A NEW EFFICIENT VANISHING POINT DETECTION FROM A SINGLE IMAGE

Xiqun Lu

College of Computer Science, Zhejiang University, China

ABSTRACT

Detecting the vanishing point from a single road image is a challenging problem since there is only very limited information in the input image which can help the computer to deduce the genuine location of the vanishing point. Moreover, sometimes the cluttered ambient environment in a real road image will hinder rather than help the detection. Learning the advantages and the limitations of current edge-based and texture-based approaches motivates us to propose a new vanishing point detection method, which integrates the efficiency of edge-based methods and the orientation coherence concept of texture-based methods that can be of a great help to improve the accuracy of detection. The proposed method has been tested on over 200 various road images downloaded from the internet. These images exhibit large variations in color, texture, illumination and ambient environment. The experimental results demonstrate that this new method is both efficient and effective at detecting vanishing point compared with some state-of-the-art edge-based and texture-based methods.

Index Terms— vanishing point detection, orientation, coherence, edge, Gabor filters

1. INTRODUCTION

Over the past few decades, there are numerous research works about developing autonomous vehicle navigation systems in either structured [1, 7], urban environments [2] or unstructured roads [3-6]. In these applications, estimating the vanishing point and detecting road boundaries is like an egg-and-chicken problem. If the vanishing point can be located correctly, then it is highly probable for the computer to detect road boundaries properly. On the other hand, the perspective projection of parallel road lanes or vehicle tracks can be of a great help to estimate the location genuine of the vanishing point.

The majority of vision-based vanishing point detection methods in the literatures can be grouped into three main categories: edge-based methods [1, 7], texture-based methods [3, 5, 6], and prior-based methods [8, 9]. Edge-based methods, e.g. in [1], edge pixels extraction by the Canny detector, and then straight lines detected by the Hough transform, finally the intersections of pairs of lines vote for the vanishing point on another Hough space. Most of these approaches can be applied to real-time applications due to their computation efficiency and appropriate for structured roads with well painted parallel road lanes, while they are sensitive to spurious edges in the scene, and may not perform well in unstructured roads without strong edges or contrasting local characteristics. Textured-based methods [3, 5, 6] search for local oriented textures and then make them vote for the locations of the road's vanishing point. Either the local soft voting scheme proposed in [5] or the global voting schemes proposed in [3, 6] are time-consuming and cannot meet the requirements of real-time applications, and at the same time these approaches are sensitive to the noise, if the scene exists some obstacles with strong edges than the tracks left by previous passed vehicles, then these strong boundaries will lead the voter to an incorrect vanishing point (please see the image in the 4th row of Fig.3).

In order to overcome the limitations of these low-level feature-based detection methods, recently some prior-based techniques have been proposed. In [8], they suggested to integrate contextual 3D information with low-level features to improve the detection performance. Such weak contextual cues include a 3D scene layout, 3D road stages and temporal road cues, and so on. In [9], they studied 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 used the pre-labeled vanishing points of the best candidates 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. All of these requirements will make it difficult to apply these prior-based methods in real-time and practical situations.

In this paper, we propose a new efficient vanishing point detection method which explores some intrinsic geometric structures and color texture properties in a road image. This method is tested on over 200 various road images downloaded from the Internet. In this very challenging image dataset, we have structured and unstructured images, front-viewed and slant-viewed road images, and road images taken under different illumination and weather conditions. The proposed method provides the highest vanishing point detection accuracy when compared to some state-of-the-art edge-based and textured-based methods [1, 5, 6].

This paper is arranged as follows: In Section 2 we introduce the new efficient vanishing point detection method which integrates both the advantages of edge-based methods and texture-based methods, and in section 3 we evaluate the performance of the proposed method quantitatively and qualitatively. Finally, conclusions are drawn in Section 4.

2. THE FRAMEWORK OF VANISHING POINT DETECTION

In this section we present a new framework for the vanishing point detection that utilizes the intrinsic geometric structures and color texture properties in a road image. Although it is based on line

segments, it is very different from common edge-based methods [1, 7] which determine the vanishing point by searching a point that is close to most line segments. Since in a real road image, the cluttered ambient environment and background will introduce many spurious edges which will lead to an incorrect vanishing point detection just based on a simple voting scheme, therefore we consider using an orientation coherence measure to select the right line segments, and checking the color texture difference between the two parts separated by the mid-line of each intersected line pair to determine which intersection point should be the vanishing point. Next we will introduce each step in detail.

