Proceedings of the 33rd Chinese Control Conference
July 28-30, 2014, Nanjing, China

A Robust Vanishing Point Estimation Method for Lane Detection

Yuan Jun[1], Tang Shuming[1], Pan Xiuqin[2], Zhang Hong[2]

1. Institute of Automation, Chinese Academy of Sciences, Beijing 100190
E-mail: yuanjun2012@ia.ac.cn and shuming.tang@ia.ac.cn

2. School of Information Engineering, Minzu University of China, Beijing 100081
E-mail: amycun@163.com and zhanghong751103@muc.edu.cn

Abstract: Vanishing points are often used as constraints in lane detection or road following systems of intelligent vehicles. This paper proposes a new method for vanishing point estimation in consecutive frames based on computer vision. Parallel lines in the real world converge to vanishing points on an image plane, caused by the perspective projection. According to the duality between points and lines, estimation of vanishing points can be converted to a problem of line parameter estimation in a parameter space. Firstly, straight lines are detected from an extracted edge map of a road image by the Progressive Probability Hough Transform (PPHT) incorporated with gradient orientation constraints. Then, vanishing points are estimated via the Maximum A Posteriori (MAP) estimate, integrating information at the current frame and the vanishing point estimated at the previous frame into a probabilistic framework. For the detected lines are noisy, a weight is put on each line to indicate the probability of being an inlier. But the weights are unknown, which are regarded as hidden variables here. Thus the Expectation Maximum (EM) algorithm is adopted to solve the MAP problem with hidden variables. Experimental results show the efficiency and robustness of the proposed method.

Key Words: Vanishing point estimation, lane detection, MAP estimate, EM algorithm

1 Introduction

A lane detection system is an indispensable part of lane departure warning systems (LDW), driver assistance systems (DAS) and autonomous driving vehicles. The relative position between lane boundaries and vehicles can be acquired by lane detection for vehicle navigation, lateral control, and lane departure warning.

Parallel road elements in the real world, such as lane markings or road borders, converge to vanishing points (VPs) on an image plane, caused by the perspective projection. Therefore, VPs are often used as constraints in lane or road detection [1-4]. Wang *et al.* [1] estimated VPs to calculate control points of the B-spline curve. Zhou *et al.* [2] took a VP as a part of the lane model. VP constraints are utilized to remove detected non-lane markings by Liu [3].

In principle, existing approaches to VP estimation for lane or road detection can be categorized to two classes: edge-based methods [1, 5] and texture-based methods [6-9]. A general process of edge-based methods is as follows: i) an edge map is extracted by a certain edge detector. ii) straight lines are detected from the edge map. iii) a VP can be obtained by a voting scheme. For instance, in [1], an edge map was firstly extracted by the Canny algorithm, and then Hough transform was performed to detect straight lines, finally the intersections of any pair of the detected lines voted for VPs on another Hough space. In general, due to character of computational efficiency, edge-based methods can be applied to real-time systems. However, edge-based methods only function well for structured roads with clear lane markings or distinct borders, while they usually fail for unstructured roads without lane markings, definite borders or regular shape. Hence, texture-based methods are developed to eliminate such limitation of the edge-based methods. Texture-based methods are based on the assumption that there are some parallel textures with obvious orientation on the ground, such as ruts left by passed cars. Generally, this kind of methods work as follows: i) dominant orientations of textures are obtained by convolving a road image with certain filters. ii) a VP is voted through a voting scheme. Such texture-based method was first proposed by Rasmussen [6]. He adopted a group of Gabor wavelet filters with 36 orientations under different scales to get the dominant orientation of each pixel, and then each pixel voted for VP candidates above it along its orientation, finally the candidate with the most votes won. Whereas this method tends to favor points those are upper in an image. Kong *et al.* [7] proposed a locally adaptive soft-voting scheme to solve the problem. Miksik [8] decomposed each Gabor wavelet into a linear combination of Harr-like box functions to achieve a faster computation speed. To achieve the same purpose, Moghadam *et al.* [9] used joint activities of only four Gabor filters to estimate local dominant orientation precisely. There are some other methods improved. However, texture-based methods are still so time-consuming that they are not suitable for real-time applications.

