Efficient and Robust Lane Detection Using Three-Stage Feature Extraction with Line Fitting

Aming Wu and Yahong Han(✉)

School of Computer Science and Technology, Tianjin University, Tianjin, China
{tjwam,yahong}@tju.edu.cn

Abstract. Lane detection plays an important role in advanced driver assistance system. The objective of lane detection is to separate lane markings from background and to locate their positions. Challenges of lane detection mainly lie in large appearance variations. These variations mainly include shadows, occlusion, variable lighting conditions, and different weather conditions, e.g., sunny or rainy days. In this paper, we propose a method named Lane Detection with Three-stage Feature Extraction to detect lane proposals. The three-stage features include contrast feature, line structure feature, and convex signal feature. Three feature extraction methods are devised for each of the three-stage respectively. After getting lane proposals through the processing of the three methods, we use straight line model to fit these proposals based on RANSAC method. Extensive experiments demonstrate that our method is stable and can find lane proposals exactly given the lane in the scene.

Keywords: Lane detection · Three-stage feature extraction
RANSAC

1 Introduction

Within recent years, more and more researches about advanced driver assistance systems [1] have been developed to ensure road safety. There are many components in advanced driver assistance system such as vehicle recognition and pedestrian recognition etc. Lane Departure Warning [2] is an important module in advanced driver assistance system. The module could reduce some accidents which are caused by driver error and distractions. And Lane detection is an important step of lane departure warning system.

The objective of lane detection is to separate lane markings from background clutter and to locate their exact positions [3]. In this work, we focus on vision-based approaches. As images contain rich scene information, we can get useful information from them. However, there still lies lots of challenges for the vision-based methods. Firstly, because realistic road scene is more complex, the captured scene images always contain noises, such as shadows, variable lighting

B. Zeng et al. (Eds.): PCM 2017, LNCS 10735, pp. 464–473, 2018.
https://doi.org/10.1007/978-3-319-77380-3_44

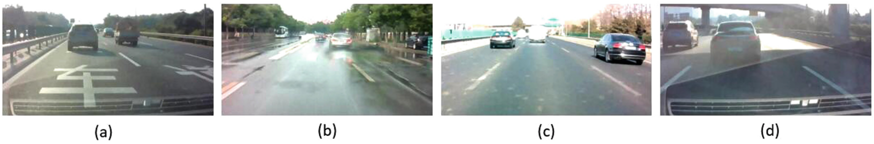

Fig. 1. Images of advanced driver assistance system. In (a), road is full of noises. In (b), in the rainy day, the lane is very blurry. In (c), the lane is worn. In (d), there are variant lighting conditions.

conditions, and variable weather conditions etc. Secondly, images captured by the advanced driver assistance system are often blurred by the windshield. In Fig. 1, we show some examples. Thus, how to extract robust lane features turns out to be a challenge.

Many algorithms for lane detection have been proposed recently. In [4,5], authors extract ridges as lane feature and implement RANSAC [6] to fit models. This method could find lane feature exactly, but extracting ridge is very complex and needs large computation. In [7], the author proposed a method which is based on Inverse Perspective Mapping (IPM) of road. He first gets the IPM of road and then filters with selective oriented gaussian filters and uses RANSAC algorithm to fit lines. However, the computation of IPM and the gaussian filters makes this method heavy in computation cost. In [8,9], authors proposed methods which are based on spine model and B-Snake model. But if we use these methods, we should firstly determine a set of control points exactly. In [10], authors first extend a road segmentation method [11] in an illumination-invariant color image and use a ridge operator to detect lane markings. Then they use RANSAC to estimate road geometry. As this method is segmentation depended, the performance is easily affected by noises. Thus, for all these methods, the common practice is firstly searching for lane features and then using a model to fit these extracted features.

Recently, learning-based methods have also shown their promising performance. In [12], the author proposed the support vector machine (SVM) based lane detection algorithm. However, learning based methods require lots of well-labeled training data to ensure the good performance. And it is hard and expensive to collect these well-labeled training data.

In this work, we propose a method of using three-stage feature extraction to locate the lane positions. The three-stage features include contrast feature, line structure feature, and convex signal feature. First, we use the three-stage method to get lane proposals and then use RANSAC to fit these proposals. Experiments show that our method could efficiently extract lane proposals.