2.1. Selection of Right Line Segments

For estimating the vanishing point, only those line segments corresponding to the lane borders and the vanishing line in the world coordinate should be used. However the cluttered ambient environment and background will introduce many spurious edges which will make the edge-based methods prone to an incorrect detection. And the sensitive derivative operators can make thing even worse. Therefore the first problem we should face is the selection of the right edge lists that will be used in the subsequent line fitting stage (Here we use the Canny algorithm [10] to obtain the edge lists). We use the following criteria to ensure that only appropriate edge lists are selected:

♦ The edge lists must be longer than a minimal length. In a general road image, the lanes of the road tend to be longer than other edges. However, to set a minimal length for various road images is a difficult problem. In our implementation we set the initial minimal length to be the half of the height of input image. Then the minimal length can be reduced gradually to a smaller value until under which enough edge lists can be selected or the minimal length reaches a low bound value (In our implementation we set this low bound as 10 pixels). The number of enough edge lists in our implementation is set as 10.

♦ The orientations of the points on these selected edge lists must be as coherent as possible. Hence we apply a simple sequential orientation coherent measure as follows:

$$C_i = \sum_{\mathbf{p}_j, \mathbf{p}_{j+1} \in edge\_list(i)} \left\| \hat{\theta}(\mathbf{p}_j) - \hat{\theta}(\mathbf{p}_{j+1}) \right\| \tag{1}$$

Where $\mathbf{p}_j$ and $\mathbf{p}_{j+1}$ are the two consecutive points on the i^{th} selected edge list, and $\hat{\theta}(\mathbf{p}_j)$ and $\hat{\theta}(\mathbf{p}_{j+1})$ are the estimated orientations of the two points (computed by the 5-tap derivative filters proposed in [11]). Then we select 5 the most coherent edge lists with smallest C values of (1) for the next line fitting stage.

2.2. Straight Lines Fitting

Based on the points of each of these remained coherent edge lists, we try to find the parameters of each straight line $\mathbf{l}_i = (a_i \quad b_i \quad c_i)^T$ by using the Least-Squares Minimization method:

$$\min \left\| \mathbf{A}\mathbf{l}_i \right\| \tag{2}$$

Where $\mathbf{A}$ is a $n_i \times 3$ matrix, and n_i is the total number of points on the i^{th} edge list, so the part inside the norm symbol of (2) is as following:

$$\begin{bmatrix} x_1^i & y_1^i & 1 \\ \vdots & \vdots & \vdots \\ x_{n_i}^i & y_{n_i}^i & 1 \end{bmatrix} \mathbf{l}_i \tag{3}$$

Then we applied the SVD to $\mathbf{A}$, and the unit singular vector corresponding to the smallest singular value is the solution of $\mathbf{l}_i$. And the intersection point of two lines can be computed as $\mathbf{s} = \mathbf{l}_i \times \mathbf{l}_k$.

2.3. Determination of the Vanishing Point

Since there are usually multiple intersection points, it is very difficult to decide which one is the genuine vanishing point because most of those intersection points have almost the same number of lines through them, and to determine the vanishing point just by searching a point close to most line segments is prone to an incorrect position.

So we check an orientation coherent ratio along each line pair downward from their intersection point respectively, and this idea is borrowed from Kong et al. [5]:

$$C(\mathbf{l}_i^s) = \frac{\#(\left| \hat{\theta}(\mathbf{p}_j) - \theta(\mathbf{l}_i) \right| < T, \mathbf{p}_j \in \mathbf{l}_i^s)}{length(\mathbf{l}_i^s)} \tag{4}$$

Where $\mathbf{l}_i^s$ indicates the line $\mathbf{l}_i$ started with the considered intersection point downward to the bottom of the input image, in other words, $\mathbf{l}_i^s$ is a part of $\mathbf{l}_i$ from the intersection point to the bottom of the input image, and $\theta(\mathbf{l}_i)$ represents the orientation of the line $\mathbf{l}_i$; and $\mathbf{p}_j$ denotes a point on the line $\mathbf{l}_i$ begun with the considered intersection point downward, and $\hat{\theta}(\mathbf{p}_j)$ are the estimated orientation of the point $\mathbf{p}_j$ as in (1); # is the operator to count how many points on $\mathbf{l}_i^s$ have the angular differences between the local edge directions and the direction of the line $\mathbf{l}_i$ that are less than a given threshold T (in our implementation T is set as 3°). So the higher value $C(\mathbf{l}_i)$ is the more coherent of the line $\mathbf{l}_i$ is.

In a general road image, the road part constructed by the two lanes should have uniform color texture no matter whether it is a structured road or an unstructured road. So we not only compute the sum orientation coherent ratios of each intersected line pair, but compute the mean color difference between the parts separated by the mid-line of the intersected line pair as well.