To perform real-time lane detection under structure environment, we propose a fast and robust VP estimation method similar with edge-based methods mentioned above. It's different in that we convert VPs estimation in an image plane to a line parameter estimation problem in a parameter space, instead of using a voting scheme in the third stage of general edge-based methods. Hence, we don't need to consider the impact of a voting interval size and possibility that a real VP would be beyond the settled voting space. Additionally, our method can achieve more accurate coordinate estimation of a VP. The method in [10] is with similar idea with ours. But it hasn't token the noise among detected lines into consideration, which affects its estimation accuracy.

* This work was supported in part by National Natural Science Foundation of China under Grants 91120008 and 61220002.

The remained structure of this paper is organized as follows: Section 2 describes our method including line detection by the PPHT with orientation constraints and the process of VP estimation. Section 3 shows the experimental results. Finally, we make a conclusion in Section 4.

2 A Proposed Vanishing Point Estimation Method

At first, straight lines are detected from an extracted edge map of a road image by the Progressive Probability Hough Transform (PPHT) [11]. In the PPHT, gradient orientation information of edge points is incorporated to relieve the impact of noise. A portion of detected lines come from edges of lane markings, which are parallel in the real world and converge to VPs on an image plane, caused by the perspective projection. According to the duality between points and lines, the estimation of VPs is converted to a problem of line parameter estimation in a parameter space. Then, a VP is estimated via the Maximum A Posteriori (MAP) estimate, integrating the information at the current frame and the VP estimated at the previous frame into a probabilistic framework.

2.1 Line Detection

Edges are extracted from each captured road image by an adaptive threshold Canny [12]. Then, line detection is performed on the generated edge map. The standard Hough Transform (SHT) [1, 5] is a widely used method in line detection. The SHT is based on the duality between pixels in the image space and lines in the parameter space. However, all pixels in the image need to vote in the SHT, which is not computationally efficient.

Compared with the SHT, the PPHT uses only a fraction of supporting points to vote. In every loop, a pixel is selected randomly to vote. Once a line segment meets given constraints, the other pixels in the line will be removed. These constraints include the minimum number of votes, the maximum gap between two pixels and the minimum length. Thus, the PPHT minimizes the amount of computation needed to detect lines reliably.

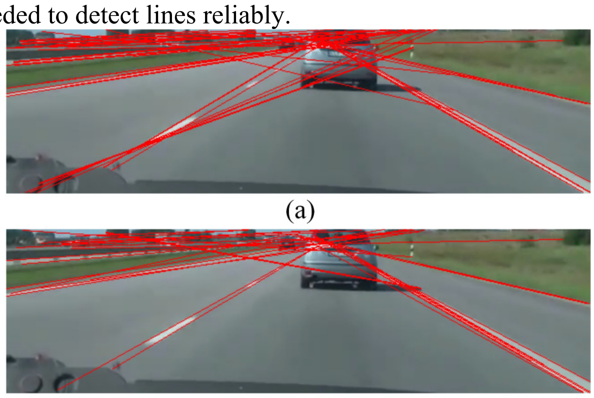

Fig. 1: Line extraction results. (a) is the result using the original PPHT; (b) is the result using the PPHT with gradient orientation constraints.

If a pixel is on the edge of a lane, the gradient orientation of the pixel should be perpendicular to the tangent direction of the lane. Based on this fact, we integrated the gradient orientation constraints into the PPHT. Therefore, a pixel only votes the accumulators which are roughly in the corresponding angle range. Then, the probability of false-alarm caused by noise is reduced, as shown in Fig. 1. Additionally, less time will be consumed owing to that the voting range of each pixel is narrowed. The gradient orientations don't need to be calculated again here, because they have been calculated in the intermediate procedure of edge detection. Before vanishing point estimation, we remove lines which are completely or approximately horizontal. Then, a part of noisy lines are filtered out by a validation gate with a rectangle shape. The center of the gate is set as the VP detected at the previous frame. Specifically, a line would be removed if it has no intersection with any side of the gate. It's based on the assumption that a VP can moves only a little between two consecutive frames.

