



**HAL**  
open science

## When Dijkstra meets vanishing point: a stereo vision approach for road detection

Yigong Zhang, Yingna Su, Jian Yang, Jean Ponce, Hui Kong

► **To cite this version:**

Yigong Zhang, Yingna Su, Jian Yang, Jean Ponce, Hui Kong. When Dijkstra meets vanishing point: a stereo vision approach for road detection. *IEEE Transactions on Image Processing*, In press, pp.1-12. hal-01678548

**HAL Id: hal-01678548**

**<https://hal.science/hal-01678548>**

Submitted on 9 Jan 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# When Dijkstra meets vanishing point: a stereo vision approach for road detection

Yigong Zhang<sup>†</sup>, Yingna Su<sup>†</sup>, Jian Yang<sup>†</sup>, *Member, IEEE* Jean Ponce<sup>‡</sup>, *Fellow, IEEE* and Hui Kong<sup>†</sup>, *Member, IEEE*

**Abstract**—In this paper, we propose a vanishing-point constrained Dijkstra road model for road detection in a stereo-vision paradigm. First, the stereo-camera is used to generate the u- and v-disparity maps of road image, from which the horizon can be extracted. With the horizon and ground region constraints, we can robustly locate the vanishing point of road region. Second, a weighted graph is constructed using all pixels of the image, and the detected vanishing point is treated as the source node of the graph. By computing a vanishing-point constrained Dijkstra minimum-cost map, where both disparity and gradient of gray image are used to calculate cost between two neighbor pixels, the problem of detecting road borders in image is transformed into that of finding two shortest paths that originate from the vanishing point to two pixels in the last row of image. The proposed approach has been implemented and tested over 2600 grayscale images of different road scenes in the KITTI dataset. The experimental results demonstrate that this training-free approach can detect horizon, vanishing point and road regions very accurately and robustly. It can achieve promising performance.

**Index Terms**—Road detection, vanishing point detection, stereo vision, Dijkstra model.

## I. INTRODUCTION

**T**HE problem of automatic road detection and following has been studied for decades in the context of autonomous ground vehicle development [1] and driver-assistance technologies [2], [3], [4].

In most of the current geometric approaches for road (border) detection in urban environments [7], [8], the road is usually fitted by various shape models based on road/lane borders, such as straight lines, piecewise linear segments [9], [10], clothoids [11], parabola [6], hyperbola [12], splines [13], [14], or snakes [14], [15]. These models can be classified as parametric methods.

Our paper also belongs to the category of geometric approaches for road (border) detection. But different from most of the above parametric ones, our method on road border detection is a non-parametric one (Fig.2). In general, our method is more accurate than the parametric ones because road borders cannot generally be represented by a parametric model when road region is partially occupied by vehicles or pedestrians. Fig.2 illustrates these scenarios.

Specifically, we propose a stereo-vision based road detection paradigm. It relies on the disparity map computed from the stereo pair [38] to find the horizon, using a RANSAC-based

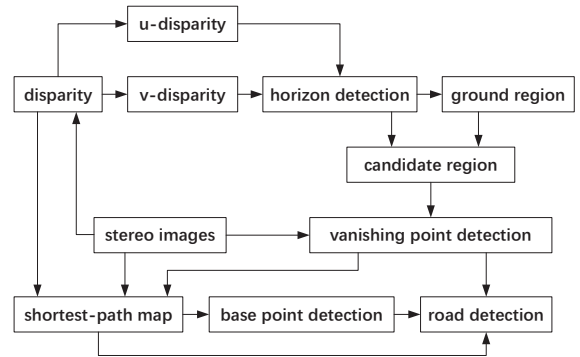


Fig. 1. The flowchart of the proposed stereo-vision based vanishing point detection for road detection framework.

variant of Labayrade et al. [16] (see [17], [18] for related work). Then the vanishing point position of road can be estimated by a voting strategy [22], [23], [24], where the vanishing point candidate space is reduced to a very small region that corresponds to an area near the horizon. Due to the robustness in horizon detection, the stereo-based vanishing point estimation is also more accurate compared to the vanishing point estimation methods using a single camera.

Thereafter, the detected vanishing point is used as a guide to find road region. Specifically, we create a weighted graph where the set of nodes correspond to all pixels, and the weight between any two nodes is calculated based on gradient information that is extracted from both the gray-scale image and disparity map. Then the vanishing point position is set as the source node of the graph, and we calculate a single-source minimum cost map based on the Dijkstra algorithm [25], where we can calculate the shortest path between the source node and the other pixels, respectively. To find the road region on which the vehicle is driving, two base (terminal) points on the image bottom row are automatically located based on the shortest-path map. The two terminal points mostly correspond to the locations where the two lane markers meet the bottom of image, or the locations where the road borders join the image bottom when there are no painted markers. Finally, the two road boundaries can be detected by tracing back from the two terminal points to the vanishing point in the shortest-path map, respectively.

The contributions of this paper are as follows: (1) To detect horizon, we propose a weight-sampling RANSAC-like method to detect the road profile line from the improved v-disparity map. (2) We propose a stereo-vision based road vanishing point detection method. (3) We propose a vanishing-point constrained Dijkstra model for road segmentation. The rest

<sup>†</sup> Yigong Zhang, Yingna Su, Jian Yang and Hui Kong are with the Department of Computer Science, Nanjing University of Science and Technology, Nanjing, China. E-mail: {ygzhang, ynsu, jianyang, konghui}@njust.edu.cn

<sup>‡</sup> Jean Ponce is with the Department of Informatique, Ecole Normale Supérieure, Paris, France. Email: jeanponce@ens.fr

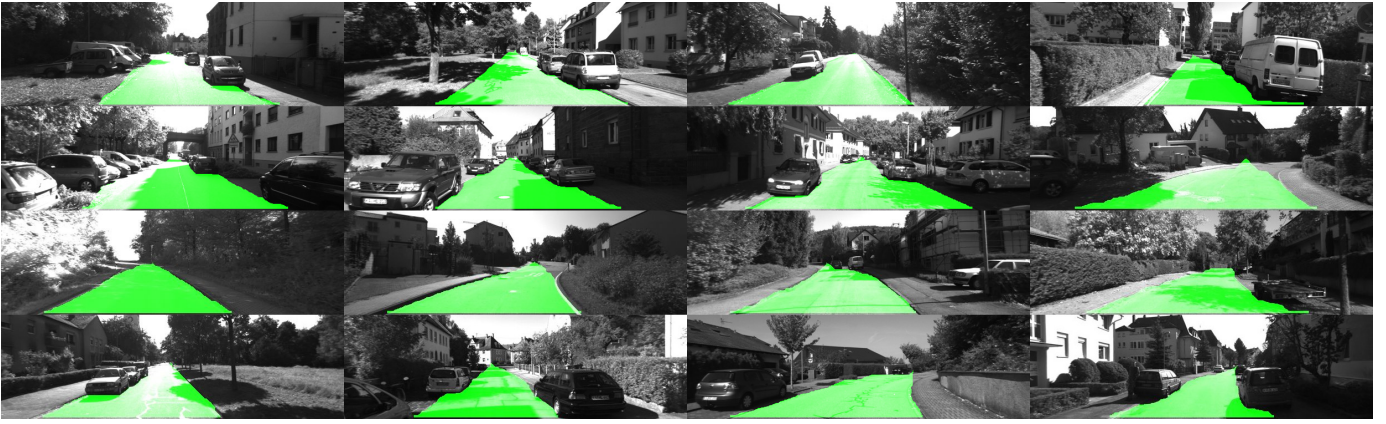


Fig. 2. Examples of road detection results by our stereo-based vanishing point constrained Dijkstra road model.

of this paper is organized as follows: we review some related works in Section II, and give the details of our method in Section III. The experimental results and comparison are given in Section IV. The conclusions are drawn in the last section. The supplemental results are uploaded to the Github site <https://github.com/kiki0378/Dijkstra-based-Road-Detection>.

## II. RELATED WORKS

In the literature, most road/lane detection methods rely on monocular camera. Among these methods, some are based on the intensity information (contrast between pixels) and color features of the road area [26], [27], [28], [29], [30]. Some methods fit a parametric 2D curve model on the geometric structures of the road, such as lane markings. Lee et al. propose a lane-departure identification system [10]. The lane boundary is extracted by their lane boundary pixel extractor and is fitted with line segment by the Hough transform. Khosla uses two continuous clothoid segments to model road edge [11]. The parameters of clothoids are estimated by Kalman Filter with a new dynamics model. Jung and Kelber apply Hough transform to fit the detected land boundary in the first frame and then follow the road boundary in the subsequent frames based on two different models: a linear model used in the near field and a quadratic model used in the far field [6]. Wang et al represent two road boundaries as a pair of hyperbolas [12]. The parameters of hyperbolas are calculated by an M-estimator which fits a line to the road vanishing point. Wang et al propose a Catmull-Rom spline model to describe the perspective effect of parallel lane boundaries [13]. The control points of spline are estimated by a maximum likelihood method. Yagi et al describe the road boundary as an active contour model based on a parallelism and width-constant constraint [15]. They also generate the 3D road model depending on the active contour model and the known yaw and pitch angles.

Except for 2D geometry models in the image space, Some researchers generate the road model in surrogate 3D image (a top view of the scene) by the inverse perspective mapping [31], [32], [33] according to camera calibration parameters or homography transformation. In Aly's work [31], all the lane boundaries are detected by Gaussian filtering and fitted by

a Bezier Spline model in the top view of the road, which runs in real-time. In [32], Seo and Rajkumar also detect lane-marking in the top view and remove false detections by using vanishing point in the original image space. Kang et al use cross ratio to estimate adjacent lane based on the parallelism constraint in the bird's eye view and verify estimations by their proposed maximum gradient square sum method with the help of vanishing point.

