Fast Road Vanishing Point Detection Based on Modified Adaptive Soft Voting

Xue Fan, Cheng Deng, Yawar Rehman, and Hyunchul Shin
Department of Electronics and Communication Engineering
Hanyang University, Ansan, Korea
e-mail: {fanxue, dengcheng, yawar}@digital.hanyang.ac.kr, shin@hanyang.ac.kr

***Abstract*— Detecting the vanishing point from a single image is a challenging task since the information contained in the input image which can be used to detect the location of vanishing point is very limited. In this paper, we propose a framework for vanishing point detection based on the Modified Adaptive Soft Voting (MASV) scheme. First of all, the input image is convolved with the generalized Laplacian of Gaussian (gLoG) filters, which are used to estimate the texture orientation map. Then, MASV scheme is used to get accurate voting map. Finally, peak identification is performed on the voting map to locate the vanishing point. In addition, a scaling method for voting map computation is proposed to further accelerate the vanishing point detection algorithm. Through experiments, we show that the proposed algorithm is about 10 times faster and outperforms by 4.64% on an average than the complete-map based gLoG, which is the state-of-the-art method.**

Keywords-vanishing point detection; gLoG; MASV; voting map.

I. INTRODUCTION

Automatic driver assistant systems and driverless vehicles have been the focus of attention for many computer vision researchers over the last twenty years [1]. There are numerous researchers who devoted themselves for developing Autonomous Vehicle Navigation Systems (AVNS) in either structured [2][3], urban environments [4] or unstructured roads [5]. Road detection is one of the crucial parts of the AVNS. Considering the continuously changing background, traffic conditions, and road types, robust road detection using a monocular vision system is a challenging problem. Many vison-based road detection methods have been proposed. Among all road detection methods, vanishing point constrained road detection schemes have given promising results in detecting both off-road areas [5][6] and urban roads [7][8].

The current method of road vanishing point detection can be generalized into three main categories: edge-based methods [2][3], prior-based methods [9][10], and texture-based method [5][11][12].

In edge-based vanishing point detection methods, two or more dominant straight lines (segments), which correspond to road borders or markings, are detected by Hough transform or random sampling and consensus, and the vanishing point is detected as the intersection of these straight lines (segments). For structured roads with well-paved lanes, edge-based approaches can be used for real-time application due to their accuracy and computation efficiency. While for unstructured roads without edges or contrasting local characteristics, their performance is really limited.

Recently, some prior based techniques have been proposed. They intend to integrate contextual 3D information with low-level features in order to improve the detection performance. Such weak contextual cues include a 3D scene layout, 3D road stages, temporal road cues, and so on. It is really difficult to apply these prior based methods in real-time and practical situations. Besides, all these methods assume that the road is structured and well-paved. In order to overcome this limitation texture-based approach is proposed [5][12][13], which can accurately detect vanishing point of both well-paved roads and unstructured off-roads. Texture-based vanishing-point detection methods apply a bank of oriented filters to compute the texture orientation of each pixel. Then each pixel votes for the candidate vanishing point through a pre-defined voting scheme. Either the local soft voting scheme proposed by Kong et al. [5], or the global voting scheme proposed in [6][11] is time-consuming and cannot meet the requirement of real-time applications.

In this paper, we propose a texture-based method for detecting vanishing point from a single image. Specially, the contributions of this paper are as follows. First, a scaling method for fast voting map computation is proposed. We first down-sample the input image, and the voting map is computed for the down-sampled image. Then, the final voting map is up-sampled for vanishing point detection. As a result, the detection time is significantly reduced. Second, we propose MASV scheme for the purpose of accurate and effective voting map generation. Instead of using exponential function based soft-voting in [12], we use Gaussian function for weighted voting [14], which is more adaptive and robust. In addition, we define a voting radius R_v to limit the voting region of each pixel and to reduce redundant voting.

The rest of this paper is organized as follows. A brief review of the related work is presented in Section II. The proposed MASV scheme based vanishing point detection is explained in Section III. Experimental results are shown in Section IV. Finally, the conclusions are summarized in Section V.

II. RELATED WORK

A vanishing point is a point in perspective images to which parallel lines converge. It plays a leading role in road detection.