2.2 Our Proposed Vanishing Point Estimation

A VP is estimated using the detected lines after the line detection process above. Here, one line in the image space *X-Y* is represented as follows:

$$y = kx + b\ , \tag{1}$$

where the parameters k and b are the slope and the intercept respectively.

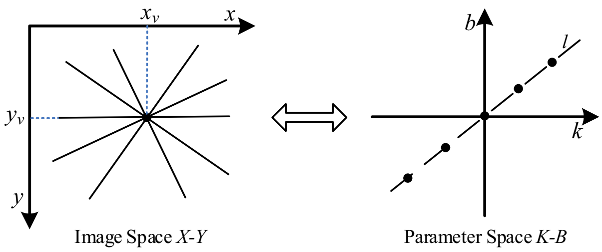

Fig. 2: Illustration of the duality between an intersection in the image space *X-Y* and a line in the parameter space *K-B*.

According to the duality between points and lines, we know that lines in the image space *X-Y* are corresponding to points in the parameter space *K-B*, as shown in Fig. 2. If these lines intersect at one point (x_v, y_v) in the image space *X-Y*, their corresponding points are on the same line l in the parameter space *K-B*. The line l is parameterized by x_v and y_v as follows:

$$b = -x_v \cdot k + y_v\ . \tag{2}$$

Lines detected from lane marking edges intersect at the VP. Hence, the VP estimation in the image space *X-Y* can be converted to a problem of line parameter estimation in the parameter space *K-B*.

There are some outliers in the parameter space. Thus, it's necessary to adopt a robust line fitting method to estimate the parameters accurately, i.e., the coordinate of a VP. Here, we adopt a probabilistic framework to estimate the parameters of the line:

$$\arg\max_{\varphi_t} p(\varphi_t | D_t, \varphi_{t-1})\ , \tag{3}$$

where $\varphi_t = [x_{v,t}, y_{v,t}]^T$ is the line parameters in the parameter space, D_t is the set of N_t points in the parameter space i.e. $D_t = \{(k_{t,1}, b_{t,1}), (k_{t,2}, b_{t,2}), \cdots (k_{t,N_t}, b_{t,N_t})\}$, and t is the frame index. Namely, we consider that the current VP is only related to the detected lines at the current frame and the VP at the previous frame. Applying Bayes rule, we turns the problem described in (3) to a Maximum A Posteriori (MAP) estimate problem:

$$\arg\max_{\varphi_t} p(\varphi_t|\varphi_{t-1})\cdot p(D_t|\varphi_t) \ , \qquad (4)$$

where $p(\varphi_t \mid \varphi_{t-1})$ is the prior distribution, $p(D_t|\varphi_t)$ is the likelihood.

The calculation of the prior is based on the assumption that the VP stays almost unchanged in two adjacent frames. This assumption is relatively reasonable in the real world since a road or a vehicle position can't change abruptly in a very short time slice. If it is the first frame, i.e. $t=1$, the prior is equal to one. Otherwise, information of the previous frame can be used in the estimation of the VP at the current frame. Assuming the prior is a Gaussian distribution, it can be represented as follows for $t>1$:

$$\begin{aligned} p(\varphi_t|\varphi_{t-1}) &= \frac{1}{2\pi|\Sigma|^{1/2}}\exp[-(\varphi_t-\varphi_{t-1})^T\Sigma^{-1}(\varphi_t-\varphi_{t-1})] \\ &= \frac{1}{2\pi|\Sigma|^{1/2}}\exp[-\lambda_1(x_{v,t}-x_{v,t-1})^2-\lambda_2(y_{v,t}-y_{v,t-1})^2] \end{aligned} \ , \qquad (5)$$

where φ_{t-1} is the mean, the covariance matrix is as follows:

$$\Sigma = \begin{bmatrix} 1/\lambda_1 & 0 \\ 0 & 1/\lambda_2 \end{bmatrix} . \qquad (6)$$

