ENHANCED PIXEL-WISE VOTING FOR IMAGE VANISHING POINT DETECTION IN ROAD SCENES

L. Nguyen, S. L. Phung, and A. Bouzerdoum

School of Electrical, Computer and Telecommunications Engineering,
University of Wollongong, Australia

ABSTRACT

Vanishing point estimation is a crucial task in vision-based road detection. This paper presents a new texture-based voting scheme, which enhances both accuracy and speed of vanishing point estimation. In the proposed method, color tensors analysis is adopted to calculate local orientations and color edges. The search space is reduced by optimizing the set of vanishing point candidates and voters. A new strategy based on Bayesian classifier is proposed to select a suitable voting function. The proposed method is evaluated on a benchmark dataset of 4000 images of pedestrian lanes with annotated vanishing points. The experimental results show that it offers an improved accuracy and significantly faster processing time compared with other state-of-the-art methods.

***Index Terms*—** vanishing point, pixel-wise voting, Bayesian classifier.

1. INTRODUCTION

A vanishing point (VP) is considered as a useful image feature in many applications, such as road detection [1], camera calibration [2], and 3D reconstruction [3]. In road detection [1,4–6], vanishing point estimation (VPE) is employed to provide contextual cues about the scene. However, the task of estimating the vanishing point from a single road image is challenging, due to the unconstrained nature of the road scenes and the real-time requirement of practical applications. A VPE method in road scenes has to offer fast running time as well as robustness to various imaging factors, including road types, lighting conditions, and weathers.

Generally, state-of-the-art VPE methods can be grouped into two categories: edge-based and texture-based. Edge-based methods [4, 7, 8] first locate dominant straight lines, which often correspond to road borders and lane markings. Algorithms such as the Hough transform [9] or Line Segments Detection [10] have been applied to extract lines from edge features of the image. Next, the intersections of these lines are considered as the VP candidates, from which the VP is selected via a voting scheme. Because usually the number of lines/intersections is small, the voting scheme in edge-based methods is simple and involves a small search space [4, 7, 8]. Overall, the edge-based methods offer fast execution time and they have reasonable accuracy for structured scenes. However, these methods are not so effective for unstructured scenes.

In comparison, texture-based methods first try to find the dominant texture orientation at each image pixel [1, 5, 11–16]. In unstructured scenes, orientation features are more suitable than edge features when voting for the vanishing point [5]. For orientation estimation, traditional methods use a large number of Gabor filters, e.g. 144 filters (2 phases, 72 orientations) in [5], or 180 filters (5 scales, 36 orientations) in [11]. Despite their proven accuracy for texture analysis [17], the Gabor filters have significant computation complexity. There are some recent efforts to address this drawback [11, 12, 14–16]. In [12], a fast algorithm involving interpolation from just four Gabor filters is proposed for texture estimation. This approach is enhanced in [16], which employs response thresholding and trigonometric adjustment. In another approach, the Gabor filters bank in [11] is approximated with a composition of binary Haar-like filters to reduce the complexity. In some studies, the Gabor filters are replaced with other solutions, e.g. gLoG [14] and color tensors [15].

To select the vanishing point from a set of candidates, texture-based methods usually utilize a voting scheme. For example, in [5], the candidates for VP include all the pixels in the 90-percent upper image region. The voting scores are accumulated from the pixels (voters) located below each candidate. This voting scheme is computationally intensive because of the large search space. It also favors the candidates located in the upper region of the image. There are a number of efforts in improving the VP voting scheme. In the Local Adaptive Soft Voting (LASV) algorithm proposed in [6], only pixels in the half-disk region located below the candidate are considered as voters. To reduce the bias, the voting score in this algorithm takes into account the spatial distance between the VP candidate and voter. In more recent studies [1, 15], the VP voters include only color edge pixels located below the candidate. However, in these texture-based methods, the search space still involves a large number of VP candidates. This hinders the use of texture-based approach despite its accuracy for various types of road scenes.

This paper aims to enhance the voting scheme of the texture-based approach in both speed and accuracy. The main contributions are twofold. First, we present a new strategy for defining the set of VP candidates and VP voters, thereby reducing the search space significantly. Second, we introduce a new approach for calculating the VP voting score, which combines different weighting functions and the Bayesian decision rule. The remaining of the paper is structured as follows. The proposed VPE method is presented in Section 2. Experiments and results are given in Section 3, and the conclusion is provided in Section 4.