As we know the angles of all these fitted lines, so the angle of mid-line of each intersected line pair can be computed as easily as $(\phi_1 + \phi_2)/2$. Also it is easy to decide which points are located inside the area enclosed by two intersected lines as shown in Fig.3: the angle of the line connected the intersection point and the considered point is larger than ϕ_1 and less than ϕ_2. Now we compute the color texture difference between the two parts enclosed by two intersected lines and their mid-line:

$$CT\_diff(\mathbf{l}_i^s, \mathbf{l}_j^s) = \left| \frac{1}{\#(\Omega_1)} \sum_{\mathbf{p}_1 \in \Omega_1} c(\mathbf{p}_1) - \frac{1}{\#(\Omega_2)} \sum_{\mathbf{p}_2 \in \Omega_2} c(\mathbf{p}_2) \right| \tag{5}$$

Where line $\mathbf{l}_i$ and line $\mathbf{l}_j$ are intersected at $\mathbf{s} = \mathbf{l}_i \times \mathbf{l}_j$; Ω_1 and Ω_2 are the two parts separated by the mid-line of the area enclosed by line $\mathbf{l}_i$ and line $\mathbf{l}_j$; # is the operator to count how many points located in Ω_1 and in Ω_2; $c(\mathbf{p})$ is the color at point $\mathbf{p}$ located either in Ω_1 or in Ω_2.

Now we define the cost function under which we select the intersected line pair to determine the final vanishing point:

$$\cos t(\mathbf{l}_i^s, \mathbf{l}_j^s) = CT\_diff(\mathbf{l}_i^s, \mathbf{l}_j^s) - C(\mathbf{l}_i^s) - C(\mathbf{l}_j^s) \tag{6}$$

The intersection point of two intersected lines which has the minimum cost value of (6) is considered as the vanishing point.

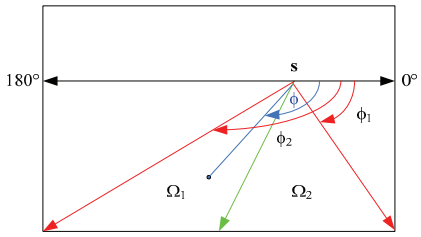

Fig.1. Example of computing the color texture difference between the two parts enclosed by two intersected lines and their mid-line (The two red arrowed lines represent the two intersected lines at **s** , and the green line is the mid-line of the two intersected lines).

3. RESULTS AND ANALYSIS

In this section, we evaluate the performance of our proposed method versus classical Hough-based method [1], Kong et al. method [5] and Moghadam et al. method [6] quantitatively and qualitatively. Over 300 various road images are downloaded by Google Image, but currently we remove some images from the test dataset for future research because the roads in those images are too curving or have several different directions simultaneously. Vanishing point estimation algorithms are tested on over 200 very different road images. Some of the test images are shown in Fig.3 (We cannot present more results here due to the limitation of space.). We can see that some of these images have well structured roads, but some of them have only unstructured roads, like vehicle tracks in desert or snow, and some of them are water roads. These images exhibit large variations in color, texture, illumination and ambient environment.

Since these downloaded images have different sizes, all images are normalized to the same size with a height of 180 and a width of 240 by using the bicubic image interpolation method [12]. To assess the algorithm's performance vs. human perception of the vanishing point location, we invite 5 students in our college to manually mark the vanishing point location in each image of the test image dataset after a brief description of the road vanishing point concept. Since students' marked vanishing points in each image are very close, we defined the center of these marked locations as the ground truth vanishing point location.

To measure the accuracy of vanishing point estimation methods, we use the normalized Euclidean distance as suggested in [6], where the Euclidean distance between the estimated vanishing point and the ground truth is normalized by the length of the diagonal of the input image as follows:

$$NormDist = \frac{\left\| v_E(x_e, y_e) - v_T(x_t, y_t) \right\|}{Diag \quad image} \tag{7}$$

Where $v_E(x_e, y_e)$ is the estimated vanishing point location and $v_T(x_t, y_t)$ is the location of ground truth vanishing point; "Diag image" is the diagonal length of the input image. Table I shows the accuracy results in terms of the average normalized Euclidean distance error for the test image dataset and those of the four vanishing point detection methods tested on different kinds of roads (structured roads, unstructured roads, snow covered roads, and night or dark ambient roads) respectively. We didn't tune any parameters when applied these algorithms to different types of roads.

