2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Using DP Towards A Shortest Path Problem-Related Application

Jianhao Jiao[1], Rui Fan[1], Han Ma[2], Ming Liu[1]

Abstract— **The detection of curved lanes is still challenging for autonomous driving systems. Although current cutting-edge approaches have performed well in real applications, most of them are based on strict model assumptions. Similar to other visual recognition tasks, lane detection can be formulated as a two-dimensional graph searching problem, which can be solved by finding several optimal paths along with line segments and boundaries. In this paper, we present a directed graph model, in which dynamic programming is used to deal with a specific shortest path problem. This model is particularly suitable to represent objects with long continuous shape structure, e.g., lanes and roads. We apply the designed model and proposed an algorithm for detecting lanes by formulating it as the shortest path problem. To evaluate the performance of our proposed algorithm, we tested five sequences (including 1573 frames) from the KITTI database. The results showed that our method achieves an average successful detection precision of 97.5%.**

I. INTRODUCTION

A. Motivations

The major bottleneck in the development of autonomous driving systems is perception problem [1], [2]. As one of the most essential tasks in perception, traffic scene understanding consists of several computer vision tasks including lane and obstacle detection. The functions of lane detection systems are to recognize and localize lane markings on roads as well as provide drivers with a traversable area. Consequently, these systems are essential to ensure traffic safety and enhance the driving experience in autonomous vehicles. However, lane detection is considered as a demanding task when the actual environment and road conditions such as occlusion, intersections, lights, and weather are considered.

Several state-of-the-art approaches [3], [4] have achieved excellent performance in real applications. However, most of them are based on precise vanishing point (VP) estimation or a specific road model assumption. They do not perform well in challenging environments, such as crossings and turnings, where multiple VPs at different positions exist.

Inspired by [5]–[7], similar to these visual recognition tasks, lane detection can also be formulated as a two-dimensional graph searching problem where lane boundaries are extracted through an optimal path search. Hence, the detection of lanes does not need to rely on the strict model assumption. However, finding the best graph model and corresponding shortest path searching algorithm for the given problem is very subtle.

In this paper, we present a directed graph model, in which dynamic programming (DP) is used to solve a specific shortest path problem (SPP). Through a simple pre-processing module, lane boundaries are represented by the designed graph model. We then formulate lane detection as an SPP and solved it effectively by implementing the proposed approach, in which lanes can be localized via the shortest path search.

B. Contributions

The contributions of this paper include:

- We introduce a graph model where a specific SPP is defined. This graph can intuitively model several objects with a long continuous shape structure e.g., roads.
- We analyze the properties of this graph and use DP to solve the well-defined SPP.
- We formulate lane detection as an SPP with this graph model, and then apply the proposed DP-based algorithm to detect multiple lanes in real environments.

C. Paper Organization

The rest of the paper is organized into the following sections: In Section II, several related work are discussed. In Section III, a general shortest path search problem (SPSP) is presented as well as the corresponding solutions including DP are evaluated using a set of generic test data. Mathematical preliminaries are provided in Section IV. The application of the proposed algorithm for detecting lanes is introduced in Section V, followed by the experimental results in Section VI. Finally, Section VII summarizes the paper and discusses several possible future works.

II. RELATED WORK

A. Lane Detection

The existing lane detection approaches are mainly categorized as feature-based and model-based. The former uses handcraft features, e.g., edges or gradient, to segment lanes boundaries from images, while the latter assumes that the appearance of lanes can be fitted using mathematical models.

The linear [8], parabolic [9], [10], and spline models [11] are commonly used in representing lanes structure. The linear model performs well in modelling straight lines, while the parabolic and spline models are more flexible when lanes have high curvature. However, more parameters introduced into the optimization process will make the computational complexity higher. Therefore, we use the parabolic model in

This work was supported by the National Natural Science Foundation under the project No. U1713211, and the Research Grant Council of Hong Kong SAR Government, China, under Project No. 11210017 and No. 21202816, awarded to Prof. Ming Liu.

[1] Jianhao Jiao, Rui Fan and Ming Liu are with the Robotics and Multi-Perception Laborotary, Robotics Institute, The Hong Kong University of Science and Technology, Hong Kong SAR, China. {jjiao, eeruifan, eelium}@ust.hk

[2] Han Ma is with the Department of Precision Instrument, School of Mechanical Engineering, Tsinghua University, Beijing, China. mh15@mails.tsinghua.edu.cn

978-1-5386-6027-0/19/$31.00 ©2019 IEEE

this paper, which is effective to represent lanes on the inverse perspective mapping (IPM) images.

Produced by the perspective projection, VP is a frequently used model to compute the intersections of lanes on images. Traditionally, VP was estimated directly on images by applying Hough transform [12] or texture extraction [13] techniques. However, these approaches usually fail in several actual environments. For instance, the accuracy of VP estimation will be greatly reduced on complex roadways. Therefore, stereo vision technologies [14], [15] were introduced to address this problem. Ozgunalp et al. [15] proposed an automatic VP estimation pipeline with a two-step process: firstly, the disparity is estimated from stereo images, and secondly, a corresponding disparity accumulator is computed. Multiple VPs of parallel lanes are found using this accumulator. But different from the above approaches which rely exclusively on VP, our method only uses it to create the IPM images.