Most of the existing edge-based vanishing point detection algorithms rely on three steps [2][3]. The first step performs edge detection on the input image in order to extract the most dominant edges such as road borders or lane markings. The next step is to determine whether there are any line segments in the image. Once all the line segments are identified, a voting procedure is applied to find the intersections of the lines.

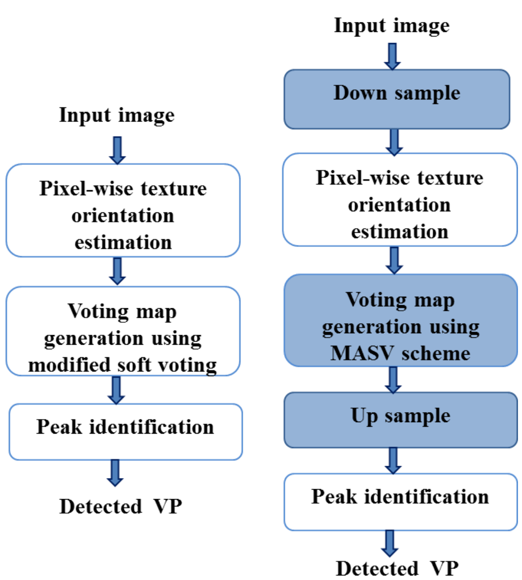
Wu et al. [10] proposed a global perspective structure matching (GPSM) scheme based on an image retrieval technique to identify the best candidate images in an image database, and then use the pre-labeled vanishing points of the best candidate images as the initial estimation of input image's vanishing point, and finally, a probabilistic model of vanishing point is used to refine the location of vanishing point. For these prior-based methods, not only a large scale image or video database is necessary in order to make these prior-based methods robust to various imaging conditions, road types and scenarios, but the training algorithm is also very important, and not to mention laborious manual label works for the training stage.

Texture-based methods are proposed to overcome the drawback of edge detection based and prior-based vanishing point detection methods. Firstly, a bank of oriented filters is applied, such as Gabor filter banks [13] and steerable filter banks [15], to estimate the dominant texture orientation of each pixel and generate the texture orientation map. Then, pixel-based voting is performed to obtain the voting map. Each pixel votes for the candidate vanishing point through a pre-defined voting scheme. Finally, vanishing point is detected by using peak point identification.

III. PROPOSED VANISHING POINT DETECTION APPROACH

In this part, our improved method will be explained in detail. Figure 1 shows the workflow comparison of our proposed method and the complete-map based gLoG vanishing point detection method [12]. The shaded blocks show the new contributions of our method.

A. Generalized Laplacian of Gaussian (gLoG) Filter



(a) The gLoG method (b) Our method

Figure 1. The framework of our proposed method and the complete-map based gLoG method.

The standard 2-D Gaussian function and the generalized 2-D Gaussian function [12] are defined as in (1) and (2) respectively.

$$G_s(x,y,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{x^2+y^2}{2\sigma^2}) \quad (1)$$

$$G_g(x,y) = A \cdot exp(-(ax^2+2bxy+cy^2)) \quad (2)$$

where A is the normalization factor, the coefficients a, b, and c explicitly control the shape and orientation of kernel $G_g(x,y)$ by means of θ, σ_x, and σ_y.

$$a = \frac{cos^2\theta}{2{\sigma_x}^2} + \frac{sin^2\theta}{2{\sigma_y}^2} \quad (3)$$

$$b = -\frac{sin2\theta}{4{\sigma_x}^2} + \frac{sin2\theta}{4{\sigma_y}^2} \quad (4)$$

$$c = \frac{sin^2\theta}{2{\sigma_x}^2} + \frac{cos^2\theta}{2{\sigma_y}^2} \quad (5)$$

Then the gLoG filter can be presented as follows:

$$\nabla^2 G_g(x,y) = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2} \quad (6)$$

Figure 2 shows the gLoG kernels with different shapes (controlled by σ_x and σ_y) and orientations (controlled by θ). The first row in Figure 2 shows the 2-D circular LoG filters. Compared with the circular LoG kernels, we can see that the proposed gLoG filters can be readily applicable to estimate the orientations of local image textures.