For better comparison, we also evaluate the results of the four vanishing point estimation algorithms in an accumulated histogram. Since the input image is with the size of 180×240, the normalized Euclidean distance of 0.1 of (7) means that the location of estimated vanishing point is about 30 pixels away from the genuine one, and this is a very large error. So we only consider the normalized Euclidean distance is less than 0.1 in an accumulated histogram as shown in Fig.2. The accumulated histogram of a perfect vanishing point detection method should be a step function, that is, there is no error between the locations of estimated vanishing points and those of the ground truth ones. Therefore the higher lines in the accumulated histogram represent better results. From Fig.2, Our proposed method's accumulated histogram line is not only higher above the other three vanishing point detection methods' lines, but includes more test images whose estimation errors are less than 0.1 as well.

Most of vanishing point detection methods are only for front-viewed road images, but one of the distinctive advantages of our proposed method is that it can detect the vanishing points for slant-viewed images (please see the images in the 2nd and 3rd row of Fig.3) because it depends on the orientation coherence of each intersecting line pair and the color difference between the two parts separated by the middle line of each intersecting line pair (In the implementation we only consider line pairs that the angles between the two lines are larger than 10°, and this is reasonable for general road images).

Although Kong et al. method [5] can do a pretty good job for unstructured road images, when the input image has some obstacles with much strong boundaries than the vehicle tracks, then the estimated location of vanishing point will be distracted from the genuine one, for example, the rear window of the car in the image in the 4th row of Fig.3. Because in their local soft voting scheme, only those pixels whose Gabor responses are larger than a threshold are qualified for voting.

In order to meet the requirements of real-time applications, Moghadam et al. in [6] suggested that first downsampled all the images to 80×60 pixels by Gaussian pyramid and then pass to the vanishing point detection method to find the vanishing point location, and then projected it back to the original image size. However the processes of down-sample and projection to the original image size will induce extra estimation errors for the vanishing point detection, hence we apply their voting scheme to the original sized images.

Among the four vanishing point detection algorithms, the Hough-based method is the fastest one, with about 0.2 seconds for one image, and our proposed method ranks the second fast, about 0.5 seconds for one image, but for the texture-based methods, e.g. Kong et al. [5] and Moghadam et al. [6], it needs about 13 to 15 seconds for one image (which are image dependent, most of the computation time is spent on the voting stage). We implemented them all in Matlab.

4. CONCLUSIONS

Detecting the vanishing point from a single road image is a challenging problem as there are only very limited features that can characterize the road. Currently there are two main techniques for vanishing point estimation: edge-based methods [1, 7] and

texture-based methods [3, 5, 6]. Edge-based methods are fit for structured road images, while texture-based methods are more suitable for unstructured roads since they exploit some consistent features of ruts and tracks based on texture analysis. In our earlier stage of research works, we find that the voting scheme plays a more important role in extracting texture orientation information than the selection of filter banks does.

Based on the analysis of both the advantages and the limitations of edge-based methods and texture-based methods, we propose a new vanishing point estimation method in this paper which integrates the efficiency of edge-based methods and the orientation coherence concept of texture-based methods. In order to test the performance of this method, a series of quantitative and qualitative analyses were conducted on a real road image dataset which contains over 200 road images downloaded from the internet. These images exhibit large variations in color, texture, illumination and ambient environment. The performance in terms of accuracy of the proposed method outperforms the state-of-the-art edge-based and texture-based vanishing point detection methods.

Table I. The accuracies of different vanishing point estimation algorithms for different road types

Methods / Road Types	Our Proposed method	Classical Hough-based [1]	Kong et al. [5]	Moghadam et al. [6]
Total Ave.	**0.13487**	0.24479	0.16668	0.14024
Structured	**0.02179**	0.05164	0.02582	0.02946
Unstructured	0.15476	0.26273	0.16429	**0.12050**
Snow	**0.12706**	0.21134	0.19257	0.14530
Night	**0.11876**	0.26469	0.23198	0.12703

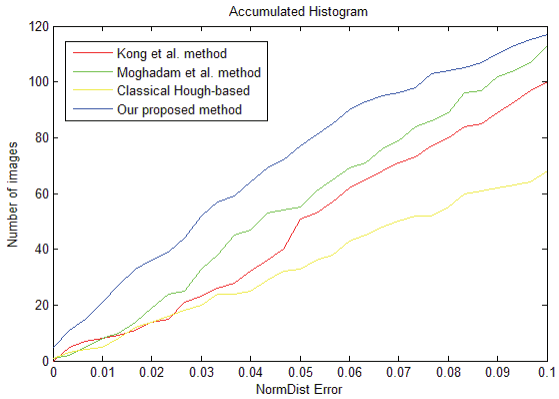

Fig.2. Comparison of vanishing point detection methods in an accumulated histogram

5. REFERENCES

[1] Y. Wang, E. K. Teoh, and D. Shen, "Lane detection and tracking using B-snake," Image and Vision Computing 22: 269-280, 2004.