As for the feature-based methods, several approaches proposed in [10], [16]–[18] are based on the edge features. Besides, the gradient is another useful feature, which can be used to evaluate the stretch of lanes [8], [19]. Generally, most of the feature-based methods require clear and distinct lane appearance to process, which are sensitive to occlusions and shadow. In our method, we adopt the robust Canny operator [20] to extract lane boundaries.

Recently, deep learning methods have shown great success in several visual recognition problems [21]–[23]. Kim et al. [24] first explored the performance of convolutional neural network (CNN) for lane detection. Lee et al. [4] presented a multi-task learning network: VPGNet, for road markings detection, classification, and VP prediction. For successfully detecting lanes at a pixel level, several networks for instance segmentation were proposed in [25], [26]. On the other side of the spectrum, the spatial CNN was demonstrated to be accurate in road markings segmentation in [27]. However, the main limitation is that these methods require a large number of correct manual annotations and much training time.

B. Visual Recognition Based on Shortest Path Problem

The lane detection problem stated in this paper can be formulated as a graph searching problem, which is degraded as SPP. As a general problem, the SPP has been applied in many fields, including path planning [28], computer vision [29], and network flow [30]. By constructing a graph from raw data, problems can be solved efficiently by applying a general shortest path searching algorithm, such as A* search [31], the Dijkstra's algorithm [32], the Floyd-Warshal algorithm [33] or the ant colony optimization algorithm [34]. Specifically, SPPs were shown to have great potential in solving visual recognition problems [5]–[7], [35]. Mortensen et al. [5] presented a DP-based object boundary searching algorithm for image composition, while Meng et al. [6] employed DP to improve the accuracy of objects segmentation. WIth the stereo vision technologies, Zhang et al. [35] proposed a VP constrained model, where the Dijkstra algorithm was implemented to detect road boundaries. However, most of these approaches only focus on the solution of single-pair shortest path problem, which does not apply to our cases.

III. SHORTEST PATH SEARCHING BASED ON DP

In this section, we introduce a specific graph model and show how to use DP to solve an SPP. The results in Section III-C show that our proposed method can general solve several visual recognition problems, such as the detection of objects with a long continuous shape.

A. Problem Statement

We consider a specific directed acyclic graph (DAG) model with a matrix structure. As a convention, the designed graph model is denoted by $\mathcal{G} = \{\mathcal{P}, \mathcal{E}\}$, and its dimension is set as $U \times V$. The nodes are denoted by their positions (u, v), and the directed edges are denoted by their starting/ending node like this: $[(s,t),(i,j)]$. In addition, we denote $\mathcal{D} = \{(1,V),(2,V),\dots,(U,V)\}$ the set of nodes on the top row. We also denote $\mathcal{F}[(i,j),(s,t)]$ the cost of $[(s,t),(i,j)]$. Two important properties of $\mathcal{G}$ are summarized as below:

- $\mathcal{G}$ is directed, $[(i,j),(s,t)]$ and $[(s,t),(i,j)]$ refer to two different edges.
- $\mathcal{G}$ has a bottom-up connection structure, which means that each node has a connection with only $2k+1$ neighbors which are located in its next row.

An example of $\mathcal{G}$ is depicted in Fig. 1. With assumptions and notations made, we can formulate an SSP as follows:

Problem: Given a well-defined graph model $\mathcal{G}$ described above and a set of nodes $\mathcal{D}$, we need to find shortest paths from the bottom row to the destination nodes in $\mathcal{D}$. Please note here, *shortest* is defined that a path should have the minimum cumulative cost.

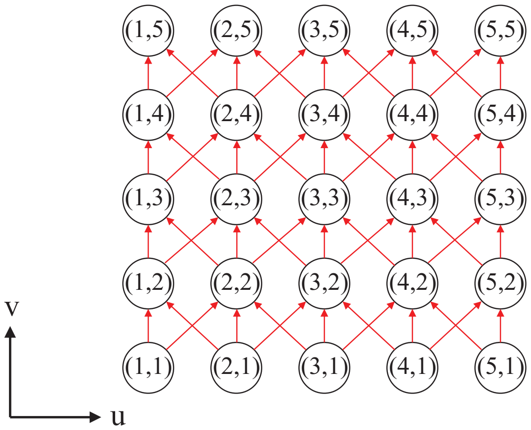


Fig. 1. The structure of a 5×5 graph ($k = 1$).

SPPs have several variants [36]: single-pair shortest path problem (SPSPP), single-source shortest path problem (SSSPP), single-destination shortest path problem (SDSPP) and all-pairs shortest path problem (APSPP). For each variation, the corresponding solutions have been proposed, e.g., the Dijkstra's algorithm and the Floyd-Warshall algorithm. The problem we aim to solve here is an SPP. Thus, we can also use the above algorithms to solve it. However, several differences in the formulated problem should be considered when compared with APSPP and SDSPP:

- Our problem is the degradation of APSPP. We only need to find U shortest paths respectively for any node on the bottom row to each node in $\mathcal{D}$.
- Our problem is an extension of SDSPP with U destination nodes at the same time.