Besides, stereo cameras have also been utilized in road/lane detection. Petrovai et al. propose a stereo-based framework for lane detection/tracking and frontal obstacle detection in mobile devices equipped with dual cameras [34]. The lane detection and tracking is performed using single camera while the obstacle detection is carried out based on stereo vision. In [35], a 3D lane detection method is given based on stereo vision, where the parameters of the 3D lane, such as width, horizontal and vertical curvature, roll, pitch, and yaw angles, are estimated. Similarly, Nedevschi et al. propose a stereo vision based lane detection and tracking scheme, where the lane is modeled as a 3D surface, defined by the vertical and horizontal clothoid curves, the lane width and the roll angle. The lane detection is integrated into a Kalman-filter tracking process. Danescu et al. propose a lane estimation method based on the particle-filtering framework [36]. The solution employs a polar-histogram based technique for pitch detection, a novel method for particle measurement and weighing using multiple lane-delimiting cues extracted by grayscale and stereo data processing, and a quality factor for deciding upon the validity of the lane-estimation results.

The most related works to ours are the road/lane detection methods based on vanishing point detection [24], [22], [19], [21], where the road vanishing point can provide a global constraint on (nearly) straight road region. The Orientation Consistency Ratio (OCR) feature [23], [24] has been extracted based on the detected vanishing point to locate most likely road borders. In general, these methods cannot work well when the road region is not perfectly straight and when there are vehicles parking along road boundaries. In addition, these vanishing point constrained road detection approaches cannot precisely locate road boundaries. In contrast, our approach can deal with these problems.

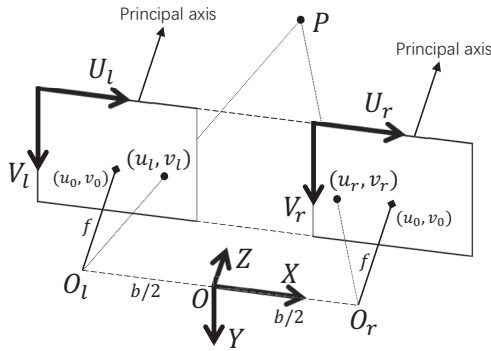


Fig. 3. The calibrated stereo rig and the coordinate system used.

### III. VANISHING POINT DETECTION BY STEREO CAMERA

The most widely used method for detecting road vanishing point is probably based on the intersection of straight lines which are detected by Hough transform [37]. However, this type of methods are not suitable for the vanishing point detection task when there are no painted road markers or when road borders are not straight enough. For such road scenes, straight lines cannot be robustly detected.

The second type of methods that are often used for road vanishing point detection is the texture-based voting framework [19], [20], [21], [22], [23], [24], where each pixel can be a voter for the vanishing point location, and can be a vanishing point candidate as well. These methods are robust for both structured and non-structured road types. However, there are two limitations in the existing texture-based methods. First, they are computationally complex in detecting vanishing point. Second, these methods are prone to fail when shadows are present on road regions.

In this section, we propose an efficient texture-based voting approach for road vanishing point detection from a stereo pair. First, we propose a robust RANSAC-like line fitting strategy to detect the horizon accurately. Then we can obtain the approximate ground region. With the horizon and ground region constraints, the vanishing point candidate region is determined and the vanishing point is detected as the pixel which receives the highest votes in the candidate region.

#### A. Detection of Horizon by the Improved $v$ -disparity Map

A calibrated stereo rig coordinate system is showed in Fig. 3. Two image planes of the stereo rig belong to the same plane.  $(U_l, V_l)$  and  $(U_r, V_r)$  are the coordinate axes in the left and the right image, respectively. The  $(u_0, v_0)$  is the image coordinate of the projection of the optical center of the camera. The  $f$  is the focal length, and  $b$  is the stereo baseline (the length between the left optical center  $O_l$  and the right one  $O_r$ ).  $(X, Y, Z)$  represent three axes of the world coordinate system. To simplify the representations, we assume that  $X$  is parallel to  $U_l$  and  $U_r$ ,  $Y$  parallel to  $V_l$  and  $V_r$ , and  $Z$  parallel to the principal axis. The  $O$  represents the origin of the world coordinate system and is located at the midpoint of two optical centers.

In this system, a 3D world point,  $P(x, y, z)$ , is projected onto the left and right rectified images. The corresponding

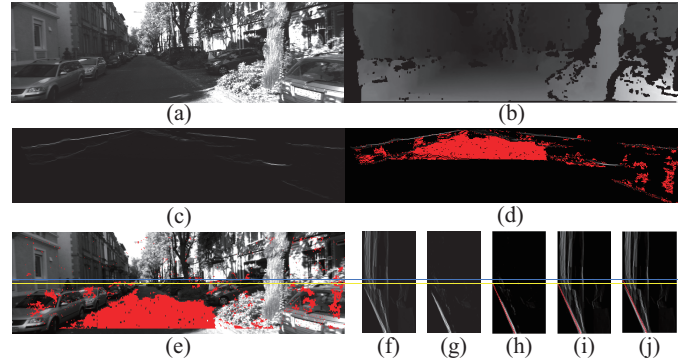


Fig. 4. Detect the straight line in the improved  $v$ -disparity map that corresponds to the planar ground region based on a RANSAC-like scheme. (a) the input image; (b) the corresponding disparity map  $\mathbf{D}$ ; (c) the  $u$ -disparity map  $\mathbf{U}$  of the input image; (d) the flatness region  $\mathcal{R}_{flatness}$  (red region) whose pixel values are smaller than 2 in  $\mathbf{U}$ ; (e) the sampling space  $\mathcal{R}_{cr}$  (red region) which corresponds to  $\mathcal{R}_{flatness}$  in the  $u$ -disparity map. Apparently, the pixels corresponding to  $\mathcal{R}_{flatness}$  are mostly in flat area; (f) the original  $v$ -disparity map of the input image; (g) the improved  $v$ -disparity map which is obtained only based on the disparity values in the region of sampling space in (e); (h) the fitting result (red line) based on (g); (i) the fitting results (red line) plotted in (f); (j) the fitting result (red line) based on (f) only. The yellow line represents the detected horizon in (i). The cyan line represents the detected horizon in (j). Apparently, the usage of pixels with low  $u$ -disparity value improves the accuracy of horizon detection.

coordinates in the left and the right image coordinate system are given in (1).

$$\begin{cases} u_{(l,r)} = u_0 + \frac{f}{z}(x \pm \frac{b}{2}) \\ v = v_l = v_r = v_0 + \frac{f}{z}y \end{cases} \quad (1)$$

Thus the disparity value can be computed as

$$\Delta = u_l - u_r = \frac{fb}{z}. \quad (2)$$

In practice, the disparity value for each pixel in the image can be obtained by a stereo matching method [16] (Fig. 4(b)).

Assuming that the road lies in a plane  $\pi_1$  parallel to the base line and with a little pitch angle,  $\pi_1$  can be formulated as  $y = a_1 z + d_1$  ( $a_1$  is small). Thus it can also be formulated as

$$v = v_0 + f a_1 + \frac{d_1}{b} \Delta. \quad (3)$$

Hence, the points lying in the road plane  $\pi_1$  form a line in the  $(\Delta, v)$  plane. In reality, the projection in the  $(\Delta, v)$  plane is achieved by the  $v$ -disparity map [16], where each row is a histogram of the disparity value  $\Delta$  obtained in the corresponding row of the disparity map. (Fig. 4(f)). Likewise, the road plane  $\pi_1$  is projected as a straight line in the  $v$ -disparity map. The horizon corresponds to the intersection of the straight line and the  $v$ -axis in the  $v$ -disparity map. Hence, by fitting a straight line to this linear structure, we can determine the road plane region and the horizon location in the image. Generally, the direct line fitting in the  $v$ -disparity map is adversely affected by the pixels in the upper part of the image that correspond to obstacles (see Fig. 4(f)).

To solve this problem, we propose an improved  $v$ -disparity map and a weighted-sampling RANSAC-like algorithm whose



Fig. 5. Detected horizons across different scenes based on v-disparity map.

detailed procedures are given in Algorithm 1. First, we compute the u-disparity map (Fig. 4(c)), where each column is a histogram of the disparity value  $\Delta$  obtained in the corresponding column of the disparity map. Assuming that an obstacle lies in an upright plane  $\pi_2$  parallel to the  $Y$ -coordinate,  $\pi_2$  can be formulated as  $z = a_2x + d_2$ . Then the relationship between  $\Delta$  and  $u_l$  for  $\pi_2$  is obtained as

$$u_l = u_0 + \frac{f}{a_2} + \left(\frac{b}{2} - \frac{d_2}{a_2}\right) \cdot \frac{\Delta}{b}. \quad (4)$$

Namely, the points lying on the obstacle plane  $\pi_2$  form a line in the  $(\Delta, u_l)$  plane, and the corresponding pixels in the original image are projected onto a line segment in the u-disparity map. Because all the pixels corresponding to the points on the obstacle plane  $\pi_2$  are projected to a line segment, the pixels on this line segment exhibit larger values in the u-disparity map. On the contrary, the pixels corresponding to the points on the road plane  $\pi_1$  are only projected onto a region in the u-disparity map, meaning that the pixels in this region have lower values in the u-disparity map. Therefore, the points on the road plane correspond to the pixels with small values in the u-disparity map. These pixels can be regarded as the flat region  $\mathcal{R}_{flatness}$ , which can be obtained approximately by a thresholding operation. After choosing the flat region, all pixels in  $\mathcal{R}_{flatness}$  in the u-disparity map are projected back to the original disparity map as the sampling space  $\mathcal{R}_{cr}$  (Fig. 4(e)). Consequently, we calculate the improved v-disparity map (Fig. 4(g)) by just taking into account the pixels in  $\mathcal{R}_{cr}$ .