2 Three-Stage Feature Extraction

Let $f(x,y)$ be gray value of a pixel of the lane image. Here x and y are the coordinates of the pixel. To a real image, we first compute image gradient.

$$G_x = f(x,y) - f(x-1,y) \tag{1}$$

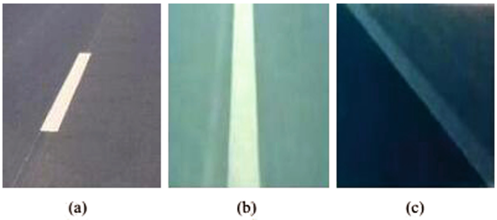

Fig. 2. Obvious contrast of lane. Along the edge of lane, the sum of lane region is higher than the sides of lane.

$$G_y = f(x, y) - f(x, y-1) \tag{2}$$

where G_x is the gradient of horizontal direction and G_y is the gradient of vertical direction. After this processing, a gray image turns into a vector map. Every pixel is represented by a three-dimension vector. Elements of the vector are gray value, vector direction, and vector length. We use the slope to represent the vector direction.

$$k = -1.0/Gd, \tag{3}$$

where

$$Gd = G_y/G_x, \tag{4}$$

where k is the vector direction which is perpendicular to the gradient direction, Gd is the gradient direction.

In the following, we will introduce extraction of the three-stage features in detail. And we only take the left side of lane for example to introduce our methods. We can use the same method to detect the right side of lane.

2.1 Obvious Contrast Feature

Obvious contrast is an important attribute of lane. In Fig. 2, along the left edge of lane, the sum of gray in the right region of left edge is bigger than the sum of gray in the left region of left edge. So if we find a pixel which has the attribute, we save it as a contrast proposal. Our contrast proposal detection procedure works as follows. We first use vector map which is computed using Eqs. (1)–(4) to compute slope and intercept of every pixel. Then we take the pixel as the center and in a small range, we calculate the gray sum in right range and the gray sum in left range. If the sum of right is bigger than the sum of left, we take the pixel as a contrast proposal. The method is detailed in Algorithm 1.

In Fig. 3, we show an example of the Algorithm 1. In (a), we show the source image which is got in rainy day and we only process the region under the red line. In (b), we show the result of Algorithm 1. We can see that lane proposals have been appeared on the result. But because we do not use the structure information, the result is not clear and exact and full of noises.

Algorithm 1. Contrast Proposal Detection

```
Require:
  Obtain vector map using Eq. 1−4
  Set threshold of gradient of horizontal direction $T_{min}$
  Set search range of vertical direction $H_{range}$
  Set sum length of horizontal direction $Length$
Ensure:
  Set of contrast proposal $S_{contrast}$

  for each pixel$(x, y)$ in vector map do
    if $G_x > T_{min}$ then
      Get the slope $k$ using Eq. 3 and compute the intercept $b = y - k * x$
      Set $Sum_L = 0$ and $Sum_R = 0$
      for each $i \in [y - H_{range}, y + H_{range}]$ do
        $t_y = i$ and $t_x = (t_y - b)/km$
        for each $n \in [t_x + 1, t_x + Length]$ do
          $Sum_R = Sum_R + gray(n)$
        end for
        for each $n \in [t_x - Length, t_x - 1]$ do
          $Sum_L = Sum_L + gray(n)$
        end for
        if $Sum_R > Sum_L$ then
          Save the pixel to the set $S_{contrast}$
        end if
      end for
    end if
  end for
```

2.2 Linear Structure Feature

After contrast detection, we obtain the contrast proposal set. Now we need structure information to locate the lane exactly. Based on human impression of lane, the lane should have linear structure. The linear structure is that lane proposals should be on many same lines. So if we can find linear structure, we could find lane proposals much more exactly.

Our linear structure detection procedure now works as follows. We first use the contrast set which is got from Algorithm 1 to compute its corresponding slope and intercept. So the pixel of the contrast set gives a straight line equation. Then to each pixel in the set, we calculate the distance of the pixel to the straight line. If this distance is smaller than our threshold, we think it belongs to the straight line and save it. Because of the curve can be partial linearization, our method also applies to curve lane. The general method is detailed in Algorithm 2.

In Fig. 4(b), we show the result of Algorithm 2. In contrast to (a), we can see that the noise is reduced.

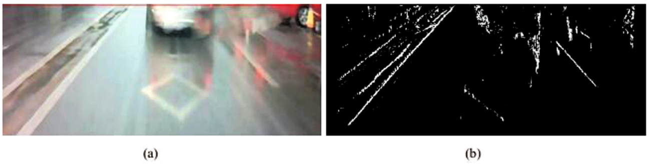

Fig. 3. Result of Algorithm 1. (a) is the source image under the rain. (b) is the result of Algorithm 1. We can find lane structure from (b).