In contrast, DP is more effective since this problem possesses the following properties:

- Optimal substructure: the solution can be obtained by the combination of optimal solutions to its subproblems: finding the shortest paths between two consecutive rows.
- Overlapping subproblems: the problem has a recursive formulation; the space of its sub-problems is small so that the solutions of the subproblems can be stored in a table to avoid recomputing.

B. Algorithm Description

$E(u,v)$ is denoted as the minimum cumulative cost function, which indicates the cost of the shortest path when it reaches (u,v). $P(u,v)$ is denoted as the position of the shortest path. We can write down an equation to recursively find the optimal $P^*(u,v)$ use the results to update $E(u,v)$:

$$\begin{aligned} P^*(u,v) = \underset{P(u,v)}{\arg\min}\Big\{ & E\big[P(u,v)\big] \\ & +\mathcal{F}\big[P(u,v),(u,v)\big]\Big\} \qquad (1) \\ \text{s.t.} \quad & P(u,v) = (u-j,v) \\ & j \in \{-k,-k+1,\dots,k\}, \end{aligned}$$

$$E(u,v) = E\big[P^*(u,v)\big] + \mathcal{F}\big[(u,v),P^*(u,v)\big]. \qquad (2)$$

With (1) and (2), the SPP can be solved by utilizing DP. The detailed implementation is summarized in Algorithm 1.

Algorithm 1 The DP algorithm

```
Input: $\mathcal{P}$, $\mathcal{D}$, $\mathcal{F}$
Output: $P$
Initialization: $E(\cdot,1) = 0$, other terms are set to $+\infty$
Initialization: $P(u,v)$
for $v = 2:V,\ u = 1:U,\ j = -k:k$ do
    if $E(u,v) > E(u-j,v-1)+\mathcal{F}\big[(u,v),(u-j,v-1)\big]$ then
        $E(u,v) = E(u-j,v-1)+\mathcal{F}\big[(u,v),(u-j,v-1)\big]$
        $P(u,v) = (u-j,v-1)$
    end if
end for
```

C. Validation

Fig. 2 shows the shortest path results in a toy example estimated by the DP algorithm. To demonstrate the usage of DP in the specific graph model in terms of time and space complexity, we compare it with two general shortest path

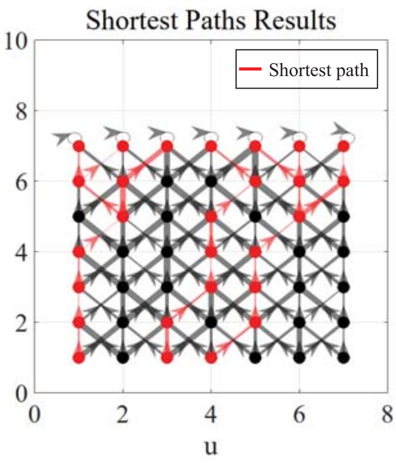


Fig. 2. The results of the formulated problem in a 7×7 graph are visualized. The shortest paths are searched by the proposed DP algorithm. The red edges represent the shortest paths, while the black one are the normal edges. The thickness of each edge has a linear relationship with its cost.

algorithms: the Dijkstra's algorithm (implemented with the Fibonacci heap) and the Floyd-Warshall algorithm. Denoting $|\mathcal{P}|$ the number of nodes, and $|\mathcal{E}| \approx k|\mathcal{P}|$ the number of edges, we can use Table I to compare the complexity of among these algorithms.

TABLE I

TIME COMPLEXITY AND SPACE COMPLEXITY COMPARISON OF THE THREE SHORTEST PATH SEARCHING ALGORITHMS.

Algorithm	Time Complexity	Space Complexity
Proposed DP	$O(\lvert\mathcal{P}\rvert)$	$O(\lvert\mathcal{P}\rvert)$
Dijkstra	$O(\lvert\mathcal{P}\rvert\log\lvert\mathcal{P}\rvert)$	$O(\lvert\mathcal{P}\rvert)$
Floyd-Warshall	$O(\lvert\mathcal{P}\rvert\lvert\mathcal{E}\rvert)$	$O(\lvert\mathcal{P}\rvert^2)$

D. Reasoning

According to the time complexity and space complexity, the proposed DP algorithm is better than the others on the designed graph model. On the one hand, it takes advantages of the connection structure of the graph model; hence the search space is significantly reduced. On the other hand, DP is only applicable to problems with specific properties. Furthermore, as a typical memory intensive algorithm, DP is ineffective in a large-scale graph model. However, considering the real applications where the graph size is usually small, meaning that our proposed algorithm is feasible in several image segmentation tasks.

IV. PRELIMINARIES

In this section, we provide readers with mathematical preliminaries regarding VP and IPM.