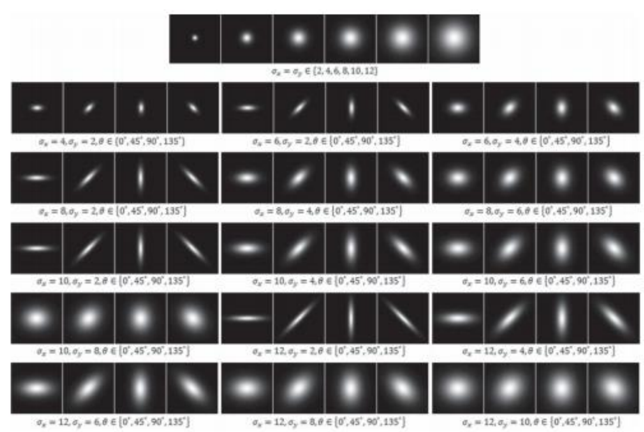


Figure 2. Generalized LoG filters [12].

B. Pixel-Wise Texture Orientation Estimation

In order to estimate texture orientations, we need to generate a set of gLoG filters. As mentioned above, the gLoG filters are determined by σ_x, σ_y and θ. According to the experimental results in [12], we set ${\sigma_x}^{max}$ =16, ${\sigma_y}^{min}$ =4, the number of orientations n_θ=12 ($n_\theta = \frac{180^\circ}{\theta}$). Because σ_x and σ_y are interchangeable, we set σ_y smaller than σ_x.

After generating a set of gLoG kernels by using different combinations of $\{\sigma_x, \sigma_y, \theta\}$, we divide the produced kernels into n_θ groups, where each groups only contains the kernels with the same orientation. Then, the test image is convolved with every kernel in each group. Each pixel's orientation is determined by the group that can produce the maximum convolution response. Figure 3 shows the estimated texture orientation map. In Figure 3(a), the image is overlaid with texture orientation bars at evenly sample locations.

(a) Input image (b) Texture orientation map

Figure 3. Visualization of estimated texture orientation map.

C. Voting Map Generation Using MASV Scheme

After estimating the texture orientation map at each pixel of the image, one can make these pixels vote to obtain the voting map. Then, vanishing point will be detected by using peak identification in the voting map.

Below is the voting scheme proposed in [12], which has achieved very good detection result.

$$Vote(p,v) = \begin{cases} exp\left(-d(p,v) * \frac{|\gamma|}{l}\right), if\ |\gamma| \le \delta \\ 0, \quad otherwize \end{cases} \tag{7}$$

where $d(p,v)$ is the Euclidean distance between pixel p and v, l is the normalization factor, which is set to the diagonal length of the input image. δ is set to the angular resolution $\frac{180^\circ}{n_\theta}$. In (7) pixel p whose vector is $\overline{V_p}$ can vote for any pixels above p as long as the angle between the direction *(pv)* and the vector $\overline{V_p}$ is below the threshold value δ, which means that each pixel in the image will vote for the sky region. It's really time consuming and will also introduce a lot of noise.

The complete-map based gLoG method uses exponential function for weighted voting. Using this voting scheme, the votes decrease rapidly as the distance increases, and pixels will mainly vote for the locations nearby themselves. Whereas vanish point is the converging point of the parallel lines in perspective images, and at least more than half of the pixels which belong to the road regions are far away from the vanishing point. It means that most of the pixels belonging to the road regions cannot generate effective votes to the ground truth vanishing point. Compared to exponential function, Gaussian function is more adaptive. As the distance increases, the weighted votes decrease slowly. Hence, we propose a modified adaptive voting scheme to further improve the voting accuracy, as shown in (8).

$$Vote(p,v) = \begin{cases} exp\left(-\frac{(d(p,v)*|\gamma|)^2}{2\sigma^2 * l}\right), if\ |\gamma| \le \delta\ and\ d(p,v) < R_v \\ 0, \quad otherwize \end{cases} \tag{8}$$

where R_v is the radius of the voting region, l is the normalization factor, and σ is experimentally set to 20.