2. PROPOSED VPE METHOD

The VPE method proposed in this paper involves three main stages: i) extracting local texture orientation and color edge map from the input image; ii) identifying vanishing point candidates and voters; and iii) determining the VP using a pixel-wise voting scheme with weight functions. These stages are illustrated in Fig. 1.

2.1. Extracting local texture orientation and color edge map

In this stage, color tensor analysis [18] is applied. For the input RGB image I, with the color channels I_k ($k = 1, 2, 3$), the color tensor

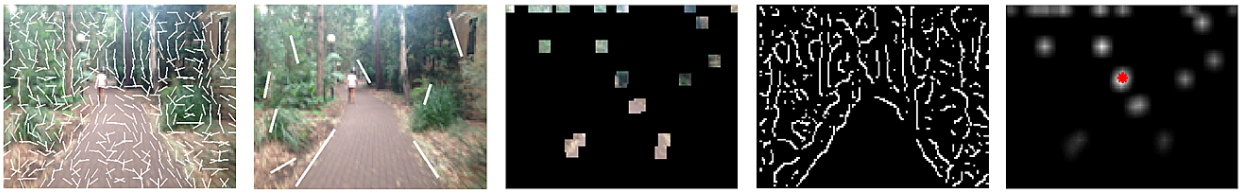

Fig. 1. Illustration of the proposed vanishing point estimation: (a) a color input image with sample pixel orientation imposed; (b) line segments detected with Hough transform; (c) VP candidates; (d) VP voters; (e) voting score map and the detected vanishing point (red asterisk). See electronic color figure.

structure of the image is given by

$$T = \begin{pmatrix} T_{xx} & T_{xy} \\ T_{xy} & T_{yy} \end{pmatrix}, \tag{1}$$

where

$$\begin{aligned} T_{xx} &= \mathbf{g} * \left[\sum D_{k,x} \circ D_{k,x} \right], \\ T_{yy} &= \mathbf{g} * \left[\sum D_{k,y} \circ D_{k,y} \right], \\ T_{xy} &= \mathbf{g} * \left[\sum D_{k,x} \circ D_{k,y} \right]. \end{aligned} \tag{2}$$

In (2), $D_{k,x}$ and $D_{k,y}$ are the derivatives of I_k along the horizontal and vertical directions respectively, $*$ denotes the 2-D convolution and $\circ$ denotes the Hadamard product, $\mathbf{g}$ is a 2-D Gaussian filter. Next, we calculate the eigenvectors and eigenvalues of the color tensors structure T similarly as in [18, 19]. The texture orientation map O is obtained from the eigenvectors as

$$O = \frac{1}{2} \arctan \left(\frac{2T_{xy}}{T_{xx} - T_{yy}} \right). \tag{3}$$

The gradient energies along the dominant orientations are given by

$$\Lambda = \frac{1}{2} \left[T_{xx} + T_{yy} + \sqrt{(T_{xx} - T_{yy})^2 + 4T_{xy}^2} \right]. \tag{4}$$

The color edge map (binary) B and the edge strength map (non-binary) E are obtained by applying non-maximum suppression and hysteresis thresholding on Λ. We will use O, B and E in the next stage.

2.2. Defining VP candidates and voters

A new strategy is proposed to optimize the number of VP candidates and VP voters. To define VP voters, we observe that the VP is located near of the intersections of the dominant straight lines in the image. Hence, the Hough transform [9] is first applied on the edge strength map E to detect the straight lines in the input image. The edge strength map E (instead of the binary map B) is employed to take into account weak edges, which often present in unstructured scenes. The intersections of the detected lines are calculated in the Hough space and projected into the image plane. Next, centered on each intersection, a square patch of size $r = \tau \times L$ is constructed, where L is the image diagonal and τ is the search threshold. The union of all the square patches is selected as the VP candidate pool. The square shape is chosen for better computational efficiency. The search space for VP candidates is regulated by the trainable threshold τ, and is now significantly reduced.