1) Vanishing Point: The VP is produced by the perspective projection, where the projection in the image plane Π of any pair of straight and parallel lines in $\mathbb{R}^3$ will coverage at one point. Fig. 3 visualizes the mentioned process. We denote L a straight line in $\mathbb{R}^3$ and $\mathbf{P}_0 = [P_X, P_Y, P_Z, 1]^\top$ a

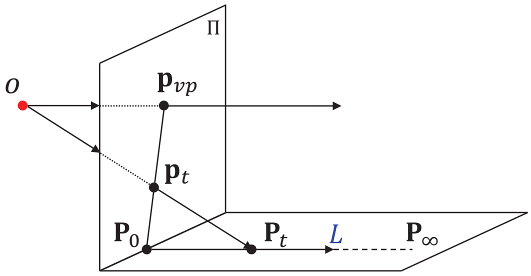


Fig. 3. The vanishing point

point in a camera cooridinate system (CCS). Therefore, the coordinates of a point $\mathbf{P}_t$ on L are parameterized as follows:

$$\mathbf{P}_t = \mathbf{P}_0 + t\mathbf{n} = \begin{bmatrix} P_X + tn_X \\ P_Y + tn_Y \\ P_Z + tn_Z \\ 1 \end{bmatrix} \cong \begin{bmatrix} \frac{P_X}{t} + n_X \\ \frac{P_Y}{t} + n_Y \\ \frac{P_Z}{t} + n_Z \\ \frac{1}{t} \end{bmatrix}, \quad (3)$$

$$\mathbf{p}_t = \mathbf{K}\mathbf{P}_t, \quad (4)$$

where $\mathbf{n} = [n_X, n_Y, n_Z]^\top$ is the unit vector giving the direction of L, t is a scalar, $\mathbf{p}_t = [u, v, 1]^\top$ is the projection of $\mathbf{P}_t$ in Π, and $\mathbf{K}$ is a camera intrinsic matrix.

When $t \to \infty$,

$$\mathbf{P}_\infty \cong \begin{bmatrix} n_X \\ n_Y \\ n_Z \\ 0 \end{bmatrix}, \quad (5)$$

$$\mathbf{p}_{vp} = \mathbf{p}_\infty = \mathbf{K}\mathbf{P}_\infty, \quad (6)$$

where $\mathbf{P}_\infty$ is a point of L at infinity and $\mathbf{p}_{vp} = [u_{vp}, v_{vp}]^\top$ is the VP. With similar derivation, we conclude that the projection of any parallel line to L will intersect at $\mathbf{p}_{vp}$.

2) Inverse Perspective Mapping: IPM has been widely used for understanding road traffic signs information [37]. Detailed derivation and generation of the IPM image are described in [38]. Fig. 4 depicts the relationship between the world coordinate system (WCS), CCS and image coordinate system (ICS). Denoting $\mathbf{P} = [X, Y, Z]^\top$ a point in the WCS and $\mathbf{p} = [u, v]^\top$ the corresponding point in the ICS.

The transformation equation from $\mathbf{P}$ to $\mathbf{p}$ is given:

$$\begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \sim \mathbf{K}[\mathbf{R}|\mathbf{t}] \begin{bmatrix} \mathbf{P} \\ 1 \end{bmatrix}, \quad (7)$$

where $\mathbf{R} = \mathbf{R}_Z(\gamma)\mathbf{R}_Y(\beta)\mathbf{R}_X(\alpha)$.

Without loss of generality, we set $\mathbf{t} = [0, 0, -h]^\top$, where h is the height of the camera. We also set $\alpha = 0$. If $\mathbf{p}_{vp}$ is acquired, β and γ can be calculated as in [37]:

$$\beta = \arctan\left[(\tan\alpha_v)(1 - \frac{2v_{vp}}{V})\right], \quad (8)$$

$$\gamma = \arctan\left[(\tan\alpha_u)(\frac{2u_{vp}}{U} - 1)\right], \quad (9)$$

where α_v and α_v are half of the vertical and horizontal angle of a camera respectively, and $V \times U$ denotes the image size.

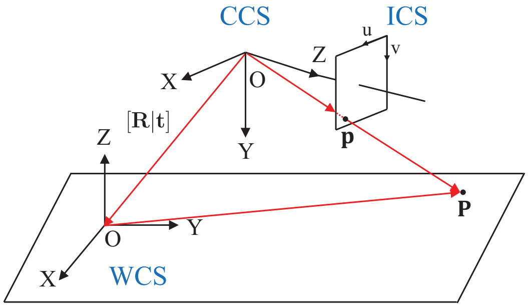


Fig. 4. The relationship between the world coordinate system (WCS), camera coordinate system (CCS) and image coordinate system (ICS).

V. METHODOLOGY

In this section, we discuss how the proposed model can be applied to detect lanes on images. Fig. 5 shows the pipeline of the proposed methods. The left and right images captured by stereo cameras are denoted by I_l and I_r respectively, the IPM image produced from I_l is denoted by I_i, and the VP for I_l is denoted by $\mathbf{p}_{vp}$.