For each point $d_{t,i}$ in D_t are independent, the likelihood can be defined as follows:

$$p(D_t|\varphi_t) = \prod_{i=1}^{N_t}[p(d_{t,i}|\varphi_t)]^{w_{t,i}} \ , \qquad (7)$$

where $w_{t,i}$ is the weight of point $d_{t,i}$ and $\sum_{i=1}^{N_t} w_{t,i} = 1$. Each weight value represents the possibility of each point being an inlier. Assuming the deviation between each point and the line obeys the Gaussian distribution, the likelihood of each point is represented as below:

$$p(d_{t,i}|\varphi_t) = \frac{1}{\sqrt{2\pi}\sigma}\exp[-(b_{t,i}+x_{v,t}\cdot k_{t,i}-y_{v,t})^2/\sigma^2] \ , \qquad (8)$$

where the mean is zero, the standard deviation is σ. Then we can get:

$$p(D_t|\varphi_t) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-E(D_t,\varphi_t)/\sigma^2) \ , \qquad (9)$$

where $E(D_t,\varphi_t)$ is the weighted residual sum of squares, defined by:

$$E(D_t,\varphi_t) = \sum_{i=1}^{N_t} w_{t,i}\cdot(b_{t,i}+x_{v,t}\cdot k_{t,i}-y_{v,t})^2 \ . \qquad (10)$$

We attempt to relieve or even eliminate influence of outliers by w. To recognize outliers, the line parameters, i.e. φ_t, are needed. To calculate φ_t precisely, we need to know which points are outliers. This is a chicken-and-egg situation. Here, w is considered as a hidden variable. Therefore, the EM algorithm will be adopted to solve the MAP problem with a hidden variable.

After we take the negative logarithm and remove the constant terms, the MAP estimate problem described in (4) is equivalent to minimize the function as follows:

$$f(x_{v,t},y_{v,t}) = \begin{cases} \alpha\sum_{i=1}^{N_t} w_{t,i}\cdot(b_{t,i}+x_{v,t}\cdot k_{t,i}-y_{v,t})^2 \ , \quad t=1 \\ \lambda_1(x_{v,t}-x_{v,t-1})^2+\lambda_2(y_{v,t}-y_{v,t-1})^2 \\ +\alpha\sum_{i=1}^{N_t} w_{t,i}\cdot(b_{t,i}+x_{v,t}\cdot k_{t,i}-y_{v,t})^2 \ , \ t>1 \end{cases} \ , \qquad (11)$$

where $\alpha = 1/\sigma^2$. The iterative solving process applying the EM is as follows:

Step 1: Choose the parameters estimated at frame $t-1$ as the initial values of the parameters to estimate at frame t, i.e. $(x_{v,t}{}^{(0)}, y_{v,t}{}^{(0)}) = (x_{v,t-1}, y_{v,t-1})$. Taking the results at the previous frame to initialize can make the calculating process converge faster. The initial values are not need to be set when $t=1$.

Step 2: The E step. Calculate weight values in iteration loop j, using the calculated parameters in the iteration loop $j-1$ for $t=1$ *and* $j>1$ or $t>1$ *and* $j\geq 1$ as follows:

$$w_{t,i}{}^{(j)} = \frac{\exp[-(b_{t,i}+x_{v,t}{}^{(j-1)}\cdot k_{t,i}-y_{v,t}{}^{(j-1)})^2/\sigma^2]}{\sum_{n=1}^{N_t}\exp[-(b_{t,n}+x_{v,t}{}^{(j-1)}\cdot k_{t,n}-y_{v,t}{}^{(j-1)})^2/\sigma^2]} \ . \qquad (12)$$

Specially, $w_{1,i}{}^{(1)} = 1/N_t$, i.e. the weights are equal in the first iteration loop when a current frame is the first one.

Step 3: The M step. Calculate the optimal parameters in iteration loop j using the calculated weights in Step 2. Once the weights are given, the calculation of the optimal parameters is a convex quadratic optimization problem. Thus, we can get the parameters in iteration loop j by solving the equations below:

$$\begin{cases} \dfrac{\partial f}{\partial x_{v,t}} = 0 \\ \dfrac{\partial f}{\partial y_{v,t}} = 0 \end{cases} . \qquad (13)$$