Next we just process the improved v-disparity map and fit the straight line based on a weight-sampling RANSAC-like scheme. In the weighted-sampling strategy, two pixels are sampled from the improved v-disparity map in each iteration with a probability, where the probability of sampling a pixel is proportional to this pixel's intensity. Therefore, the higher a pixel's intensity, the more likely it could be sampled. Note that our proposed weighted-sampling method is different from PROSAC [54]. In PROSAC, the samples are not drawn from all data, but from a subset of the data with the highest quality. During the sampling procedure, the capacity of the sample set is growing. However, in our weighted sampling method, the samples are drawn from all data, and the capacity of the sample set is constant. In our method, the number of times a pixel appears in the sample set is datum-specific. For example, the intensity value of a pixel  $p$  in the v-disparity map is 5, then  $p$  will appear 5 times in the sample set. It is equivalent to sample a point with a probability. Fig. 5 shows more detected horizon results for different road scenes.

---

#### Algorithm 1 Fitting a straight line in v-disparity map

---

##### Require:

- 1:  $\mathbf{D}$ : the disparity map;
- 2:  $\epsilon$ : the inlier threshold for flatness region in u-disparity map
- 3:  $d_t$ : the inlier threshold of distance from a pixel to a line;
- 4:  $n_m$ : the number of maximum trials;

##### Ensure: $\mathcal{L}$ : a straight line;

- 5: Compute the u-disparity map  $\mathbf{U}$  from  $\mathbf{D}$ ;
  - 6: Choose the pixel  $\mathbf{U}_{ij} \in [1, \epsilon]$  as the flatness region  $\mathcal{R}_{flatness}$ ;
  - 7: Project the pixels in  $\mathcal{R}_{flatness}$  back to  $\mathbf{D}$  and obtain the sampling space  $\mathcal{R}_{cr}$ ;
  - 8: Calculate the v-disparity map  $\mathbf{V}$  using the pixels in  $\mathcal{R}_{cr}$ ;
  - 9: Set  $n=0$ ;
  - 10: Set the largest inlier value  $f_{max}=0$ ;
  - 11: **while** ( $n < n_m$ ) **do**;
  - 12:     Weighted-Sample two pixels,  $\mathbf{p}_1$  and  $\mathbf{p}_2$ , from  $\mathbf{V}$ ;
  - 13:     Obtain a straight line  $L$ , based on  $\mathbf{p}_1$  and  $\mathbf{p}_2$ ;
  - 14:     Find  $S_{inlier}$ , whose distance to  $L$  is less than  $d_t$ ;
  - 15:     Let  $\mathbf{V}_i$  be the value of  $i^{th}$  pixel of  $\mathbf{V}$ ;
  - 16:     Calculate the total inlier value,  $f = \sum_{i \in S_{inlier}} \mathbf{V}_i$ ;
  - 17:     **if**  $f > f_{max}$  **then**
  - 18:          $f_{max} = f$ ;
  - 19:          $\mathcal{L} = L$ ;
  - 20:     **end if**
  - 21:      $n=n+1$ ;
  - 22: **end while**
- 

#### B. Detection of Approximate Road and Obstacle Regions

Given the detected straight line in the v-disparity map (by mapping the detected line from the improved v-disparity map to the v-disparity map), we can approximately find the approximate ground and obstacle regions in the image. The approximate ground region mostly corresponds to road area, and the obstacle regions include buildings, vehicles, trees or bushes etc. Specifically, given the element located at the  $i^{th}$  row and  $j^{th}$  column of the disparity map, its disparity value is represented by  $d_{ij}$ . The fitted straight line  $\mathcal{L}$  in the v-disparity map is given by

$$x_i = \frac{1}{k} \times (y_i - b), i \in [1, h] \quad (5)$$

where  $k$  and  $b$  are the line parameters. The  $L_i = (x_i, y_i)$  corresponds to the horizontal and vertical coordinates of the  $i^{th}$  point on the straight line  $\mathcal{L}$ . The  $h$  is the height of the v-disparity map. The pixel at the  $i^{th}$  row and  $j^{th}$

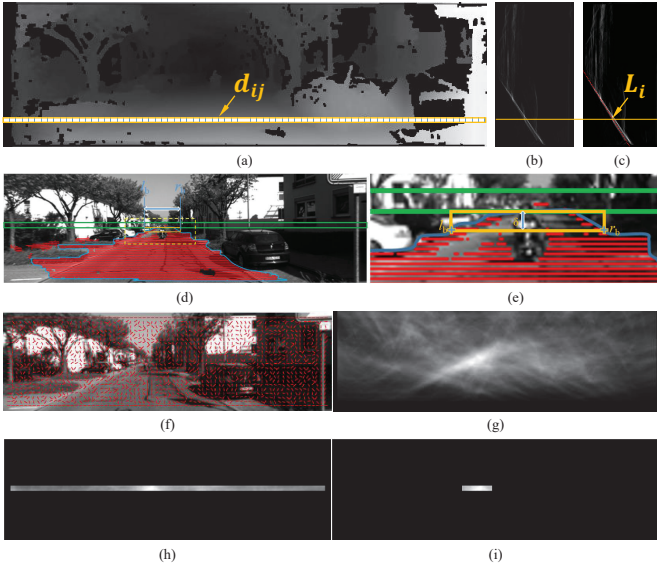


Fig. 6. (a) the disparity map of the input image, where  $d_{ij}$  is the disparity value at the  $i^{th}$  row and  $j^{th}$  column; (b) the v-disparity map of the input image; (c) the detected road-profile line in the v-disparity image, where  $L_i$  is the horizontal coordinates of the  $i^{th}$  point whose vertical coordinate is  $i$ ; (d) the approximately detected ground region (red) based on Eq.6, where the area between two green lines is set to the region for vanishing point candidates due to its closeness to horizon. In reality, we only vote for a part of the candidate region that are delimited by  $l_b$  and  $r_b$ ; (e) the enlarged version of the dotted yellow rectangle in (d). The height of the orange rectangle is  $\delta$  (8 pixels in our paper), and the top side is coincident with the top of the encompassing boundary of the approximate ground region. The  $l_b$  and  $r_b$  are obtained as the joints of the bottom side of the orange rectangle with the encompassing boundary of the approximate ground region; (f) the orientation bars overlaid on image. (g) the voting map for vanishing point based on [23]; (h) the reduction of vanishing point candidate region to a narrow strip close to the horizon; (i) a further reduction of vanishing point candidate region.

column,  $p_{ij}$ , can be approximately classified into approximate ground/obstacle region if the corresponding disparity value  $d_{ij}$  satisfies

$$\begin{cases} p_{ij} \in \text{approximate ground region, if } ||d_{ij} - x_i|| \leq \alpha * x_i \\ p_{ij} \in \text{obstacle region, otherwise} \end{cases} \quad (6)$$

where  $x_i$  is the horizontal coordinate of the  $i^{th}$  point  $L_i$  of the fitted straight line  $\mathcal{L}$ . Fig. 6 shows an example of approximate ground and obstacle region detection based on Eq.6.

### C. Vanishing Point Voting With Horizon and Road Region Constraints

Once we have estimated the location of horizon and the approximate ground region, we adopt the texture-based voting methods [22], [23], [24] for vanishing point detection. The procedure of voting-based vanishing point detection consists of selection of vanishing point candidates and voters (pixels used to vote for a vanishing point). As illustrated in the second image of Fig. 6(g), there are too many possibilities for the locations of the vanishing point candidates if there is no constraint on it, and this will result in a very time-consuming voting procedure in these voting based methods. With the constraint that vanishing point of road is coincident with the estimated horizon, the horizontal coordinate of the true vanishing point can be decided by the several top rows



Fig. 7. Example of a road image with two marked voters  $p_1$  and  $p_2$  with estimated orientation of  $o_1$  and  $o_2$ , respectively.

of ground region in the image space, and the vanishing point candidate space can be decreased significantly. Fig. 6 illustrates the voting-based vanishing point procedure with the two constraints on selecting vanishing point candidate region. In Fig. 6(d), the vanishing point candidates are chosen as the pixels within a strip of image region around the detected horizon, with a height of eight pixels, shown as the region between the two green lines in Fig. 6(d). We choose a strip of image region of pixels instead of only the pixels on the horizon in case that the horizon might not be accurately detected. To further reduce complexity, we reduce the vanishing point candidate region from (d) to get (e) based on the ground region constraint. As shown in Fig. 6(e), the top  $\delta$  rows of ground region (in the orange rectangle) are chosen as the end of the road, with the left and right bounds being  $l_b$  and  $r_b$ , respectively. Because vanishing point should exist at the end of road in theory, the horizontal coordinate of the vanishing point should be between  $l_b$  and  $r_b$ .

Next, we need to set the pixels (voters) which can be used to vote for the vanishing point. In this paper, the pixels that are under the estimated horizon in the image are selected as voters. We estimate texture orientation for each pixel based on the generalized Laplacian of Gaussian (gLoG) filter [22]. Fig. 6 (f) shows the estimated orientation bars overlaid on the left image of the stereo camera. We adopt the voting method given in [22] to obtain the road vanishing point (Fig. 7). That is, a pixel  $\mathbf{p}$ , whose texture orientation is  $\mathbf{o}_p$ , can vote for any pixel  $\mathbf{v}$  in the vanishing point candidate region as long as the angle (in degrees) between the direction  $\overline{\mathbf{p}\mathbf{v}}$  and the vector  $\mathbf{o}_p$ ,  $\gamma = \angle(\overline{\mathbf{p}\mathbf{v}}, \mathbf{o}_p)$ , is below a fixed threshold  $\rho$ . Note that we use every pixels in the voter region to vote for each vanishing point candidate.

$$\text{Vote}(\mathbf{p}, \mathbf{v}) = \begin{cases} \exp(-d(\overline{\mathbf{p}\mathbf{v}}) * |\gamma|/\tau) & \text{if } \gamma \leq \rho \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

where  $d(\overline{\mathbf{p}\mathbf{v}})$  means the distance between  $\mathbf{p}$  and  $\mathbf{v}$ . The  $\tau$  is a normalization factor, set as the diagonal length of image.

## IV. A DIJKSTRA-MODEL FOR ROAD DETECTION

Vanishing point provides a strong cue for the extension of road region into distant area. It has been utilized to find a rough road region based on an Orientation Consistency Ratio (OCR) feature [23], [24]. However, the utilization of vanishing point plus OCR feature is not able to accurately locate road borders when road is not strictly straight. Although the replacement of the straight-line road border model with a nonlinear (such as

a quadratic contour model) curve could improve localization accuracy, it is a very challenging task when the road region is affected by shadows or occlusions, which could severely corrupt the fitting accuracy of the parametric models.

