson Hayes, and Mark T Smith, "Robust lane detection and tracking with ransac and kalman filter," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE, 2009, pp. 3261–3264.

[13] Abdulhakam AM Assidiq, Othman O Khalifa, R Islam, and Sheroz Khan, "Real time lane detection for autonomous vehicles," in *Computer and Communication Engineering, 2008. ICCCE 2008. International Conference on*. IEEE, 2008, pp. 82–88.

[14] Amol Borkar, Monson Hayes, Mark T Smith, and Sharathchandra Pankanti, "A layered approach to robust lane detection at night," in *Computational Intelligence in Vehicles and Vehicular Systems, 2009. CIVVS'09. IEEE Workshop on*. IEEE, 2009, pp. 51–57.

[15] M Boumediene, A Ouamri, and N Dahnoun, "Lane boundary detection and tracking using nnf and hmm approaches," in *Intelligent Vehicles Symposium, 2007 IEEE*. IEEE, 2007, pp. 1107–1111.