Then, we can get:

$$\begin{bmatrix} x_{v,t}{}^{(j)} \\ y_{v,t}{}^{(j)} \end{bmatrix} = A^{-1}\begin{bmatrix} \lambda_1 x_{v,t-1}(1-\delta(t-1)) - \alpha\sum_{i=1}^{N_t} w_{t,i}{}^{(j)} b_{t,i} k_{t,i} \\ \lambda_2 y_{v,t-1}(1-\delta(t-1)) + \alpha\sum_{i=1}^{N_t} w_{t,i}{}^{(j)} b_{t,i} \end{bmatrix} , \qquad (14)$$

where,

$$A = \begin{bmatrix} \alpha\sum_{i=1}^{N_t} w_{t,i}{}^{(j)} k_{t,i}{}^2 + \lambda_1(1-\delta(t-1)) & -\alpha\sum_{i=1}^{N_t} w_{t,i}{}^{(j)} k_{t,i} \\ -\alpha\sum_{i=1}^{N_t} w_{t,i}{}^{(j)} k_{t,i} & \alpha+\lambda_2(1-\delta(t-1)) \end{bmatrix}$$

and δ is the unit impulse sequence, i.e. $\delta(t-T)=1$ if $t=T$ and $\delta(t-T)=0$ if $t\neq T$.

Step 4: Stop the iteration if j reaches the given maximum iteration number or $x_{v,t}{}^{(j)}$ and $y_{v,t}{}^{(j)}$ satisfy the condition below:

$$\left|x_{v,t}{}^{(j)} - x_{v,t}{}^{(j-1)}\right| \leq \varepsilon_x \ and \left|y_{v,t}{}^{(j)} - y_{v,t}{}^{(j-1)}\right| \leq \varepsilon_y \ , \qquad (15)$$

where ε_x and ε_y are given thresholds of convergence. The result is $(x_{v,t}, y_{v,t}) = (x_{v,t}^{(j)}, y_{v,t}^{(j)})$. Otherwise, go to Step 2.

Finally, the calculated line parameters in the parameter space are the location of the VP in the image space. Hence, the VP in each frame can be estimated through the above method, even though noisy lines exist.

3 Experimental Results

Our proposed VP estimation method is implemented with C++. Then it's tested on a video clip [13] and the Caltech Lanes dataset [14]. To reduce the amount of calculation, each frame is cropped. The resolution of each frame of the two datasets turns to 640×180 and 640×320, respectively. The testing platform is a laptop with Intel Pentium dual-core CPU of 2.1 GHz and 2G memory. The average time cost by VP estimation for a single frame is about 28ms and 54ms on the two datasets respectively. It's necessary to declare that the time cost by edge extraction and line detection is already included. The average frame rates of our lane detection algorithm on the two datasets are about $30\,fps$ and $14\,fps$, respectively. Noticing the resolutions are relatively high, we can accelerate the process just by image downsampling.

Our method can locate the VPs correctly, even though there are some noisy lines detected. Some VP estimation results on the first dataset are shown in Fig. 3. Each VP is marked by a red cross. The blue lines are the lines used to estimate VPs. Definitely, (a) shows obvious interference from a car, (b) and (c) are road areas with lane merging and lane separating respectively, (d) displays some noise from an arrow-shaped lane marking, (e) is a road area with curve, and (f) is a road area with serve shadows. The shown robustness is owing to two factors. On the one hand, we have considered the probability of a line being an inlier. On the other hand, we have token the result at the previous frame as prior information, which is essentially a tracking process.

To evaluate our method quantitively, the ground-truth VPs are labeled manually. In Dataset 1, i.e. the video clip, one in every twenty-one frames is labeled, with 221 frames labeled totally. There are 213 frames labeled in the cordova1 part of the Caltech Lanes dataset, which is referred to Dataset 2 in ensuring paragraphs. We evaluate the proposed method on the labeled datasets, compared with another two methods, i.e. Liu's method in [10] and the method using Hough transform as a voting scheme. The distance between a detected VP and the ground-truth one is took as the error measurement. The average error on each dataset is listed in Table 1. The average errors of our method on both datasets are the minimum, which demonstrates that our method outperforms another two on average precision.