To deal with the difficulty in fitting a parametric model to corrupted road image, we propose a non-parametric road border detection/segmentation method based on a vanishing point constrained Dijkstra road model. The Dijkstra algorithm [39] can compute the shortest distance from a given source node to any other nodes based on the weight (cost) between nodes. In the image analysis community, especially medical image analysis, the Dijkstra algorithm has been applied to object boundary segmentation with success [40], [41], [42], [43]. In general, these methods are interactive, requiring users to specify a seed pixel (as the source node) to the Dijkstra algorithm and some control points (markers) as a guide to the segmentation algorithm. Therefore, it is not appropriate to apply these approaches directly to road border segmentation. In addition, these conventional Dijkstra-based approaches calculate cost between nodes based on gradient and edge orientation in the gray-scale image. It turns out that the calculated cost is very sensitive to shadows, which are very common in road images, and can result in wrong road boundary segmentations.

To overcome these problems, we propose a vanishing point constrained Dijkstra approach to locate road borders. All pixels of the given image are used as nodes and edges are linked between each pixel and its eight neighbors. Because vanishing point provides a strong cue for the extension of road region into distant area, it is chosen as the source node. Then the Dijkstra algorithm is applied to calculate the shortest path from the vanishing point to all other pixels to generate the shortest-path map based on the defined edge weights. At last, two base points on the image bottom row are determined from the shortest-path map. The two road boundaries can be detected by tracing back from the two base points to the vanishing point in the shortest-path map, respectively.

#### A. Definition of Various Types of Costs

Since the road boundary detection problem is formulated as a shortest-path problem, the edge weights used in Dijkstra-based approach are of the primary importance. The edge weight is a weighted sum of each component cost function, which ensures that the obtained shortest path between two road boundary nodes is expected to be coincident with the road boundary. Let  $c(\mathbf{p}_i, \mathbf{p}_j)$  represent the edge weight for the edge from a pixel  $\mathbf{p}_i$  to its neighbor pixel  $\mathbf{p}_j$ , and the function is

$$c(\mathbf{p}_i, \mathbf{p}_j) = w_g \cdot f_g(\mathbf{p}_j) + w_f \cdot f_f(\mathbf{p}_j) + w_{df} \cdot f_{df}(\mathbf{p}_j) + w_{gd} \cdot f_{gd}(\mathbf{p}_j) + w_o \cdot f_o(\mathbf{p}_i, \mathbf{p}_j) \quad (8)$$

where  $f_g$  is the gradient cost,  $f_o$  being the link (orientation consistency) cost,  $f_f$  being the flatness cost,  $f_{df}$  being the disparity feature cost and  $f_{gd}$  being the texture orientation cost. The notations of these costs are listed in Table I.

TABLE I  
THE VARIOUS KINDS OF COSTS BETWEEN TWO NEIGHBORING PIXELS

cost function	notation
gradient cost	$f_g$
orientation consistency (link) cost	$f_o$
flatness cost	$f_f$
disparity feature cost	$f_{df}$
gradient direction cost	$f_{gd}$

**Gradient cost:** The cost of  $f_g(\mathbf{p}_v)$  at pixel  $\mathbf{p}_v$  is calculated based on the gradient magnitude of  $\mathbf{p}_v$  in the gray-scale image,

$$f_g(\mathbf{p}_v) = \frac{G_{\max} - G(\mathbf{p}_v)}{G_{\max}} = 1 - \frac{G(\mathbf{p}_v)}{G_{\max}} \quad (9)$$

where  $G(\mathbf{p}_v)$  is the gradient magnitude at pixel  $\mathbf{p}_v$ , and  $G_{\max}$  is the maximum gradient magnitude of all pixels.

**Link cost:** The cost of  $f_o(\mathbf{p}_u, \mathbf{p}_v)$  is a constraint to the boundary by associating a high cost for sharp changes in boundary direction, given by Eq.10,

$$f_o(\mathbf{p}_u, \mathbf{p}_v) = \frac{2}{3\pi} \{ \arccos[O_{\mathbf{p}_u}(\mathbf{p}_u, \mathbf{p}_v)] + \arccos[O_{\mathbf{p}_v}(\mathbf{p}_u, \mathbf{p}_v)] \} \quad (10)$$

where

$$O_{\mathbf{p}_u}(\mathbf{p}_u, \mathbf{p}_v) = O_{\mathbf{p}_u} \cdot L(\mathbf{p}_u, \mathbf{p}_v) \quad (11)$$

$$O_{\mathbf{p}_v}(\mathbf{p}_u, \mathbf{p}_v) = O_{\mathbf{p}_v} \cdot L(\mathbf{p}_u, \mathbf{p}_v) \quad (12)$$

measure the orientation consistencies between the vector  $O_{\mathbf{p}_u}$  and the vector  $L(\mathbf{p}_u, \mathbf{p}_v)$ , and between the vector  $O_{\mathbf{p}_v}$  and the vector  $L(\mathbf{p}_u, \mathbf{p}_v)$ , respectively. Note that the vector  $O_{\mathbf{p}_u}$  is the texture orientation at  $\mathbf{p}_u$  which can be calculated as  $O_{\mathbf{p}_u} = [I_y(\mathbf{p}_u), -I_x(\mathbf{p}_u)]$ . with  $G_{\mathbf{p}_u} = [I_x(\mathbf{p}_u), I_y(\mathbf{p}_u)]$  being the gradient direction at  $\mathbf{p}_u$ . Likewise, the vector  $O_{\mathbf{p}_v}$  is the texture orientation at  $\mathbf{p}_v$  which can be calculated as  $O_{\mathbf{p}_v} = [I_y(\mathbf{p}_v), -I_x(\mathbf{p}_v)]$ .

The  $L(\mathbf{p}_u, \mathbf{p}_v)$  in Eq.11 and Eq.12 is a vector linking pixels  $\mathbf{p}_u$  and  $\mathbf{p}_v$ , defined in Eq.13. Because  $\mathbf{p}_u$  and  $\mathbf{p}_v$  are neighboring pixels, the  $L(\mathbf{p}_u, \mathbf{p}_v)$  can be horizontal, vertical, or diagonal. Obviously, The  $f_o(\mathbf{p}_u, \mathbf{p}_v)$  associates a high cost to a link between two pixels that have similar gradient directions but are perpendicular, or near perpendicular, to the link direction between them, and a low cost when the gradient directions of the two pixels are both similar to each other and to the link direction between them.

$$L(\mathbf{p}_u, \mathbf{p}_v) = \begin{cases} \mathbf{p}_v - \mathbf{p}_u & \text{if } O_{\mathbf{p}_u} \cdot (\mathbf{p}_v - \mathbf{p}_u) \geq 0 \\ \mathbf{p}_u - \mathbf{p}_v & \text{if } O_{\mathbf{p}_u} \cdot (\mathbf{p}_v - \mathbf{p}_u) < 0 \end{cases} \quad (13)$$

Because  $\arccos[O_{\mathbf{p}_u}(\mathbf{p}_u, \mathbf{p}_v)]$  equals the angle between the vector  $\mathbf{p}_u$  and the vector  $L(\mathbf{p}_u, \mathbf{p}_v)$ ,  $\arccos[O_{\mathbf{p}_v}(\mathbf{p}_u, \mathbf{p}_v)]$  is equal to the angle between the vector  $\mathbf{p}_v$  and the vector  $L(\mathbf{p}_u, \mathbf{p}_v)$ , the values of  $\arccos[O_{\mathbf{p}_u}(\mathbf{p}_u, \mathbf{p}_v)]$  and  $\arccos[O_{\mathbf{p}_v}(\mathbf{p}_u, \mathbf{p}_v)]$  are positive and no bigger than  $\pi$ . In addition, the definition of  $L(\mathbf{p}_u, \mathbf{p}_v)$  (defined in Eq.13) constrains the angle between  $\mathbf{p}_u$  and  $L(\mathbf{p}_u, \mathbf{p}_v)$  to be positive and no bigger than  $\pi/2$ . Therefore, the sum of  $\arccos[O_{\mathbf{p}_u}(\mathbf{p}_u, \mathbf{p}_v)]$  and  $\arccos[O_{\mathbf{p}_v}(\mathbf{p}_u, \mathbf{p}_v)]$  is no bigger than  $3\pi/2$ . Therefore, the normalization factor is set to  $2/(3\pi)$ .



Fig. 8. Example of the gray-scale images (left) and the corresponding disparity feature cost map (right). The disparity feature cost in the curb region is obviously lower than that of its surrounding road region.

**Flatness cost:** To calculate the flatness cost at pixel  $\mathbf{p}_v = [\mathbf{p}_v^x, \mathbf{p}_v^y]$ , where  $\mathbf{p}_v^x$  and  $\mathbf{p}_v^y$  represents the horizontal and vertical coordinates of  $\mathbf{p}_v$ , we first locate the point ( $\mathcal{L}_{\mathbf{p}_v^y}$ ) on the fitted straight line in the  $v$ -disparity map, the horizontal coordinate of the point  $\mathcal{L}_{\mathbf{p}_v^y}$  is calculated as follows,

$$\chi = \frac{1}{k} \times (\mathbf{p}_v^y - b) \quad (14)$$

where  $k$  and  $b$  are the parameters of the fitted line in the  $v$ -disparity map (as described in III-B). Note that  $\chi$  is also the “dominant” disparity value on the ( $\mathbf{p}_v^y$ )-th row of the disparity map, or the disparity value of the pixels in the flat road region. Then the flatness cost at pixel  $\mathbf{p}_v = [\mathbf{p}_v^x, \mathbf{p}_v^y]$  is calculated by

$$f_f(\mathbf{p}_v) = \begin{cases} 1 & \text{if } \|d_{\mathbf{p}_v} - \chi\| \leq \alpha * \chi \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

where  $d_{\mathbf{p}_v}$  is the disparity value at  $\mathbf{p}_v$ .