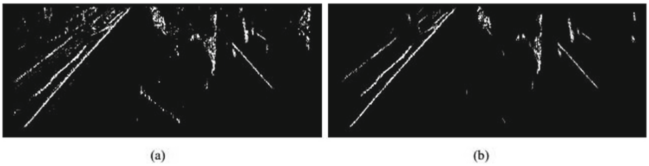

Fig. 4. Result of Algorithm 2. (a) is the result of Algorithm 1. (b) is the result of Algorithm 2. We can see that noise getting less.

2.3 Convex Signal Feature

After processing of the above two steps, we can get lane proposals exactly. However, if we add the constraint of convex signal, the result could get much better. The constraint of convex signal is that the gray sum of lane is much bigger than the sum of both sides of lane. Obviously, this constraint holds in most cases. In practice, the lane is often full of noises and often blurry. In Fig. 5, we show some signal analysis of middle row of lane images in many situations. In (a), (b) and (c), the lane is blurry and full of noise. But all signal structures of these lane image are all convex. So the convex constraint can apply to a number of situations and help to get much more robust lane proposals.

Our convex signal structure detection procedure now works as follows. We first get line structure set which is got from Algorithm 2 and to every pixel of line set, we get its neighbourhood and count the number of pixel which belongs to the contrast set. If the number is smaller than threshold, we throw away the pixel. Then we get gray sum of left region, gray sum of middle region, and gray sum of right region. If sum of middle region is the biggest among these three sum of region, we save corresponding pixel to the ultimate set. The proposed method is detailed in Algorithm 3.

In Fig. 5(b), we show the result of Algorithm 3. In contrast to the result of Algorithm 2, we can see that the noises is reduced. After processing of the above three steps, we can get the final result. Although there may be some noises left in the results, these noises have little impact on the detected lane.

2.4 Vanishing Point Detection

With vanishing point, we can only process the image region under the vanishing point and reduce the amount of computation. In graphical perspective,

Algorithm 2. Linear Structure Detection

```
Require:
  Obtain vector map using Eq. 1−4
  Obtain contrast set $S_{contrast}$ using Algorithm 1
  Set search range of vertical direction $H$
  Set distance threshold $DistThresh$ and threshold $N_{thresh}$
Ensure:
  Linear structure set $S_{line}$

  for each pixel$(x, y)$ in $S_{contrast}$ do
    Get the slope $k$ using Eq. 3 and compute the intercept $b = y-k*x$ Set $Number = 0$
    for each $i \in [y - H, y + H]$ do
      $t_y = i$ and $t_x = (t_y - b)/k$
      if the pixel belongs to $S_{contrast}$ then
        Calculate the distance of pixel to the straight line. The distance is computed
        as follows: $d = |k * t_x - t_y + b|/\sqrt{k * k + 1}$
        if $d < DistThresh$ then
          $Number = Number + 1$
        end if
        if $Number > N_{thresh}$ then
          save these pixels to the set $S_{line}$
        end if
      end if
    end for
  end for
```

a vanishing point is a point in the picture plane that is the intersection of the projections (or drawings) of a set of parallel lines in space on to the picture plane. There are many methods for vanishing point detection [13]. Some methods make use of line segments detected in images.

There are significantly large numbers of vanishing points present in an image. Therefore, the aim is to detect the vanishing point that correspond to the principal direction of a scene. In this paper, we can use the detected lane to find the vanishing point. In real scene, the left and right lane lines are parallel expect the curve lane. So we can use the detected lane to find the principal direction and the vanishing point. The intersection of left and right lane lines is the vanishing point.

3 Lane Detection Using Three-Stage Feature Extraction and RANSAC

This paper uses the lane proposals generated by three-stage feature extraction. This section details our whole framework for lane detection.

Learning or fitting based approach is the main method used for lane detection. In [12], the author has proposed the SVM based lane detection algorithm. In [14], authors have proposed the boosting algorithm to select relevant features