In the pre-processing module, a graph model $\mathcal{G}$ is constructed from I_i using edge information, as depicted in Fig. 6. Then, several iterations are executed until the desired number of lanes are found in the path searching and model fitting modules. At each separate iteration, we employ DP to find the shortest path in $\mathcal{G}$ for each node in $\mathcal{D}$. After that, we select one path with the minimum cumulative cost. Its correctness will be checked through a parabolic model fitting. Finally, the resulting models are projected to I_l for visualization.

A. Pre-processing

1) Vanishing Point Estimation: VP is estimated using our previously published method [39], where two dense accumulators are optimized to extract the desirable paths with the minimum cost. The extracted paths are then interpolated into a quadratic and quadruplicate polynomial, respectively. The VP can be computed using these two polynomials.

2) Edge Detection: The Canny detector is used to find the lane boundaries information, which includes a multi-stage process to detect a wide range of edges in images. It takes I_i as input, and returns a binary feature function $f_e(u, v)$, where u, v indicate the coordinates of a pixel and the pixel value is either 0 or 1. Specifically, we define that the pixel value of an edge is 1.

3) Graph Construction: Using the color, edge, and spatial information of pixels, the original image I_i can be easily modelled by the designed graph model $\mathcal{G}$. For formulating lane detection as an SPP, the edge cost should be defined appropriately. Since lanes are typically assumed to be painted in white or yellow in high contrast, we can intuitive utilize the edge and gray value to calculate the edge cost. Before defining $\mathcal{F}(\cdot, \cdot)$, we introduce $f(u, v)$, which is computed as a weighted sum of two component feature functions as:

$$\begin{aligned} w_e \cdot f_e(u, v) + w_g \cdot f_g(u, v) = f(u, v) \\ \text{s.t.} \quad w_e + w_g = 1, \end{aligned} \quad (10)$$

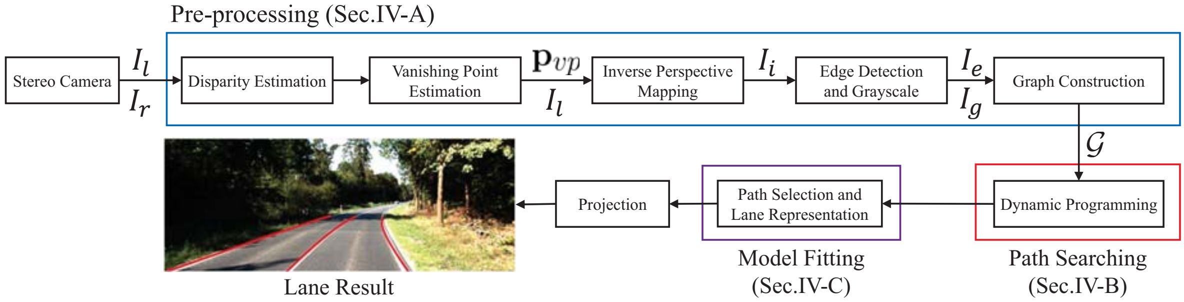
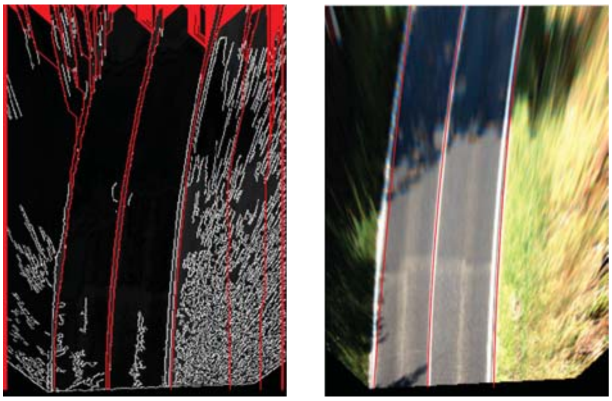


Fig. 5. Pipeline of the proposed multiple lanes detection method.



(a) Shortest paths in the graph (b) Lane detection result

Fig. 6. Pre-processing: (a) The Shortest paths in the graph. (b) Lane detection results.

where w_e and w_g are two scalars. The gray feature function $f_g(u,v)$ is defined as the normalization of I_g. Hence, the value $f(u,v)$ should be high if (u,v) is located at lanes.

With the above discussion, the value of $\mathcal{F}\left[(i,j),(s,t)\right]$ is defined as $[1-f(s,t)]$. During the process of shortest path searching, the paths will tend to pass through the lane areas.

B. Path Seaching Based on DP for lane detection

We apply the DP algorithm to a specific application: lane detection. In our implementation, we improve (1) with an additional regularization term λj^2:

$$\begin{aligned} P^*(u,v) = \underset{P(u,v)}{\arg\min} \Big\{ & E\big[P(u,v)\big] \\ & +\mathcal{F}\big[P(u,v),(u,v)\big] + \lambda j^2 \Big\} \qquad (11) \\ \text{s.t.} \quad & P(u,v) = (u-j,v) \\ & j \in \{-k,-k+1,\ldots,k\}, \end{aligned}$$

where λ is a scalar that is defined as a sum of the square of horizontal movement. We can observe that noisy data are randomly distributed, but lane boundaries are connected continuously with strong spatial relationship. For this reason, we design λj^2 to regulate the path generation in horizontal direction. On the other hand, the setting of λ may influence the structure of the resulting paths, i.e. we can correspondingly set a large λ to detect curved lanes.