**Disparity feature cost:** The disparity feature  $f_{df}$  is computed for each pixel from the disparity map based on a descriptor, which describes the relationship between each pixel and its eight neighbor pixels. We use it to emphasize curb regions because the disparity feature cost value in the curb region is smaller than that of the other road region, as shown in Fig. 8. Consequently, we can design a cost to determine the curb region as the road boundary through the use of this feature.

Next, we introduce how to generate the disparity feature cost function. In theory, when the base line of the stereo camera is parallel to the road surface, the pixels of road region have the same disparity value in the same row, and the farther the road region pixels, the smaller the disparity values. The disparity values of pixels on upright obstacles remain the same. The descriptor at each pixel can be extracted within a  $3 \times 3$  block to describe the relationship between the central pixel and its eight neighbor pixels. Let  $d_{i,j}$  denote the disparity value of pixel  $\mathbf{p}_{i,j}$ . For convenience, the disparity values of central pixel and its eight neighbor pixels are represented as follows:  $b_0 = d_{i-1,j-1}$ ,  $b_1 = d_{i-1,j}$ ,  $b_2 = d_{i-1,j+1}$ ,  $b_3 = d_{i,j-1}$ ,  $b_4 = d_{i,j}$ ,  $b_5 = d_{i,j+1}$ ,  $b_6 = d_{i+1,j-1}$ ,  $b_7 = d_{i+1,j}$  and  $b_8 = d_{i+1,j+1}$ . The descriptor at  $\mathbf{p}_{i,j}$ ,  $F(\mathbf{p}_{i,j})$ , can be calculated as:

$$F(\mathbf{p}_{i,j}) = \sum_{k=0}^7 c_k 2^k \quad (16)$$

where  $c_k$  ( $k = 0, 1, \dots, 7$ ) is defined as follows:

$$c_0 = \begin{cases} 1 & \text{if } b_0 + b_1 + b_2 < b_3 + b_4 + b_5 \\ 0 & \text{if } b_0 + b_1 + b_2 \geq b_3 + b_4 + b_5 \end{cases},$$

$$c_1 = \begin{cases} 1 & \text{if } b_3 + b_4 + b_5 < b_6 + b_7 + b_8 \\ 0 & \text{if } b_3 + b_4 + b_5 \geq b_6 + b_7 + b_8 \end{cases},$$

$$c_2 = \begin{cases} 1 & \text{if } b_1 < b_4 \\ 0 & \text{if } b_1 \geq b_4 \end{cases}, \quad c_3 = \begin{cases} 1 & \text{if } b_4 < b_7 \\ 0 & \text{if } b_4 \geq b_7 \end{cases},$$

$$c_4 = \begin{cases} 1 & \text{if } b_0 < b_4 \\ 0 & \text{if } b_0 \geq b_4 \end{cases}, \quad c_5 = \begin{cases} 1 & \text{if } b_2 < b_4 \\ 0 & \text{if } b_2 \geq b_4 \end{cases},$$

$$c_6 = \begin{cases} 1 & \text{if } b_4 < b_6 \\ 0 & \text{if } b_4 \geq b_6 \end{cases}, \quad c_7 = \begin{cases} 1 & \text{if } b_4 < b_8 \\ 0 & \text{if } b_4 \geq b_8 \end{cases}.$$

At last, the disparity feature cost at pixel  $\mathbf{p}_{i,j}$  is formulated as Eq.17:

$$f_{df}(\mathbf{p}_{i,j}) = \frac{F(\mathbf{p}_{i,j})}{F_{\max}} \quad (17)$$

where  $F(\mathbf{p}_{i,j})$  is the calculated descriptor value at pixel  $\mathbf{p}_{i,j}$ , and  $F_{\max}$  is the maximum descriptor value of all pixels.

**Gradient direction cost:** The gradient direction cost  $f_{gd}(\mathbf{p}_v)$  at pixel  $\mathbf{p}_v$  is calculated based on the angle between the gradient direction  $Q_{\mathbf{p}_v}$  at  $\mathbf{p}_v$  and the direction  $\overline{\mathbf{v}\mathbf{p}_v}$ ,  $\gamma_{\mathbf{p}_v} = \angle(\overline{\mathbf{v}\mathbf{p}_v}, Q_{\mathbf{p}_v})$ :

$$f_{gd}(\mathbf{p}_v) = \begin{cases} 1 & \text{if } \gamma_{\mathbf{p}_v} \leq (1 - \frac{d(\overline{\mathbf{v}\mathbf{p}_v})}{\max(d(\overline{\mathbf{v}\mathbf{p}_{lb}}), d(\overline{\mathbf{v}\mathbf{p}_{rb}}))}) * \beta \\ 0, & \text{otherwise} \end{cases} \quad (18)$$

where  $Q_{\mathbf{p}_v}$  is the gradient direction at pixel  $\mathbf{p}_v$ ,  $\overline{\mathbf{v}\mathbf{p}_v}$  being the vector linking the vanishing point  $\mathbf{v}$  and  $\mathbf{p}_v$ ,  $d(\overline{\mathbf{v}\mathbf{p}_v})$  being the distance between  $\mathbf{v}$  and  $\mathbf{p}_v$ ,  $d(\overline{\mathbf{v}\mathbf{p}_{lb}})$  being the distance between  $\mathbf{v}$  and the left-bottom corner of image  $\mathbf{p}_{lb}$ ,  $d(\overline{\mathbf{v}\mathbf{p}_{rb}})$  being the distance between  $\mathbf{v}$  and the right-bottom corner of image  $\mathbf{p}_{rb}$ .

### B. The Vanishing Point Constrained Dijkstra Model

We apply the Dijkstra algorithm with the vanishing point being the source node to compute the shortest paths to the other pixels. The cost between two nodes  $\mathbf{p}_i$  and  $\mathbf{p}_j$ ,  $c(\mathbf{p}_i, \mathbf{p}_j)$ , is calculated based on Eq.8. Technically, all pixels of the given image are used as nodes to construct a weighted graph, where each node is connected to its neighbors. Because the vanishing point is generally above road area in images, although each node is connected to its eight neighbors, we do not calculate the cost between each node and its up-left, top and up-right neighbors, respectively, in the Dijkstra graph search procedure. Consequently, we obtain the shortest-path map which computes the least cost from vanishing point to any other pixels in the image (Fig. 9(g)).

### C. Finding Two Base Points in the Shortest-path Map

With the obtained shortest-path map, we still need two terminal (control) points if we want to utilize the shortest-path map to find road borders. To find the two terminal points  $t_1$  and  $t_2$ , we search the last row of the shortest-path map. Let  $S_{i,j}$  be the value at the  $i^{th}$  column and  $j^{th}$  row of the shortest-path map, and  $\ell(rb_i)$  is the length of the  $i^{th}$  path  $rb_i$ . Note that the  $\ell(rb_i)$  is the sum of the distances between any two neighbor pixels on  $rb_i$ . For example, the distance



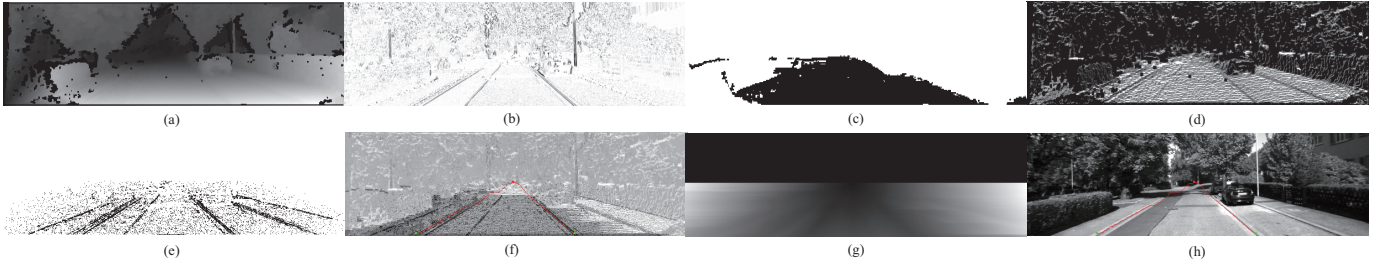


Fig. 9. Example of the Dijkstra-model for road border segmentation. (a): the disparity map. (b): the gradient cost  $f_g$ . (c): the flatness cost  $f_f$ . (d): the disparity feature cost,  $f_{df}$ , extracted from (a). (e): the gradient direction cost  $f_{gd}$ . (f): the weighted sum of (b), (c), (d) and (e). (g): the shortest-path map based on  $f_g$ ,  $f_f$ ,  $f_{df}$ ,  $f_{gd}$  and  $f_o$ . (h) the segmented road border based on the local cost given in Eq.8

between a pixel and its upper neighbor is 1, and the distance between a pixel and its upper-left neighbor is  $\sqrt{2}$ . The two terminal points are to the left and right of the middle point of the last row of the shortest-path map, respectively. The search criterion is based on Eq.19 and Eq.20, where  $\ell(rb_i)$  acts as a normalization parameter.

$$t_1 = \arg \min_{i=1}^{W/2} (\mathcal{S}_{i,H} / \ell(rb_i)), \quad (19)$$

$$t_2 = \arg \min_{i=W/2+1}^W (\mathcal{S}_{i,H} / \ell(rb_i)), \quad (20)$$

where  $H$  and  $W$  are the image height and width. Once we find the two terminal points  $t_1$  and  $t_2$ , we can find the two road borders from the vanishing point to  $t_1$  and  $t_2$ , respectively.

## V. EXPERIMENTS

### A. Dataset

To evaluate the performance of our proposed approach, we randomly select 2621 gray-scale stereo image pairs from the KITTI odometry dataset [50]. All of these images have noticeable road vanishing points and road regions. The selected images contain several variant scenarios including straight and curve roads, highways, residential and village roads, roads with shadows and vehicles, etc. To facilitate the evaluation and comparison, all images are normalized to the same size with the height of 188 and the width of 620.