Table 1: Average errors (in pixels)

	Hough Transform	Liu's Method [10]	Our method
Dataset 1	12.20	7.92	5.93
Dataset 2	11.54	6.03	4.40

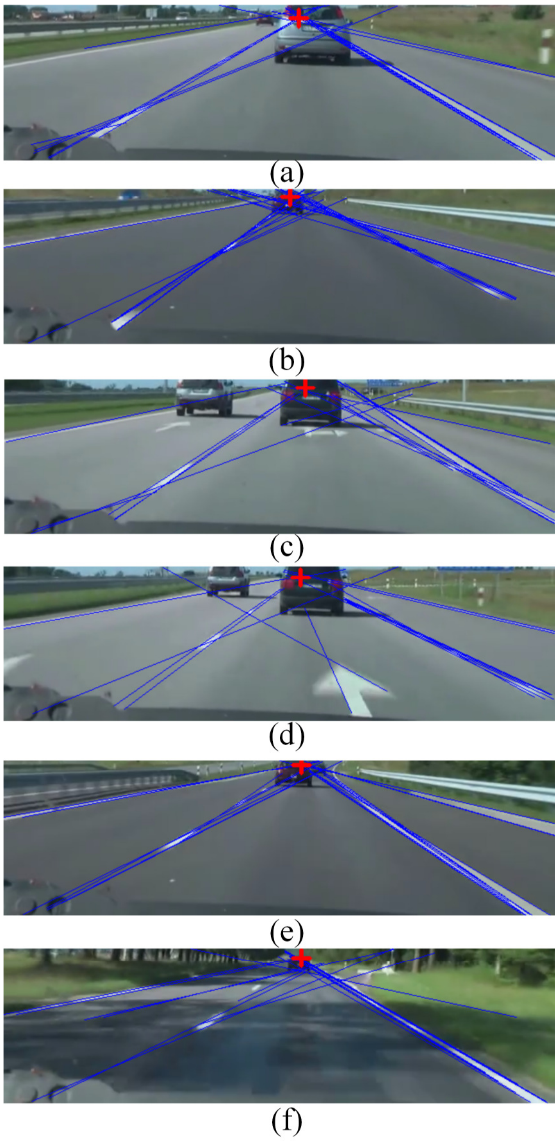

Fig. 3: VP estimation results.

For a deeper comparison, accuracy [15] of each method is calculated under different distance threshold. Namely, a detected VP is regarded as a correct one when the distance deviation between it and the ground-truth is lower than a threshold. The statistics of accuracy on the two datasets are plotted in Fig. 4 and Fig. 5. It's shown that our method can achieve the highest accuracy under low or high threshold. With the threshold set to 15 pixels, the accuracy of our method on both dataset reaches above 90%. It converges to approximate 100% when the threshold is set to 18 pixels. The statistic data prove further that the accuracy of our method exceed the another two. The performance of our method is better than Liu's method because we have token the probability of a line being an inlier into consideration. Whereas Liu's method simply took the number of edge pixels a line passes through as a weight. Thus, the weight of a line detected from a long shadow border may even higher than that of a line from a dash lane marking. Both our method and Liu's method achieved better performance than the Hough transform, owing to the tracking process.

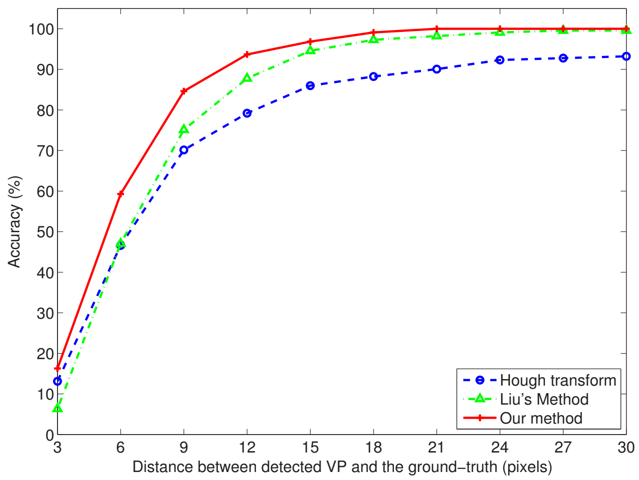

Fig. 4: Accuracy of VP detection on Dataset 1.