To define VP voters, we initially consider the edge pixels (represented by the binary edge map B) located below each VP candidate. Based on the observation that the road boundaries consist of pixels with similar orientations, a histogram of local orientations for the all the pixels is calculated. The edge pixels with a small count in this orientation histogram are removed from the VP voter pool. Furthermore, the pixels having vertical orientations are not considered as voters, because they mostly represent the non-lane objects, such as trees, lamp posts, and walls.

2.3. Pixel-wise voting with weighting method

Consider a VP candidate $\mathbf{c}$ and a VP voter $\mathbf{v}$. Let $f(\mathbf{v}, \mathbf{c})$ be the voting score contributed by voter $\mathbf{v}$ to candidate $\mathbf{c}$. The accumulative voting score $S(\mathbf{c})$ is defined as

$$S(\mathbf{c}) = \sum_{\mathbf{v}} f(\mathbf{v}, \mathbf{c}). \tag{5}$$

The candidate with the highest score is considered as the image vanishing point. The accuracy of the voting scheme depends on the voting score $f(\mathbf{v}, \mathbf{c})$. Generally, this voting score should be high if the local orientation at voter $\mathbf{v}$ has a similar direction as the line connecting $(\mathbf{v}, \mathbf{c})$, and if voter $\mathbf{v}$ is spatially close to candidate $\mathbf{c}$.

Let $O(\mathbf{v})$ be the local orientation at $\mathbf{v}$, and $\phi(\mathbf{v}, \mathbf{c})$ be the angle of the line connecting $(\mathbf{v}, \mathbf{c})$. The angle difference $\Delta(\mathbf{v}, \mathbf{c})$ is calculated as

$$\Delta(\mathbf{v}, \mathbf{c}) = |O(\mathbf{v}) - \phi(\mathbf{v}, \mathbf{c})|. \tag{6}$$

If the angle difference $\Delta(\mathbf{v}, \mathbf{c})$ exceeds a threshold δ, the contribution of the voter is nullified. Let $d(\mathbf{v}, \mathbf{c})$ be the Euclidean distance between $\mathbf{v}$ and $\mathbf{c}$. To improve consistency, we use the normalized distance $\bar{d} = d(\mathbf{v}, \mathbf{c})/L$, and the normalized angle difference $\bar{\Delta} = \Delta(\mathbf{v}, \mathbf{c})/\delta$.

The voting function is formed by combining two weight functions:

$$f(\mathbf{v}, \mathbf{c}) = \begin{cases} w_1(\bar{d}) \times w_2(\bar{\Delta}) & \text{if } \bar{\Delta} \leq 1, \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where $w_1(\bar{d})$ depends on the normalized distance, and $w_2(\bar{\Delta})$ depends on the normalized angle difference.

As mentioned above, $w_1(\bar{d})$ and $w_2(\bar{\Delta})$ should be decreasing

functions. We propose the following weighting functions[1]:

$$w_1(\bar{d}) = \exp(-\bar{d}),$$
$$w_2(\bar{\Delta}) = \frac{1}{1+\bar{\Delta}}. \tag{8}$$

Hence, the first choice for the voting function is given as

$$f_1(\mathbf{v},\mathbf{c}) = \begin{cases} \frac{\exp(-\bar{d})}{1+\bar{\Delta}} & \text{if } \bar{\Delta} \leq 1, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

The second choice for the voting function is given as

$$f_2(\mathbf{v},\mathbf{c}) = \frac{1}{1+\bar{\Delta}}. \tag{10}$$

We find that for some images $f_1(\mathbf{v},\mathbf{c})$ is a better voting function than $f_2(\mathbf{v},\mathbf{c})$, and for other images $f_2(\mathbf{v},\mathbf{c})$ is a better voting function than $f_1(\mathbf{v},\mathbf{c})$. To adaptively select $f_1(\mathbf{v},\mathbf{c})$ or $f_2(\mathbf{v},\mathbf{c})$, we propose to adopt the Bayesian decision rule for minimum cost [20].