D. Speed Up by Using Image Scaling

Although the accuracy of vanishing-point detection is very promising based on the pixel-wise texture orientation estimation and voting, whereas, it is really time-consuming during the voting stage. The complexity of the voting stage [14] is $O(w^2 * h(h+1)/2)$, assuming that the dimension of the input image is $w * h$. We can see that the complexity of the voting stage is mainly determined by the dimension of the input image. If the image is down sampled, its computation time will be significantly reduced. Hence, in this paper, a scaling method is used to reduce the computation time.

(a) No scaling (b) 1/4 scaling

Figure 4. Detected vanishing point with different scaling factors of the voting map.

The method is very simple, we first down sample the input image, and then compute the voting map of the down sampled image. The final voting map is up sampled by up sampling the voting map. Considering that the bilinear interpolation makes a good trade-off between computation time and image quality, we use the popular bilinear interpolation method for image scaling. The scale factor is experimentally set to 1/4. The results are shown in Figure 4. As we see, almost no difference can be found for different scales in the detection of vanishing point.

IV. Experimental Results

There are several methods reported in [12]. The one based on the complete texture orientation map (complete-map based gLoG) can achieve the best result among all. In our paper, fair comparison is performed with the complete-map based gLoG [12], which is considered as the state-of-the-art method.

 ISBN: 978-1-61208-393-3

The experiments contain the comparison of vanishing point detection accuracy and the computation time. The dataset in this experiment consists of 1000 images taken from local roads with various types of background, illuminations, and traffic conditions. For every image, the ground truth of vanishing point is manually labeled through the method suggested in [12].

In order to measure the accuracy of vanishing point detection methods, we use the normalized Euclidean distance as suggested in [16][17], where the Euclidean distance between the estimated vanishing point and the ground truth is normalized by the length of diagonal of the input images as follows:

$$NormDist = \frac{\|V_e(x,y) - V_g(x,y)\|}{DiagImage} \quad (8)$$

where $V_e(x, y)$ and $V_g(x, y)$ are the estimated vanishing point and the ground-truth of vanishing point respectively, DiagImage is the diagonal length of the given image, which is used for normalization. If NormDist value is close to 0, it means that the detected vanishing point is close to the ground truth; otherwise, it may correspond to incorrectly estimated vanishing point.

A. Vanishing Point Detecting Accuracy Comparsion

Figure 5 shows the comparison of vanishing point detection accuracy between our method and the complete-map based gLoG. The *x* axis is the *NormDist* distance between the detected vanishing point and the ground truth, and y axis is the detection accuracy.

We will consider the detected vanishing point is accurate when the *NormDist* distance between the detected vanishing point and the ground truth is smaller than a certain *NormDist* value, as suggested in references [5]][12][16][17]. It can be seen that our method outperforms by 4.64% on average under the same *NormDist* distance. Figure 6 shows some examples of detected vanishing points by our method and the reference method.

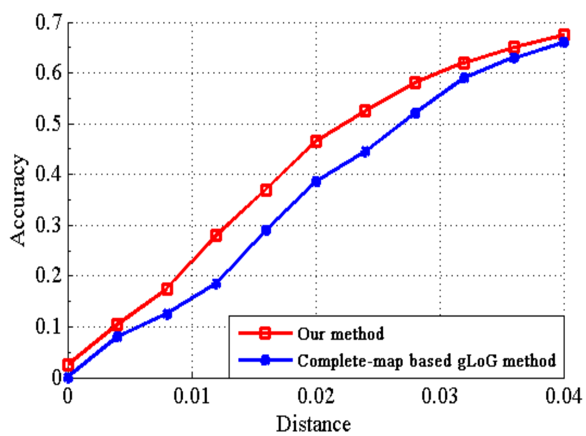


Figure 5. Results comparison of our method and the complete-map based gLoG method.

Figure 6. Sample detection results of our method (right) and the complete-map based gLoG method (left).

B. Computation Time Comparison

The experiments are conducted in Matlab with a 3.4 GHz Intel i7 Processor. Table 1 shows the computation time comparison of our method and the complete-map based gLoG method [12].

TABLE I. COMPARISON OF CPU TIME

Image Size	*Complete-map based gLoG*	*Our Method*	*CPU time reduced*
360*480	594.5 s	46.8 s	92.1%

We can see that our method is approximately 12 times faster than the complete-map based gLoG method.