The improved method for lane detection is detailed in Algorithm 2. Lines 10–14 will be explained in Section V-C.

C. Path Selection and Lane Representation

To enhance the detection accuracy, a model fitting module is developed. In this paper, we utilize a parabolic model $f(v)=\beta_2 v^2+\beta_1 v+\beta_0$ to fit the paths, which is sufficient to represent lanes in the IPM images. We use $\mathbf{P}$ to indicate the path with the minimum cumulative cost, which is a $V\times 2$ matrix. The i^{th} column of $\mathbf{P}$ is denoted by $[u_i,v_i]^\top$. Consequently, the parameter vector $\boldsymbol{\beta}=[\beta_2,\beta_1,\beta_0]^\top$ can be estimated by solving a least-squares problem:

$$\boldsymbol{\beta}^* = \underset{\boldsymbol{\beta}}{\arg\min} \sum_{i=1}^{V} \Big[u_i - (\beta_2 v_i^2 + \beta_1 v_i + \beta_0)\Big]^2. \qquad (12)$$

We employ random sample consensus (RANSAC) [40] to update the parameter vector $\boldsymbol{\beta}$ iteratively. To detemine whether a given candidate $[u_i,v_i]^\top$ is an inlier, the corresponding square residual $r_i=[u_i-f(v_i)]^2$ is computed. n_{inlier} will be updated if $r_i<t_r$.

Algorithm 2 Improved DP for Lane Detection

```
Input: $\mathcal{P}$, $\mathcal{D}$, $\mathcal{F}$, $\lambda$,
Output: $\mathcal{L}$
1:  for $v=2:V,\ u=1:U,\ j=-k:k$ do
2:      if $E(u,v) > E(u-j,v-1)+\mathcal{F}\big[(u,v),(u-j,v-1)\big]$
3:                                  $+\lambda j^2$
4:      then
5:          $E(u,v) = E(u-j,v-1)+\mathcal{F}\big[(u,v),(u-j,v-1)\big]$
6:                                  $+\lambda j^2$
7:          $P(u,v) = (u-j,v-1)$
8:      end if
9:  end for
10: while maximum-value$\big(E(\cdot,V)\big) > \epsilon$ do
11:     $i$ = maximum-value-index$\big(E(\cdot,V),P(\cdot,V)\big)$
12:     $\mathbf{P}$ = shortest-path$\big(i\big)$
13:     $\boldsymbol{\beta}$ = model-fitting$\big(\mathbf{P}\big)$
14:     if $\mathbf{P}$ has few outliers then
15:         $\mathcal{L} = \mathcal{L} \cup \{\boldsymbol{\beta}\}$
16:     end if
17:     $E(i,V) = 0$
18: end while
```

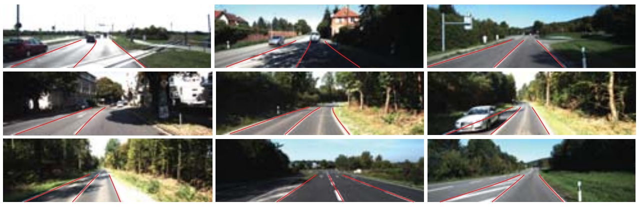


Fig. 7. Experimental results in various scenarios, where sample detection results are demonstrated under dense shadows and high exposure. The lines in red are detected lanes.

VI. Experiments

The proposed algorithm is compared with another published lane detection method [41] (except for tracking step). In this method, lanes are detected based on VP estimation. All the experiments are evaluated with the KITTI dataset. With empirically setting the parameters $w_e = 0.6$, $w_g = 0.4$, $\lambda = 2$, $k = 3$, $t_r = 5$, our method can obtain promising results. The IPM range covers a ground area of 30 m and 12 m in the Y and X directions, respectively. Lanes will be detected if they are located in this range.

A. Results

We test the accuracy of the proposed algorithm using the KITTI dataset [42]. To evaluate its performance, we select five sequences from the KITTI dataset : 1573 frames containing 3146 lanes from the *Road Category*[1] of the KITTI dataset. The lane detection results of the proposed algorithm and that presented in [41] are detailed in Table II and Table III. The correctness of lanes are manually checked, and only the detection of the closest two lanes are counted. In Table II, the proposed algorithm presents a better precision, where 99% lanes are successfully detected in the first three sequences, and the average precision in all sequences is 97.5%, while the precision of [41] only reaches 94.5%. Several successful detection results are shown in Fig. 7.

In our experiments, misdetections and incorrect detections of lanes induce several failed cases. The misdetections are mainly caused by camera over-exposure, partially occluded by obstacles and highly curved lanes. The detailed examples are presented in Fig. 8(a)–8(c) respectively. In Fig. 8(a), due to over-exposure, lanes very obscure. In Fig. 8(b), we can see the grass partially occludes several lanes. The occlusion decreases the edge features extracted by the Canny operator. In Fig. 8(c), all lanes have a high curvature, making the cumulative value of λj^2 large.