For each input image, two features are computed. Let s be the longest line segment detected in the previous stage. The first feature is the the length of s, normalized against the image diagonal L:

$$l = |s|/L. \tag{11}$$

The second feature is the angle between s and the vertical axis, normalized by the angle threshold δ:

$$\gamma = \angle s/\delta. \tag{12}$$

Using a training set of input images and labeled vanishing points, we estimate the class-conditional probability density functions $p_1(l,\gamma)$ and $p_2(l,\gamma)$ that $f_1(\mathbf{v},\mathbf{c})$ or $f_2(\mathbf{v},\mathbf{c})$ performs better than the other, respectively. For a new test image with features l and γ, the voting function $f_1(\mathbf{v},\mathbf{c})$ is used if

$$\frac{p_1(l,\gamma)}{p_2(l,\gamma)} \geq \alpha, \tag{13}$$

where α is a threshold. The class-conditional pdfs can be estimated in many ways, e.g. see [20]. In this paper, we use a 2-D histogram approach with a total of N^2 bins.

3. EXPERIMENTS AND ANALYSIS

In this section, the image dataset and experiment methods are described (Subsection 3.1). Then, the effects of different parameters of the proposed method are analyzed (Subsection 3.2). Finally, the proposed VPE method is compared with other state-of-the-art methods (Subsection 3.3).

3.1. Image dataset and experimental methods

We utilized the Pedestrian Lane Detection and Vanishing Point Estimation (PLVP) dataset [1], which currently consists of 4000 images of unmarked pedestrian lanes, taken under various imaging conditions. The ground-truth vanishing point for each image has been marked manually by volunteers. The first 500 images in the dataset were dedicated for testing, and the remaining 3500 images were utilized for training.

The performance of the VPE methods on the dataset was evaluated based on three metrics. The first metric is the mean value e of the normalized Euclidean distances between the detected vanishing points and the ground-truth vanishing points. The second metric is the standard deviation std of these normalized distances. The third metric is the average time t to process an image. For comparison purpose, all the images were resized to the same height of 100 pixels. All experiments were conducted in MATLAB on a PC with a 3.4 GHz CPU and 8 GB of RAM.

[1] A justification for these weighting functions is given in Section 3.2.

3.2. Analysis of algorithm parameters

Several experiments were conducted to select a suitable voting function. First, we evaluated different versions of the weight functions $w_1(\bar{d})$ and $w_2(\bar{\Delta})$ in (7) on 200 training images. The results, shown in Table 1, indicate that $w_1(\bar{d}) = \exp(-\bar{d})$ and $w_2(\bar{\Delta}) = 1/(1+\bar{\Delta})$ give higher accuracy than the other weight functions.

Second, we evaluated the voting function $f_1(\mathbf{v},\mathbf{c})$ in (9) and other common voting functions. The results, shown in Table 2 on 200 training images, show that $f_1(\mathbf{v},\mathbf{c})$ performs better than three other voting functions.

Table 1. Comparison of different weight functions on 200 training images of the PLVP dataset.

	Weight function	e	std
$w_1(\bar{d})$	$\frac{1}{1+\bar{d}}$	0.0568	0.0768
	$\frac{1}{1+\bar{d}^2}$	0.0568	0.0769
	$\exp(-\bar{d})$	0.0541	0.0707
	$\exp(-\bar{d}^2)$	**0.0540**	**0.0711**
$w_2(\bar{\Delta})$	$\frac{1}{1+\bar{\Delta}}$	**0.0658**	**0.0907**
	$\frac{1}{1+\bar{\Delta}^2}$	0.0663	0.0917
	$\exp(-\bar{\Delta})$	0.0664	0.0916
	$\exp(-\bar{\Delta}^2)$	0.0664	0.0916

Table 2. Comparison of different VP voting functions on 200 training images of the PLVP dataset.

Voting function	e	std
$\frac{\exp(-d)}{1+\hat{\Delta}}$	**0.0534**	**0.0706**
1 [5]	0.0651	0.0898
$\frac{1}{1+d\times\hat{\Delta}}$ [11]	0.0591	0.0820
$\exp(-d\times\hat{\Delta})$ [14]	0.0578	0.0806

Third, we compared different histogram sizes for the Bayesian classifier described in (13). Note that a higher histogram size provides a finer pdf estimation, but requires more storage and training samples. In this experiment, 3500 training images were used to construct 4 different Bayesian classifiers with 32^2, 64^2, 128^2, and 256^2 bins. Table 3 shows the VPE errors on the 3500 training images. For comparison purpose, the VPE error obtained when using only VP voting function $f_1(\mathbf{v},\mathbf{c})$ (i.e. no Bayesian classifier) is also shown.