Algorithm 3. Convex Signal Detection

```
Require:
  Obtain contrast sequence $S_{contrast}$ using Algorithm 1
  Obtain line set $S_{line}$ using Algorithm 2
  Set threshold $N_{thresh}$, and sum range $LEN$
Ensure:
  Ultimate set $S_{proposal}$

  for each pixel$(x, y)$ in $S_{line}$ do
    Get the neighbourhood $S_n$ of the pixel and set $Count = 0$
    for each pixel in $S_n$ do
      if the pixel belongs to $S_{contrast}$ then
        $Count = Count + 1$
      end if
      if $Count > N_{thresh}$ then
        set $sum_L = 0$, $sum_M = 0$, and $sum_R = 0$
        for $c \in [1, LEN]$ do
          $Lp = (x - c, y)$ and $sum_L = sum_L + gray(Lp)$
          $Mp = (x + c - 1, y)$ and $sum_M = sum_M + gray(Mp)$
          $Rp = (x + LEN + c - 1, y)$ and $sum_R = sum_R + gray(Rp)$
        end for
        if $sum_M > sum_L$ and $sum_M > sum_R$ then
          save the pixel to the ultimate set $S_{proposal}$
        end if
      end if
    end for
  end for
```

used for lane detection. However, learning based methods require lots of well-labeled training data to ensure the good performance, which is hard or expensive to be collected. In this paper, we use the random sample consensus (RANSAC) method to fit straight lines. RANSAC is an iterative method to estimate parameters of a mathematical model from a set of observed data that contains outliers, which is much more stable than hough transform.

We first use the above method to extract lane feature proposals and then use RANSAC method to fit lines. At last, we find that we can make use of slope and intercept to make tracking and lane departure warning (LDW). In Fig. 6, we show an example result of lane detection.

4 Experimental Results

We have tested the proposed method in sequences which got in vehicle by different CMOS based cameras. In all cases the images are 640 × 480. We only process the region below the red line in order to reduce computation amount. In all sequences, we set those parameters of algorithms with same value. In Algorithm 1, we set $T_{min} = 3$, $H_{range} = 15$, and $Length = 15$. In Algorithm 2, we set

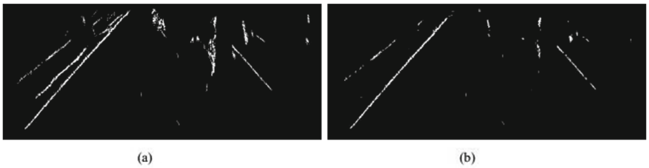

Fig. 5. Result of Algorithm 3. (a) is the result of Algorithm 2. (b) is the result of Algorithm 3.

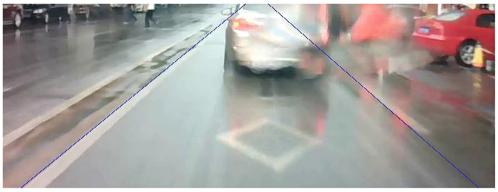

Fig. 6. Lane detection results. The blue line is the detected lane. (Color figure online)

$H = 5$, $DistThresh = 5$, and $N_{thresh} = 5$. In Algorithm 3, we set $N_{thresh} = 8$ and $LEN = 15$. And we mainly evaluate the proposed detection algorithm on daylight and rainy day.

A major problem in acquiring quantitative results is the lacking of ground truth in real scene. Thus we only show examples of challenging situations involving lanes in rainy day, shadows, and different lighting conditions etc.

In Fig. 7, we give some results of lane proposals detection. In these experiments, we respectively apply our algorithm to detect the left side proposals and the right side proposals. In (a), (b), (d) and (e), even in the presence of variant lighting conditions, the proposed method is able to find lane proposals exactly. However, we can see that some noise results also appear because of the interference of the windshield. In (c), we can see that the method is able to find lane proposals in the presence of curve lane. In (f)−(i) and (l), in the rainy day, lanes are very blurry. The method also can find lane proposals exactly though there exits some noise in the results. In (j) and (k), the road is full of noises and the lanes are also very blurry. We can see that this method also can find lane proposals exactly. If we use learning based lane detection, these proposals could reduce search range and accelerate feature extraction. In practice, we have tested this method in lots of scenes and found that if there exits lane in scene, our method can find lane proposals exactly.

In Fig. 8, we show some results of lane detection with the method of RANSAC based on the results of Fig. 8. Note that, in these experiments, we only fit the left side proposals. In (a), (b), (d) and (e), even in the presence of noise, the method could fit line exactly. In (g)−(i) and (l), in the rainy day, the lane is very blurry and this method could also fit it very exactly. In (j) and (k), the results are also very exactly. We find that if the lane proposals are detected exactly, we can fit the lane exactly. What's more, we add some strategies based on slope and intercept for tracking and lane departure warning.