For the factors leading to incorrect detections, we group them into two main categories. The first category is that some objects, like road boundaries or railings, look very similar to lanes in the IPM images and sometimes fail the proposed algorithm. This case can be seen in Fig. 8(d). The other one is the shadow caused by vehicles or trees. In this case, the edges of the shadow cannot be removed cleanly, which may also fail the proposed algorithm. Fig. 8(f)-8(e) shows two examples within this category.

TABLE II

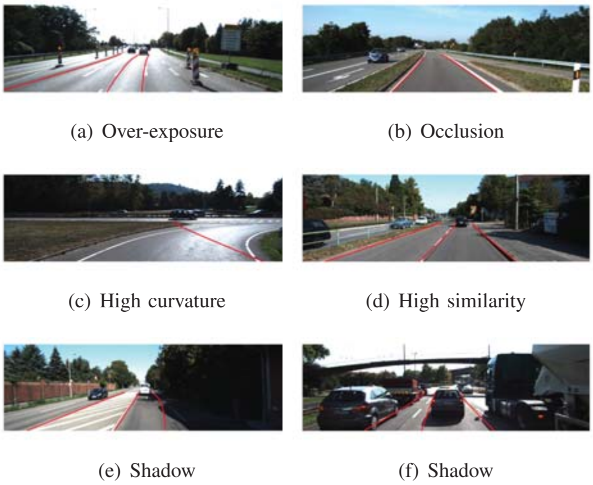
Detection results of the proposed algorithm.

Sequence	Total Lane markings	Incorrect detection	Misdetection	Precision
0926_0015	594	4	1	**99.2%**
0926_0027	376	12	2	**96.3%**
0926_0028	860	1	0	**99.9%**
0926_0070	760	29	0	**96.2%**
0930_0016	556	21	7	**95.0%**
Total	3146	67	10	**97.5%**

TABLE III

Detection results of [41].

Sequence	Total Lane markings	Incorrect detection	Misdetection	Precision
0926_0015	594	44	0	**92.6%**
0926_0027	376	44	0	**88.3%**
0926_0028	860	12	0	**98.6%**
Total	1830	100	0	**94.5%**



(a) Over-exposure

(b) Occlusion

(c) High curvature

(d) High similarity

(e) Shadow

(f) Shadow

Fig. 8. (a)–(c) illustrates the failed examples in misdetections, and (d)–(f) shows another failed examples in incorrect detections.

VII. Conclusion and Future Work

In this paper, we proposed a directed graph model and a corresponding shortest path searching algorithm based on DP that can tackle a specific SPP effectively. We also presented its application to lane detection. The results showed that the designed model and the proposed DP-based algorithm with great potential in solving visual recognition problems.

We observe that the proposed algorithm relies highly on the pre-processing results, and is thus sensitive to the changing light, occlusion and the similar appearance of other objects. But the main advantage of the proposed method is its flexibility to be a versatile solution of other visual recognition tasks. There are several possible extensions to this work, (1) finding a way to determine the start node and destination node of a lane, (2) fusing the stereo IPM images to construct an images in a broader range from bird's eye view, (3) detecting multiple lanes at crossings and turnings, where VPs in different directions exist.

[1] `http://www.cvlibs.net/datasets/kitti/raw_data.php?type=road`

References

[1] C. Thorpe, M. Herbert, T. Kanade, and S. Shafer, "Toward autonomous driving: The cmu navlab. part i-perception," *IEEE Intelligent Systems*, no. 4, pp. 31–42, 1991.

[2] M. Liu, F. Colas, F. Pomerleau, and R. Siegwart, "A markov semi-supervised clustering approach and its application in topological map extraction," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 4743–4748.

[3] J. J. M. U. M. B. R. F. Han Ma, Yixin Ma, "Multiple lane detection algorithm based on optimised dense disparity map estimation," *2018 IEEE International Conference on Imaging Systems and Techniques (IST)*, 2018.

[4] S. Lee, J. Kim, J. S. Yoon, S. Shin, O. Bailo, N. Kim, T. H. Lee, H. S. Hong, S. H. Han, and I. S. Kweon, "VPGNet: Vanishing Point Guided Network for Lane and Road Marking Detection and Recognition," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2017-Octob, no. October, pp. 1965–1973, 2017.

[5] E. N. Mortensen and W. A. Barrett, "Intelligent scissors for image composition," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 1995, pp. 191–198.

[6] F. Meng, H. Li, G. Liu, and K. N. Ngan, "Object co-segmentation based on shortest path algorithm and saliency model." *IEEE Trans. Multimedia*, vol. 14, no. 5, pp. 1429–1441, 2012.

[7] P. Yan and A. A. Kassim, "Medical image segmentation using minimal path deformable models with implicit shape priors," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 4, pp. 677–684, 2006.