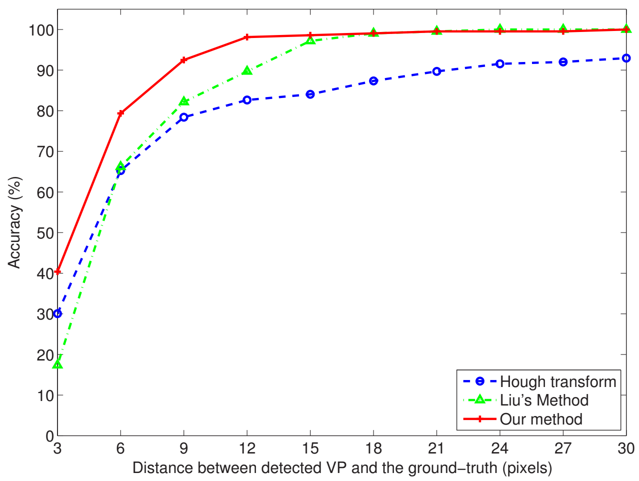

Fig. 5: Accuracy of VP detection on Dataset 2.

The estimated VPs are also used as constraints in our lane detection method, i.e., the distance between a line from lane marking edges and a VP is supposed to be within a certain limit. Lines those are beyond the limit are removed. Finally, the most probable line pair is picked out to model the lane, based on local color feature and historical information. Fig. 6 and Fig. 7 show some VP estimation and lane detection results under challenging scenarios from the first dataset and the second dataset, respectively. Each VP is still marked by a red cross, and lane boundaries are marked by blue lines. In Fig. 6, (a) and (b) are the road areas with lane separating and lane merging respectively, and (c)-(d) are road under severe shadows. In Fig. 7, (a)-(d) are road areas with lane marking wear, curve, one side of lane markings missing, and shadows, respectively.

The above experimental results have shown the efficiency and robustness of both our proposed vanishing point estimation method and lane detection method.

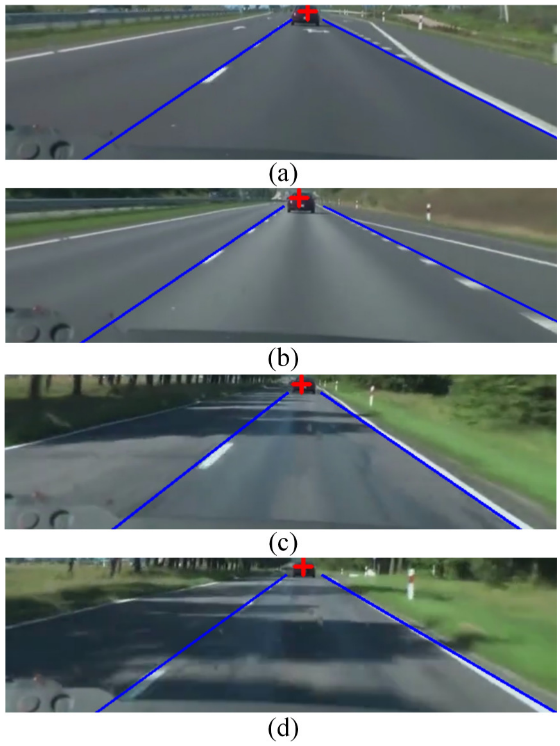

Fig. 6: VP estimation and lane detection results under challenging environment (1).

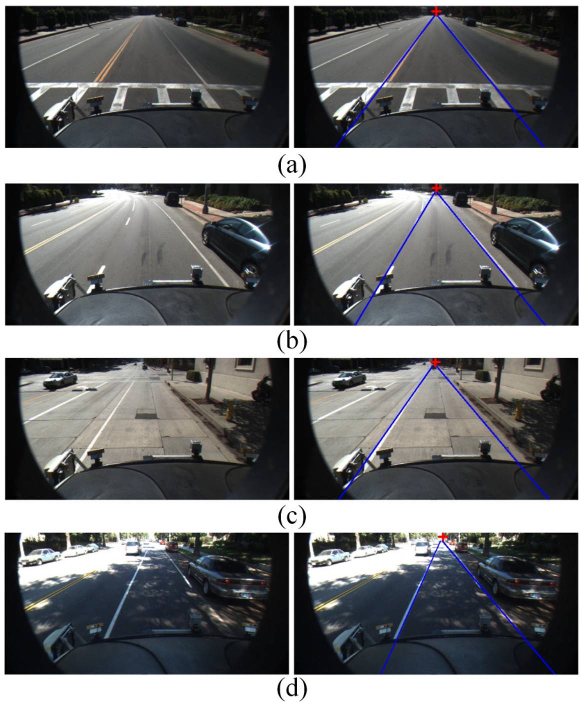

Fig. 7: VP estimation and lane detection results under challenging environment (2).