To evaluate the accuracy of vanishing point detection of the proposed method and other compared methods, we compare them with the ground-truth vanishing point locations, which are labeled manually in each left gray-scale image. To guarantee an accurate labeling, we request five research students to record the locations of road vanishing point for the whole dataset after they are trained to understand the road vanishing point concept. To remove the influence of subjectivity, the median results are regarded as the true locations. In addition, we regard the vertical coordinate of the ground-truth vanishing point as the ground-truth position of horizon. Examples of the labeled vanishing points are shown in Fig. 10. We also label the ground-truth road regions in every gray-scale left image manually. Note that the labeled road regions only contain the driving zone in front of car and the other regions at the joint of two crossed roads are ignored. Examples of the ground-truth road regions are shown in Fig. 11.

We also test our road detection method on the Oxford Robotcar Dataset [53] to show the generalization ability. This dataset provides road scenarios under various illumination



Fig. 10. Examples of the ground-truth vanishing point (green cross).



Fig. 11. Examples of the ground-truth road region (red).

conditions. We randomly select 1643 color frames to evaluate the performance. The region of the vehicles bonnet in the bottom of image is cut out and then the image is normalized to the same size with the height of 221 and the width of 427. Similarly, we label the ground-truth road regions manually.

### B. Horizon Detection

To evaluate the accuracy of the proposed weighted-sampling RANSAC-like approach in the improved v-disparity map, and validate the the effectiveness of using the u-disparity map, we compare it with the following methods: the conventional RANSAC method without using u-disparity map (RANSAC\_wu), the conventional RANSAC method with the usage of u-disparity map (RANSAC\_uu) and the weighted-sampling RANSAC-like approach without the usage of u-disparity map (RANSAC-like\_wu). To be fair, the inlier threshold of distance from a pixel to the fitted line,  $d_t$ , is set to 0.5 for all schemes mentioned above. Moreover, to evaluate the robustness of these schemes, the number of sampling are set to 100, 200, 500 and 1000, respectively.

TABLE II  
THE AUC VALUES FOR QUANTITATIVE COMPARISON.

Method	100	200	500	1000
RANSAC_wu	0.0374	0.0321	0.0302	0.0285
RANSAC_uu	0.5268	0.5938	0.6291	0.6425
RANSAC-like_wu	0.6473	0.6556	0.6697	0.6701
RANSAC-like_uu (our method)	<b>0.6849</b>	<b>0.6850</b>	<b>0.6877</b>	<b>0.6883</b>

The error between the estimation and the ground-truth horizon is defined as the absolute difference of vertical coordinates

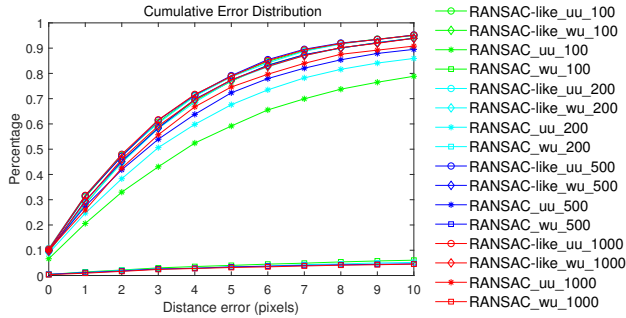


Fig. 12. Statistics of the horizon estimation accuracy.

of them. The quantitative comparison of these methods is shown in Fig. 12. The errors are represented in an 11-bin histogram (from 0 to 10 pixels) and the errors larger than 10 pixels are ignored. The horizontal axis of Fig. 12 represent the errors. When increasing the number of iterations, the accuracy of horizon detection of all methods changes. The conventional RANSAC method with the usage of u-disparity has the largest change in performance and the proposed method has the smallest change.

It indicates that the proposed method is more robust than the others in detecting horizon and can reach the optimal result within a small number of iterations. Moreover, both the usage of u-disparity and weighted-sampling method improves the accuracy of horizon detection. The performance of the conventional RANSAC method is the worst within equal number of iterations. The AUC (Area Under Curve) value of cumulative error distribution is listed in Table II. The proposed method within 1000 iterations achieves the best result with an AUC value of 0.6883. Within 100 iterations, our proposed method can obtain an AUC value of 0.6849, which is even better than the other methods that run within 1000 iterations.

### C. Vanishing Point Estimation

In this part, we compare the performance of the proposed vanishing point detection method with the other two popular methods: the Gabor-filter-based method (Garbor) [23] and the gLoG-filter-based method (gLoG) [22]. The threshold  $\alpha$  is set to 0.13 based on tuning on the selected validation images, which include various types of road scenes. Fig. 13 shows some visual comparisons of vanishing point estimation by the three methods. Thanks to the selection of vanishing point candidate region in the section III-C, our vanishing point estimation method is insensitive to misleading shadows and messy textures such as grass, bushes or trees, which are very challenging for traditional methods [23] and [22].

Fig. 14 illustrates the statistics of results for the three vanishing-point detection methods. The Euclidean distance between the estimate position and ground-truth one is computed as the error. The errors are distributed in a 31-bin histogram (from 0 to 30 pixels) and the errors greater than 30 are ignored. The horizontal axis of the histogram represents the error. It shows that, by our method, nearly 69% of images have got errors of less than or equal to 10 pixels, and more than 87% of images have got errors of less than or equal to 20 pixels.

However, by the Gabor based method, only 46% of images have detection errors of less than or equal to 10 pixels, and only 69% of images have detection errors of less than or equal to 20 pixels. By the gLoG based method, only 45% of images have detection errors of less than or equal to 10 pixels, and only 68% of images have detection errors of less than or equal to 20 pixels. Fig. 14 also provides the AUC values of cumulative error distribution by the three methods. Our method obtains the AUC value of 0.6831, which is the largest compared to the one by the Gabor based method with 0.5064 and the one by the gLoG based method with 0.5037. This indicates that the proposed vanishing point detection method is robust for different road condition and can provide accurate vanishing-point estimation for subsequent road detection.

### D. Road Detection

In our approach, the detected road region is indicated by the area enclosed by the detected boundaries and the last row of the image, which excludes the obstacles between the boundaries using the method in III-B. The weights of the cost functions are set based on tuning on the selected validation images, which include various types of road scenes. We choose the values which can obtain the best performance as the weights of the cost functions. With these values, we also obtained very promising results on other images. In general, we tune these weights around 0.2. Specifically, the the tuning step is set to 0.02 such that all the weights are tuned from 0.1 to 0.3, respectively. The default values for these weights are given as follows:  $f_g = 0.16$ ,  $f_o = 0.20$ ,  $f_f = 0.22$ ,  $f_{df} = 0.24$ ,  $f_{gd} = 0.16$ . Based on tuning on the selected validation images which include various types of road scenes, the parameters  $\alpha$  and  $\beta$  are set to 0.13 and 20, respectively, which can obtains the best performance.

To evaluate the performance of different road detection methods, four evaluation criteria are used to quantify the pixel-wise comparison, including PRE (precision), REC(recall), ACU (accuracy), and F-score. The definitions of the four criteria are formulated as follows:  $PRE = \frac{TP}{TP+FP}$ ,  $REC = \frac{TP}{TP+FN}$ ,  $ACU = \frac{TP+TN}{TP+FP+TN+FN}$  and  $F = \frac{2*PRE*REC}{PRE+REC}$ , where TP means true positive, TN meaning true negative, FP meaning false positive and FN meaning false negative.

To evaluate the influence from the detected vanishing points, we first compare three base methods in the perspective view. The three base methods adopt different vanishing point detection approaches, plus a vanishing point constrained Dijkstra-based road border segmentation scheme with the same composition of cost functions. In the first base method, the vanishing point is detected by the Gabor-based scheme[23], and only the gradient cost and orientation consistency cost are used in the Dijkstra road model. We denote this base method by Gabor\_g\_o. In the second base method, the vanishing point is detected by the gLoG-based scheme[22], and only the gradient cost and orientation consistency cost are used in the Dijkstra road model. We denote this base method by gLoG\_g\_o. In the third base method, the vanishing point is detected by the one proposed in this paper, and only the gradient cost and orientation consistency cost are used in the Dijkstra road

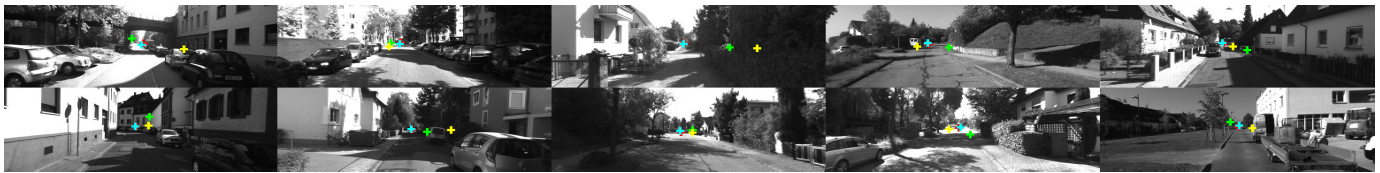


Fig. 13. Comparison of the vanishing point estimation. Red cross corresponds to the ground-truth. Green cross corresponds to the Gabor based method. Yellow cross corresponds to the gLoG based method. Cyan cross corresponds to our method.

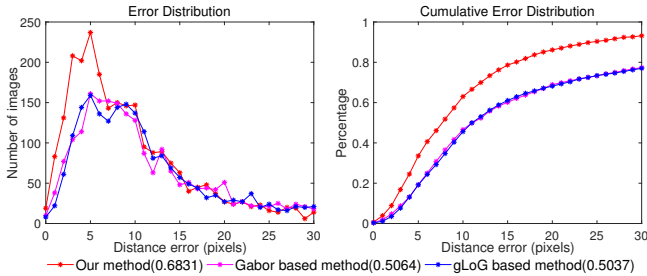


Fig. 14. Statistics of the road vanishing point detection accuracy: a comparison with the Gabor based method and gLoG based method. The numbers in the parentheses are the AUC values in the cumulative error distribution by each method, respectively.

model. We denote this base method by Stereo\_g\_o. Fig. 15 shows some visual comparison results of road region detection by the three methods. Table III illustrates the quantitative comparison results in details. With the highest AUC score of vanishing point detection, Stereo\_g\_o obtains also obtains the largest F-score. It shows that more accurate vanishing point detection method can lead to more accurate road region detection.