[8] R. Fan, "Real-time computer stereo vision for autonomous applications," Ph.D. dissertation, 07 2018.

[9] C. Kreucher and S. Lakshmanan, "LANA: A lane extraction algorithm that uses frequency domain features," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 343–350, 1999.

[10] Y. Zhou, R. Xu, X. Hu, and Q. Ye, "A robust lane detection and tracking method based on computer vision," *Measurement science and technology*, vol. 17, no. 4, p. 736, 2006.

[11] Y. Wang, E. K. Teoh, and D. Shen, "Lane detection and tracking using b-snake," *Image and Vision computing*, vol. 22, no. 4, pp. 269–280, 2004.

[12] B. Fardi and G. Wanielik, "Hough transformation based approach for road border detection in infrared images," in *Intelligent Vehicles Symposium, 2004 IEEE*. IEEE, 2004, pp. 549–554.

[13] H. Kong, J.-Y. Audibert, and J. Ponce, "Vanishing point detection for road detection," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 96–103.

[14] R. Fan and N. Dahnoun, "Real-time stereo vision-based lane detection system," *Measurement Science and Technology*, vol. 29, no. 7, p. 074005, 2018.

[15] U. Ozgunalp and N. Dahnoun, "Robust lane detection & tracking based on novel feature extraction and lane categorization," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 8129–8133.

[16] Y. Wang, L. Bai, and M. Fairhurst, "Robust road modeling and tracking using condensation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 4, pp. 570–579, 2008.

[17] R. Chapuis, R. Aufrere, and F. Chausse, "Accurate road following and reconstruction by computer vision," *IEEE transactions on intelligent transportation systems*, vol. 3, no. 4, pp. 261–270, 2002.

[18] 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.

[19] H.-Y. Cheng, B.-S. Jeng, P.-T. Tseng, and K.-C. Fan, "Lane detection with moving vehicles in the traffic scenes," *IEEE Transactions on intelligent transportation systems*, vol. 7, no. 4, pp. 571–582, 2006.

[20] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[21] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[23] P. Yun, L. Tai, Y. Wang, C. Liu, and M. Liu, "Focal loss in 3d object detection," *IEEE Robotics and Automation Letters*, pp. 1–1, 2019.

[24] J. Kim and M. Lee, "Robust lane detection based on convolutional neural network and random sample consensus," in *International Conference on Neural Information Processing*. Springer, 2014, pp. 454–461.

[25] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool, "Towards end-to-end lane detection: an instance segmentation approach," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 286–291.

[26] G. L. Oliveira, W. Burgard, and T. Brox, "Efficient deep models for monocular road segmentation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4885–4891.

[27] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial As Deep: Spatial CNN for Traffic Scene Understanding," 2017. [Online]. Available: http://arxiv.org/abs/1712.06080

[28] R. Valencia and J. Andrade-Cetto, "Path planning in belief space with pose slam," in *Mapping, Planning and Exploration with Pose SLAM*. Springer, 2018, pp. 53–87.

[29] J. Ulen, P. Strandmark, and F. Kahl, "Shortest paths with higher-order regularization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 12, pp. 2588–2600, 2015.

[30] R. K. Ahuja, *Network flows: theory, algorithms, and applications*. Pearson Education, 2017.

[31] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[32] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[33] S. Hougardy, "The floyd–warshall algorithm on graphs with negative cycles," *Information Processing Letters*, vol. 110, no. 8-9, pp. 279–281, 2010.

[34] M. Dorigo and T. Stützle, "Ant colony optimization: overview and recent advances," *Techreport, IRIDIA, Universite Libre de Bruxelles*, vol. 8, 2009.

[35] Y. Zhang, Y. Su, J. Yang, J. Ponce, and H. Kong, "When dijkstra meets vanishing point: A stereo vision approach for road detection," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2176–2188, 2018.

[36] K. Magzhan and H. M. Jani, "A review and evaluations of shortest path algorithms," *International journal of scientific & technology research*, vol. 2, no. 6, pp. 99–104, 2013.

[37] M. Nieto, L. Salgado, F. Jaureguizar, and J. Cabrera, "Stabilization of inverse perspective mapping images based on robust vanishing point estimation," in *Intelligent Vehicles Symposium, 2007 IEEE*. IEEE, 2007, pp. 315–320.

[38] H. A. Mallot, H. H. Bülthoff, J. Little, and S. Bohrer, "Inverse perspective mapping simplifies optical flow computation and obstacle detection," *Biological cybernetics*, vol. 64, no. 3, pp. 177–185, 1991.

[39] U. Ozgunalp, R. Fan, X. Ai, and N. Dahnoun, "Multiple Lane Detection Algorithm Based on Novel Dense Vanishing Point Estimation," vol. 18, no. 3, pp. 621–632, 2017.

[40] M. A. Fischler and R. 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.

[41] Y. Wang, N. Dahnoun, and A. Achim, "A novel system for robust lane detection and tracking," *Signal Processing*, vol. 92, no. 2, pp. 319–334, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.sigpro.2011.07.019

[42] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.