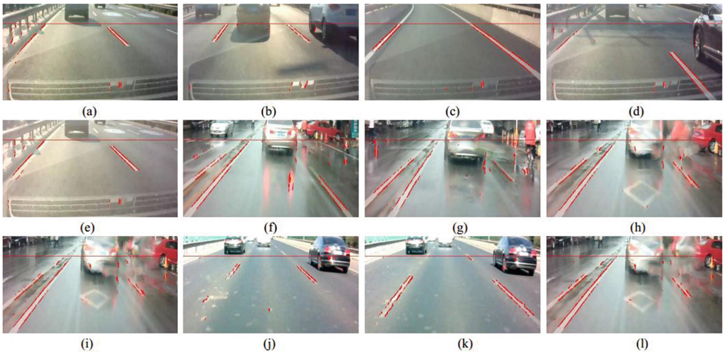

Fig. 7. Lane proposals detection. There are many scenes in the figure such sunny day, rainy day etc.

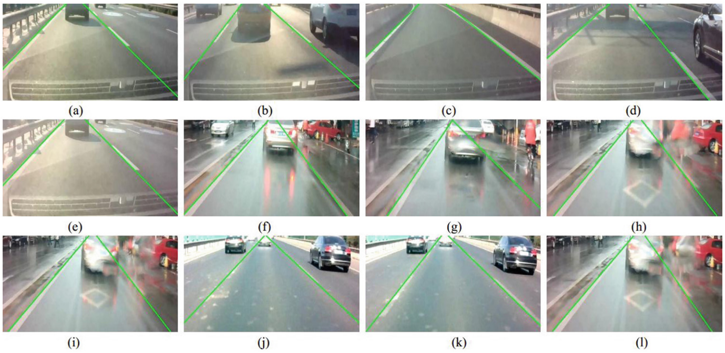

Fig. 8. Lane detection results under some challenge scenarios.

Since the proposed method does not have very complex operations, we can easily transplant the algorithm to the embedded platform, and require a small amount of computing resources.

5 Conclusion

A novel framework for detecting lane proposals from one single image is proposed, which is based on three-stage feature extraction. With the processing of this method, we can get lane proposals. Then we use these proposals to fit lines with RANSAC.

We have made extensive experiments and found this method is very stable. If there exits lane in a scene, the method can find lane proposals.

Acknowledgments. This work was supported by the NSFC (under Grant U1509206, 61472276).

References

1. Gujanatti, R., Tigadi, A., Gonchi, A.: Advanced driver assistance systems. Int. J. Eng. Res. Gen. Sci. **4**, 151–158 (2016)
2. Kumar, A.M., Simon, P.: Review of lane detection and tracking algorithms in advanced driver assistance system. Int. J. Comput. Sci. Inf. Technol. **7**, 65–78 (2015)
3. Mingliang, X., Lv, P., Niu, J., Lu, J., Zhao, X.: Robust lane detection using two-stage feature extraction with curve fitting. Pattern Recognit. **59**, 225–233 (2016)
4. Canero, C., Lopez, A., Serral, J., et al.: Detection of lane markings based on ridgeness and RANSAC. In: IEEE Conference on Intelligent Transportation Systems, pp. 13–16 (2005)
5. Canero, C., Lopez, A., Serral, J., et al.: Robust lane markings detection and road geometry computation. Int. J. Automot. Technol. **11**, 395–407 (2009)
6. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. In: ACM, vol. 6, pp. 381–395 (1981)
7. Aly, M.: Real time detection of lane markers in urban streets. In: Intelligent Vehicles Symposium (2008)
8. Teoh, E.K., Wang, Y., Shen, D.: Lane detection using spline model. In: Image and Vision Computing, pp. 677–689 (2000)
9. Teoh, E.K., Wang, Y., Shen, D.: Lane detection and tracking using B-snake. In: Image and Vision Computing, pp. 269–280 (2003)
10. Verl, A., Beyeler, M., Mirus, F.: Vision-based robust road lane detection in urban environments. In: ICRA (2014)
11. Alvarez, J.M.A., Lopez, A.M.: Road detection based on illuminant invariance. In: IEEE Conference on Intelligent Transportation Systems, vol. 12, pp. 184–193 (2008)
12. Kim, Z.: Robust lane detection and tracking in challenging scenarios. In: IEEE Conference on Intelligent Transportation Systems, vol. 9, pp. 16–26 (2008)
13. Audibert, J., Ponce, J., Kong, H.: Vanishing point detection for road detection. In: CVPR, pp. 96–103 (2009)
14. Shneier, M., Gopalan, R., Hong, T., Chellappa, R.: A learning approach towards detection and tracking of lane markings. In: IEEE Conference on Intelligent Transportation Systems, vol. 13 (2012)