In addition, we evaluate the influence from different compositions of the cost functions. The weight items of the cost functions are set to zero one by one and other weight items remain the same. We compare our method with four base methods. These base methods are all based on our proposed stereo based vanishing point detection method. To facilitate understanding, our proposed road detection method is denoted as Stereo\_g\_f\_df\_gd\_o, and other five base methods are denoted as Stereo\_f\_df\_gd\_o (without gradient cost), Stereo\_g\_df\_gd\_o (without flatness cost), Stereo\_g\_f\_gd\_o (without disparity feature cost), Stereo\_g\_f\_df\_o (without gradient direction cost) and Stereo\_g\_f\_df\_gd (without link cost), respectively. Table III also illustrates the quantitative comparison results of these six methods in details. It shows that all the five cost functions improve the final road detection result. Among these cost functions, flatness cost, gradient direction cost and disparity feature cost have considerable influence on the road detection results. Compared with the result of Stereo\_g\_o, the use of flatness cost, disparity feature cost and gradient direction cost further increases the accuracy of road detection significantly.

Moreover, we also compare with several state-of-the-arts methods that have been independently tested on the KITTI-URBAN-ROAD dataset [51] in the bird’s-eye-view (BEV). These approaches include stereo-based [44], learning-based [45], [46], and Lidar-based [48], [49] road detection methods. Their results on the KITTI-URBAN-ROAD dataset are pub-

TABLE III  
STATISTICS OF ROAD DETECTION ACCURACY ON KITTI ODOMETRY DATASET.

Method	PRE	REC	ACU	F
Gabor_g_o	0.8014	0.8921	0.9621	0.8443
gLoG_g_o	0.7941	0.8799	0.9599	0.8348
Stereo_g_o	<b>0.8216</b>	<b>0.8946</b>	<b>0.9655</b>	<b>0.8565</b>
Stereo_f_df_gd_o	0.9089	0.9134	0.9795	0.9112
Stereo_g_df_gd_o	0.8208	0.9375	0.9692	0.8753
Stereo_g_f_gd_o	0.9340	0.8614	0.9770	0.8962
Stereo_g_f_df_o	0.8680	0.9118	0.9739	0.8894
Stereo_g_f_df_gd	0.9136	0.9067	0.9774	0.9101
Stereo_g_f_df_gd_o (Our Method)	<b>0.9137</b>	<b>0.9192</b>	<b>0.9807</b>	<b>0.9164</b>

licly available on the KITTI-road website [52]. Table IV shows the comparisons, where the performance of our approach is very close to that of the deepCNN based method [45]. Although our method is not the best among the compared methods in terms of either PRE or REC, the two indicators are both pretty high. Therefore, the F-measure is the highest except [45]. Our method is better than the stereo-based approach [44] and the Lidar-based approaches [48], [49].

TABLE IV  
COMPARISON WITH THE STATE-OF-THE-ARTS METHODS ON KITTI-URBAN-ROAD DATASET.

Method	PRE	REC	F
NNP[44]	0.8967	0.8968	0.8968
Up-Conv-Poly[45]	0.9400	0.9367	0.9383
FCN-LC[46]	0.9087	0.9072	0.9079
FTP[47]	0.9104	0.9220	0.9161
LidarHisto[48]	0.9306	0.8841	0.9067
FusedCRF[49]	0.8362	0.9344	0.8825
Our Method	0.9184	0.9221	0.9202

In addition, we compare our method with Up-Conv-Poly[45] on our selected 1643 images in the Oxford Robotcar Dataset in the perspective view. Up-Conv-Poly is a deep-learning based method which has the best performance on the KITTI-URBAN-ROAD dataset among the published methods. Table V shows the quantitative comparison results. As shown, our method obtains a better score. In practice, the generalization ability of the state-of-the-art deep-learning based methods is still limited by the training data. Although Up-Conv-Poly achieves very promising performance when the test images are similar to the training ones, which is the case for the KITTI-URBAN-ROAD dataset, however, its performance usually degrades a lot when the training road scenes are different from that of testing ones. Considering that image recognition performance is usually constrained by lighting conditions and weather changes heavily, we have to collect all-hour and

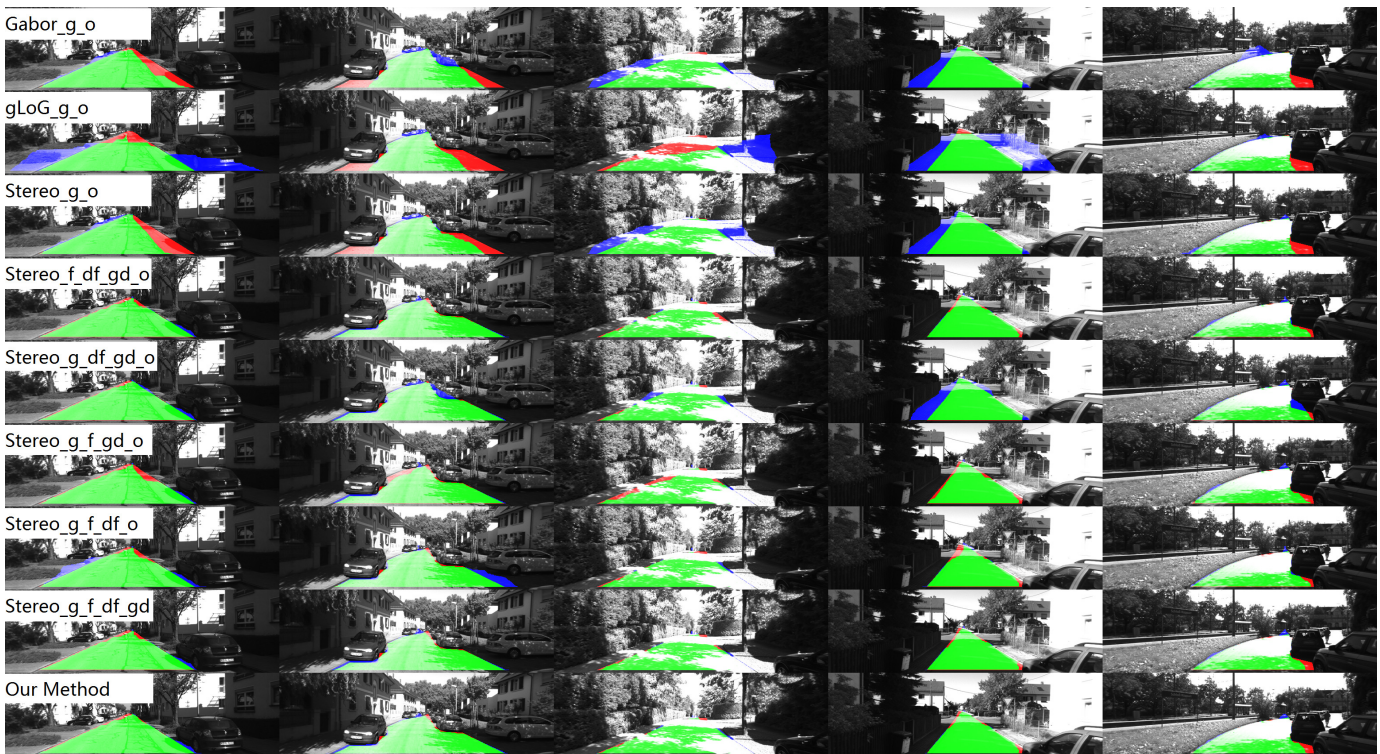


Fig. 15. Comparison of the road detection accuracy. Green region corresponds to true positive. Red region represent false negative. Blue region denote false positive.

four-season images for training, which is a very challenging mission. Although the performance of the proposed method is not the best, our method is quite robust on different sets of images and databases. Our method is a sort of data-driven approach, and needs little training procedure. It is a non-parametric approach in detecting road borders, which is robust to obstacles and shadows.

TABLE V  
COMPARISON ON OXFORD ROBOTCAR DATASET.

Method	PRE	REC	F
Up-Conv-Poly[45]	0.9021	0.8433	0.8717
Our Method	0.9604	0.8707	0.9133

The computational time is also an important factor to evaluate road detection method. For our proposed method, the computation cost mainly contains four parts: disparity computation, vanishing point detection, cost function computation and shortest-path map computation based on Dijkstra algorithm. Our method is implemented in single threaded C++ and tested on a general PC with 8GB memory and Intel Core i5-6500 CPU. The average runtime on the KITTI-URBAN-ROAD dataset (with the image size of  $1242 \times 376$ ) is about 0.25 seconds except computing the disparity map. The most time-consuming part of our framework is computing the disparity map because of the large size of images. Although our current implemented version could not run in real-time, with the use of GPU and parallel computing, we believe that our method can achieve real-time performance.

## VI. CONCLUSION

In this paper, we have proposed a novel vanishing point constrained Dijkstra model for road detection. First, the proposed method is based on the u- and v-disparity extracted from stereo camera to estimate the position of vanishing point. Then a vanishing-point constrained Dijkstra model is given to generate a minimum-cost map. By the Dijkstra algorithm, the optimal path from each point in the cost map to the vanishing point is obtained. The road boundaries correspond to two paths which have the minimum costs. Experiments on a challenging dataset of over 2000 frames show that the proposed method can achieve promising results.