V. CONSLUSION

In this paper, we have proposed a new framework for vanishing point detection. By using image scaling and MASV, our approach shows better results both in detection accuracy and CPU time than those of complete-map based gLoG. We believe that our method is useful for fast and robust vanishing point detection.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MOE) (No. NRF-2013R1A1A2004421).

REFERENCES

[1] J. C. McCall and M. M. Trivedi, “Video Based Lane Estimation and Tracking for Driver Assistance: Survey, System, and Evaluation,” IEEE Transactions on Intelligent Transportation Systems, vol. 7, no.1, Mar. 2006, pp. 20-37.

[2] Y. Wang, E. K. Teoh, and D. Shen, “Lane Detection and Tracking Using B-snake,” Image and Vision Computing, vol. 22, Apr. 2004, pp. 269-280.

[3] T. Suttorp and T. Bucher, “Robust Vanishing Point Estimation for Driver Assistance,” Proc. of the IEEE Intelligent Transportation Systems Conference, 2006, pp. 1550-1555.

[4] Y. He, H. Wang, and B. Zhang, “Color-based Road Detection in Urban Traffic Scenes,” IEEE Trans. on Intelligent Transportation Systems, vol. 5, no. 4, Dec. 2004, pp. 309-318.

[5] H. Kong, J. Audibert, and J. Ponce, “General Road Detection from A Single Image,” IEEE Trans. on Image Processing, vol. 19, no. 8, Aug. 2010, pp. 2211-2220.

[6] C. Rasmussen, “Grouping Dominant Orientations for Ill-structured Road Following,” Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, 2004, pp. I-470–I-477.

[7] Q. Wu, W. Zhang, and B. V. Kumar, “Example-based Clear Path Detection Assisted by Vanishing Point Estimation,” Proc. of the IEEE International Conference on Robotics and Automation, 2011, pp. 1615–1620.

[8] N. Hautiere, J. Tarel, and D. Aubert, “Mitigation of Visibility Loss for Advanced Camera-based Driver Assistance,” IEEE Transaction Intelligent Transportation Systems, vol. 11, no. 2, Jun. 2011, pp. 474–484.

[9] J.M. Alvarez, T. Gevers, and A.M. Lopez, “3D Scene Priors for Road Detection,” Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 57-64.

[10] Q. Wu, T. Chen, and B. Kumar, “Prior-based Vanishing Point Estimation Through Global Perspective Structure Matching,” Proc. of IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), 2010, pp. 2110-2113.

[11] P. Moghadam, J. A. Starzyk, and W. S. Wijesoma, “Fast Vanishing Point Detection in Unstructured Environments,” IEEE Transactions on Image Processing, vol.21, no.1, Jan. 2012, pp. 425-430.

[12] H. Kong, S. Sarma, and F. Tang, “Generalizing Laplacian of Gaussian Filters for Vanishing-Point Detection,” IEEE Transactions on Intelligent Transportation Systems, vol.14, no.1, Mar. 2013, pp. 408-418.

[13] H. Kong, J.Y. Audibert, and J. Ponce, “Vanishing Point Detection for Road Detection,” Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 96-103.

[14] H. Guo, “A Simple Algorithm for Fitting a Gaussian Function [DSP Tips and Tricks],” IEEE Transactions on Signal Processing Magazine, vol.28, no.5, Sept. 2011, pp. 134-137.

[15] M. Nieto and L. Salgado, “Real-Time Vanishing Point Estimation in Road Sequences Using Adaptive Steerable Filter Banks,” Proc. of International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS), 2007, pp. 840–848.

[16] P. Moghadam, J.A. Starzyk, and W.S Wijesoma, "Fast Vanishing-Point Detection in Unstructured Environments," IEEE Transactions on Image Processing, vol.21, no.1, Jan. 2012, pp. 425-430.

[17] W. Yang, X. Luo, B. Fang, and Y. Y.Tang, "Fast and accurate vanishing point detection in complex scenes," IEEE International Conference on Intelligent Transportation Systems (ITSC), Oct. 2014, pp. 93-98.

 ISBN: 978-1-61208-393-3