4 Conclusions

We have proposed a fast and robust vanishing point (VP) estimation method which functions well in our lane detection system. The estimation of a VP is converted to a problem of line parameter estimation in the parameter space. Definitely, a VP is estimated via the Maximum A Posteriori (MAP) estimate, integrating information at the current frame and the VP at the previous frame into a probabilistic framework. To deal with noise, a weight is put on each line to indicate the probability of being an inlier, which is regarded as a hidden variable here. Thus, the Expectation Maximum algorithm is adopted to solve the MAP problem with hidden variables. Experimental results have shown the efficiency and robustness of the proposed method.

Owing to the assumption that the prior of a VP is a Gaussian distribution, the location of the VP may not be estimated precisely when the orientation of the vehicle head or the road direction change abruptly. Therefore, a more general assumption on the prior or another tracking scheme will be adopted to tackle the problem in the future. With regard to lanes with sharp curves, the VP may be not unique at one frame. Thus, an effective method needs to be developed to estimate multiple VPs for curve lanes.

References

[1] Y. Wang, E.K. Teoh and D. Shen, Lane detection and tracking using B-Snake, *Image and Vision computing*, 22(4): 269-280, 2004.

[2] S. Zhou, Y. Jiang, and *et al.*, A novel lane detection based on geometrical model and gabor filter, in *Proceedings of Intelligent Vehicles Symposium*, 2010: 59-64.

[3] W. Liu, S. Li, and X. Huang, Extraction of lane markings using orientation and vanishing point constraints in structured road scenes, *International Journal of Computer Mathematics*, (ahead-of-print):1-15, 2013.

[4] Yifei Wang, NaimDahnoun, and Alin Achim, A novel system for robust lane detection and tracking, *Signal Processing*, 92(2): 319-334, 2012.

[5] G. Zhang, N. Zheng, and *et al.*, An efficient road detection method in noisy urban environment, in *Proceedings of Intelligent Vehicles Symposium*, 2009: 556-561.

[6] C. Rasmussen, Grouping dominant orientations for ill-structured road following, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2004: 470-477.

[7] H. Kong, J.-Y. Audibert, J. Ponce, Vanishing point detection for road detection, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009: 96-103.

[8] O. Miksik, Rapid vanishing point estimation for general road detection, in *Proceedings of IEEE Conference on Robotics and Automation*, 2012: 4844-4849.

[9] P. Moghadam, J.A. Starzyk, W.S. Wijesoma, Fast Vanishing-Point Detection in Unstructured Environments, *IEEE Trans. on Image Processing*, 21(1): 425-430, 2012.

[10] H. Liu, Z. Guo, J. Lu and *et al.*, A Fast Method for Vanishing Point Estimation and Tracking and Its Application in Road Images, in *Proceedings of International Conference on ITS Telecommunications*, 2006:106-109.

[11] J. Matas, C. Galambos, J. Kittler, Robust Detection of Lines Using the Progressive Probabilistic Hough Transform, *Computer Vision and Image Understanding*, 78(1): 119-137, 2000.

[12] T. Sun, S. Tang, and *et al.*, A robust lane detection method for autonomous car-like robot, in *Proceedings of 4th IEEE International Conference on Intelligent Control and Information Processing*, 2013: 373-378.

[13] https://code.google.com/p/opencv-lane-vehicle-track/.

[14] http://vision.caltech.edu/malaa/datasets/caltech-lanes/.

[15] H. Kong, S.E. Sarma, and F. Tang, Generalizing Laplacian of Gaussian Filters for Vanishing-Point Detection, *IEEE Trans. On Intelligent Transportation Systems*, 14(1): 408-418, 2013.