## REFERENCES

- [1] Joel C. McCall and Mohan M. Trivedi. Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation. *IEEE Transactions on Intelligent Transport Systems*, 7(1):20-37, Mar. 2006.
- [2] Nicholas Apostoloff and Alexander Zelinsky. Vision in and out of vehicles: Integrated driver and road scene monitoring. *Int. Journal of Robotics Research*, 23(4-5):513-538, Apr. 2004.
- [3] Massimo Bertozzi, Alberto Broggi, and Alessandra Fascioli. Vision-based intelligent vehicles: State of the art and perspectives. *Robotics and Autonomous Systems*, 1:1-16, 2000.
- [4] ZuWhan Kim. Robust lane detection and tracking in challenging scenarios. *IEEE Trans. Intelligent Transportation Systems*, 9(1):16-26, Mar. 2008.
- [5] Heechul Jung, Junggon Min and Junmo Kim, An Efficient Lane Detection Algorithm For Lane Departure Detection, *IEEE Intelligent Vehicles Symposium (IV) June 23-26, 2013, Gold Coast, Australia*
- [6] Claudio Rosito Jung, Christian Roberto Kelber. Lane following and lane departure using a linear-parabolic model, *Image and Vision Computing*, 23 (2005) 1192-1202
- [7] Aharon Bar Hillel, Ronen Lerner, Dan Levi, Guy Raz, Recent progress in road and lane detection: a survey, *Machine Vision and Applications* (2014) 25:727-745

- [8] S. Yenikaya, G. K. Yenikaya, E. DeVen, Keeping the Vehicle on the Road: A Survey on On-Road Lane Detection Systems, *ACM Computing Surveys*, Vol. 46, No. 1, Article 2, October 2013.
- [9] M.A. Nasirudin, M.R. Arshad. A feature-based lane detection system using hough transform method. *International Symposium on Intelligent Transport System*, pp.166-169 (2007)
- [10] Joon Woong Lee and Un Kun Yi. A lane-departure identification based on LBPE, Hough transform, and linear regression. *Computer Vision and Image Understanding*, 99(3):359-383, 2005.
- [11] D. Khosla, Accurate estimation of forward path geometry using two-clothoid road model. *IEEE Symp. Intell. Veh.*, 154-159 (2002)
- [12] Y. Wang, L. Bai, M. Fairhurst. Robust road modeling and tracking using condensation. *IEEE Trans. Intell. Transp. Sys.* 9, 570-579 (2008)
- [13] Yue Wang, Eam Khwang Teoh, and Dinggang Shen. Lane detection using Catmull- Romspline. *IEEE International Conference on Intelligent Vehicles*, pp. 51-57 (1998)
- [14] Yue Wang, Eam Khwang Teoh, and Dinggang Shen. Lane detection and tracking using B-Snake. *Image and Vision Computing*, 22(4):269-280, 2004.
- [15] Y. Yagi, M. Brady, Y. Kawasaki, M. Yachida. Active contour road model for smart vehicle. *International Conference on Pattern Recognition*, 3819-3822 (2000)
- [16] R. Labayrade, D. Aubert, and J. P. Tarel. Real time obstacle detection in stereovision on non-fat road geometry through "vdisparity" representation. In *Proceedings of Intelligent Vehicle Symposium*, vol.2, pp. 646-651, June 2002.
- [17] Zhencheng Hu, Francisco Lamosa and Keiichi Uchimura, A Complete U-V-Disparity Study for Stereovision Based 3D Driving Environment Analysis. *Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling (3DIM)*, 2005.
- [18] Alexandru Iloie, Ion Giosan, Sergiu Nedevschi, UV disparity based obstacle detection and pedestrian classification in urban traffic scenarios, *IEEE International Conference on Intelligent Computer Communication and Processing*, 2004
- [19] C. Rasmussen. Grouping dominant orientations for ill-structured road following. in *Proc. IEEE Int. Conf. Computer Vision Pattern Recognition*, 2004, vol. 1, pp. 470-477.
- [20] C. Russmusen. Texture-based vanishing point voting for road shape estimation. in *Proc. British Machine Vision Conf.*, 2004.
- [21] C. Rasmussen. Roadcompass: Following rural roads with vision + lidar using vanishing point tracking. *Autonomous Robots*, vol. 25, no. 3, pp. 205-229, 2008.
- [22] Hui Kong, Sanjay E. Sarma, Feng Tang: Generalizing Laplacian of Gaussian Filters for Vanishing-Point Detection. *IEEE Trans. Intelligent Transportation Systems* 14(1): 408-418 (2013)
- [23] Hui Kong, Jean-Yves Audibert, Jean Ponce: General Road Detection From a Single Image. *IEEE Trans. Image Processing* 19(8): 2211-2220 (2010)
- [24] Hui Kong, Jean-Yves Audibert, Jean Ponce: Vanishing Point Detection for Road Detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 2009
- [25] M. Alsuwaiyel, *Algorithm Design Techniques and Analysis*, World Scientific, 1999
- [26] M. Nieto, J.A. Laborda, L. Salgado. Road environment modeling using robust perspective analysis and recursive Bayesian segmentation. *Machine Vision and Applications Journal*, vol.23, no.3, pp. 927-945, November 2011
- [27] Tsung-Ying Sun, Shang-Jeng Tsai and Vincent Chan. HSI Color Model Based Lane-Marking Detection. *IEEE Intelligent Transportation Systems Conference*, 2006.
- [28] Miguel Angel Sotelo, Francisco Javier Rodriguez, Luis Magdalena, Luis Miguel Bergasa and Luciano Boquete. A Color Vision-Based Lane Tracking System for Autonomous Driving on Unmarked Roads. *Autonomous Robots* 16, 95-116, 2004.
- [29] Jin-Wook Lee and Jae-Soo Cho. Effective lane detection and tracking method using statistical modeling of color and lane edge-orientation. *Fourth International Conference on Computer Sciences and Convergence Information Technology*, 2009
- [30] Kuo-Yu Chiu and Sheng-Fuu Lint, Lane Detection using Color-Based segmentation, *IEEE Intelligent Vehicles Symposium*, pp.706-711, June 2005
- [31] M. Aly. Real time detection of lane markers in urban streets. *IEEE Intelligent Vehicles Symposium*, pp.7-12, June 2008
- [32] Y.W. Seo, R.R. Rajkumar. Utilizing instantaneous driving direction for enhancing lane-marking detection. *IEEE Intelligent Vehicles Symposium*, pp.170-175, June 2014
- [33] S.N. Kang, S. Lee, J. Hur, S.W. Seo. Multi-lane detection based on accurate geometric lane estimation in highway scenarios. *IEEE Intelligent Vehicles Symposium*, pp.221-226, June 2014
- [34] Andra Petrovai, Radu Danescu and Sergiu Nedevschi. A stereo vision based approach for detecting and tracking lane and forward obstacles on mobile devices, *IEEE Intelligent Vehicles Symposium (IV)*, 2015
- [35] N. Benmansour, R. Labayrade, D. Aubert, S. Glaser. Stereovision-based 3D lane detection system: a model driven approach. *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pp.182-188, October 2008
- [36] Radu Danescu and Sergiu Nedevschi. Probabilistic Lane Tracking in Difficult Road Scenarios Using Stereovision. *IEEE Transactions on Intelligent Transportation Systems*, Vol.10, No.2, Jun 2009.
- [37] Rafael C. Gonzales and Richard E. Woods. *Digital image processing*. Nueva Jersey, 2008.
- [38] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient Large-Scale Stereo Matching, *Asian Conference on Computer Vision*, 2010.
- [39] S. Skiena, "Dijkstra's algorithm." *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*, Reading, MA: Addison-Wesley (1990): 225-227.
- [40] Eric N. Mortensen and William A. Barrett. "Interactive segmentation with intelligent scissors." *Graphical models and image processing* 60, no. 5 (1998): 349-384.
- [41] Alexandre X. Falcao, Jayaram K. Udupa, and Flavio Keidi Miyazawa. "An ultra-fast user-steered image segmentation paradigm: live wire on the fly." *IEEE transactions on medical imaging* 19, no. 1 (2000): 55-62.
- [42] Thiago Vallin Spina, Paulo AV de Miranda, and Alexandre Xavier Falcao. "Hybrid approaches for interactive image segmentation using the live markers paradigm." *IEEE Transactions on Image Processing* 23, no. 12 (2014): 5756-5769.
- [43] Boguslaw Obara, Mark Fricker, David Gavaghan, and Vicente Grau. "Contrast-independent curvilinear structure detection in biomedical images." *IEEE Transactions on Image Processing* 21, no. 5 (2012): 2572-2581.
- [44] X. Chen, K. Kundu, Y. Zhu, A. Berneshawi, H. Ma, S. Fidler and R. Urtasun. "3D Object Proposals for Accurate Object Class Detection." *NIPS* 2015.
- [45] G. Oliveira, W. Burgard and T. Brox. "Efficient Deep Methods for Monocular Road Segmentation." *IROS* 2016.
- [46] C. Mendes, V. Frmont and D. Wolf. "Exploiting Fully Convolutional Neural Networks for Fast Road Detection." *IEEE Conference on Robotics and Automation (ICRA)* 2016.
- [47] A. Laddha, M. Kocamaz, L. Navarro-Serment and M. Hebert. "Map-Supervised Road Detection." *IEEE Intelligent Vehicles Symposium Proceedings* 2016.
- [48] L. Chen, H. Kong and J. Yang. "A fast Lidar-guided scene analysis framework for autonomous driving." *IEEE Conference on Robotics and Automation (ICRA)* 2017, to appear.
- [49] L. Xiao, B. Dai, D. Liu, T. Hu and T. Wu. "CRF based Road Detection with Multi-Sensor Fusion." *Intelligent Vehicles Symposium (IV)* 2015.
- [50] A. Geiger, P. Lenz and R. Urtasun. "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite." *CVPR* 2012.
- [51] J. Fritsch, T. Kuehnl and A. Geiger. "A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms." *ITSC* 2013.
- [52] KITTI Road Benchmark: [http://www.cvlibs.net/datasets/kitti/eval\\_road.php](http://www.cvlibs.net/datasets/kitti/eval_road.php).
- [53] W. Maddern, G. Pascoe, C. Linegar and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset", *IJRR*, 2016.
- [54] O. Chum and J. Matas, "Matching with PROSAC Progressive Sample Consensus", *CVPR*, 2005.