[2] Y. He, H. Wang and B. Zhang, "Color-based road detection in urban traffic scenes," IEEE Trans. On Intelligent Transportation Systems, 5(4): 309-318, 2004.

[3] C. Rasmussen, "Grouping dominant orientations for ill-structured road following," in Proc. Of IEEE CVPR, 2004.

[4] Y. Alon, A. Ferencz, A. Shashua, "Off-road path following using region classification and geometric projection constraints," in Proc. Of IEEE CVPR, 2006.

[5] H. Kong, J.-Y. Audibert, and J. Ponce, "General road detection from a single image," IEEE Trans. On Image Processing, 19(8):2211-2220, 2010.

[6] P. Moghadam, J. A. Starzyk, and W. S. Wijesoma, "Fast vanishing point detection in unstructured environments," IEEE Trans. On Image Processing, 2011 (to appear).

[7] T. Suttorp and T. Bucher, "Robust vanishing point estimation for driver assistance," in Proc. of the IEEE ITSC, 2006.

[8] J. M. Alvarez, T. Gevers, and A. M. Lopez, "3D scene priors for road detection," in Proc. Of IEEE CVPR, 2010.

[9] Q. Wu, W. Zhang, T. Chen, and B. V. K. Vijaya Kumar, "Prior-based vanishing point estimation through global perspective structure matching," in Proc. Of IEEE ICASSP, 2010.

[10] J. Canny, "A computational approach to edge detection,"IEEE Trans. On Pattern Analysis and Machine Intelligence, 8(6): 679-698, 1986.

[11] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," IEEE Trans. On Pattern Analysis and Machine Intelligence, 13(9): 891-906, 1991.

[12] R. G. Keys, "Cubic convolution interpolation for digital image processing," IEEE Trans. On Acoustics Speech and Signal Processing, 29(6): 1153-1160, 1981.

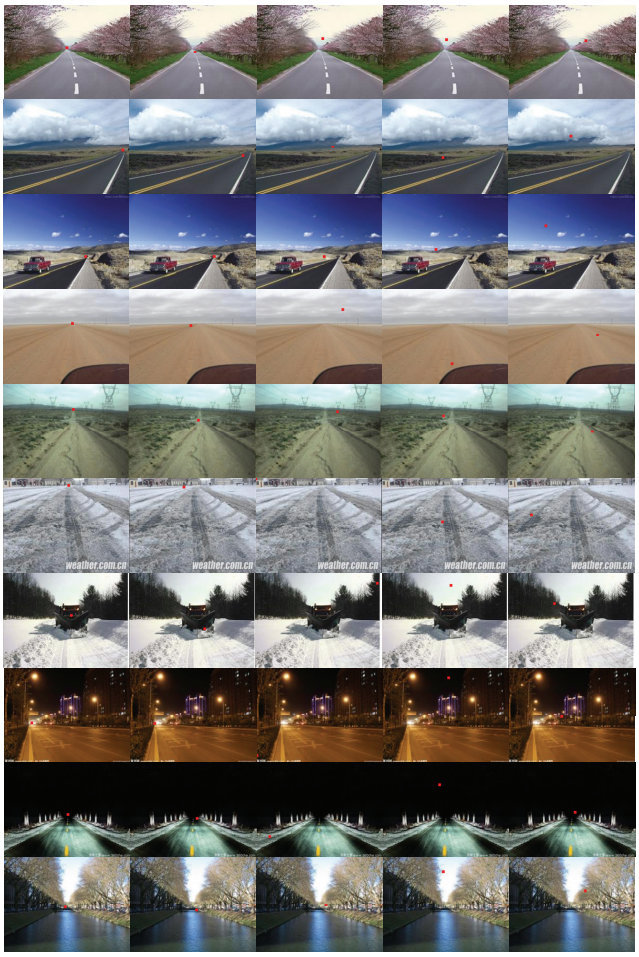

Fig.3. Examples of vanishing point detection for different kinds of roads. The first column is the ground truth locations of vanishing points. The second column is the estimated vanishing point locations of our proposed method, and the third column is the estimated locations of the Hough transform-based method [1]. The forth and the fifth columns are the vanishing point locations estimated by Kong et al. [5] and by Moghadam et al. [6] respectively.