The results in Table 3 indicate that the Bayesian classifier needs at least 64^2 bins to improve VPE accuracy compared with using the voting function $f_1(\mathbf{v},\mathbf{c})$ alone. In addition, the best result is achieved with the 256^2 histogram bins. Therefore, in the subsequent experiments, a histogram size of 256^2 bins was used.

Fig. 2. Visual results of vanishing point estimation methods. Ground-truth vanishing point: red asterisk. Proposed method: white square. Edge-based [4]: yellow cross. Kong *et. al*. [6]: blue triangle. Moghadam *et. al*. [12]: cyan triangle. Phung *et. al*. [1]: green circle. See electronic color image.

Table 3. VPE error performance on 3500 training images for different histogram sizes of of the Bayesian classifier.

Configuration	e	std
32^2 bins classifier	0.0539	0.0713
64^2 bins classifier	0.0528	0.0706
128^2 bins classifier	0.0517	0.0656
256^2 bins classifier	0.0504	0.0639
No classifier, $f_1(\mathbf{v}, \mathbf{c})$ only	0.0538	0.0712

Table 4. Performance of vanishing point estimation methods on 500 test images of the PLVP dataset.

Method	e	std	t (s)
Edge-based [4]	0.1211	0.1771	0.0249
Kong *et. al* [6]	0.0811	0.1039	2.9982
Moghadam *et. al* [12]	0.2009	0.1032	0.5942
Phung *et. al* [1]	0.0710	0.0949	0.5739
Proposed method	**0.0503**	**0.0638**	**0.1315**

3.3. Comparison with other methods

The proposed vanishing point estimation method was compared with the edge-based method (Wang *et. al* [4]) and three texture-based methods (Kong *et. al*. [6], Moghadam *et. al*. [12], and Phung *et. al*. [1]).The evaluation was performed on 500 test images of the PLVP dataset.

Table 4 presents the performance of the above VPE methods on the test set. The proposed method ($e = 0.0503$, $std = 0.0638$) provides the best accuracy and stability. In terms of execution time, the proposed method ($t = 0.1315$ s) is significantly faster than three evaluated texture-based methods. The edge-based method has the shortest computation time (0.0253 s), but it has the highest e and std.

Figure 2 provides some visual results of the VPE methods on the test images. In most cases, compared with the other methods, the vanishing points detected with the proposed method (white squares) are closer to the ground-truth vanishing points (red asterisks).

4. CONCLUSION

An enhanced voting scheme is proposed for texture-based vanishing point estimation in unstructured road scenes. The method employs a new strategy to reduce the search space and computation time. The VP voting score includes new weight functions, which are selected adaptively using a Bayesian classifier. Overall, the proposed VPE method yields significant improvements in computation time and accuracy over several state-of-the-art approaches.

5. REFERENCES

[1] S. L. Phung, M. C. Le, and A. Bouzerdoum, "Pedestrian lane detection in unstructured scenes for assistive navigation," *Computer Vision and Image Understanding*, vol. 149, pp. 186–196, 2016.

[2] D. Liebowitz and A. Zisserman, "Metric rectification for perspective images of planes," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun 1998, pp. 482–488.

[3] P. Parodi and G. Piccioli, "3D shape reconstruction by using vanishing points," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 211–217, 1996.

[4] Y. Wang, E. K. Teoh, and D. Shen, "Lane detection and tracking using B-snake," *Image and Vision Computing*, vol. 22, pp. 269–280, 2004.

[5] C. Rasmussen, "Grouping dominant orientations for ill-structured road following," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, Jan 2004, pp. 470–477.

[6] H. Kong, J. Y. Audibert, and J. Ponce, "General road detection from a single image," *IEEE Transactions on Image Processing*, vol. 19, no. 8, pp. 2211–2220, 2010.

[7] A. Almansa, A. Desolneux, and S. Vamech, "Vanishing point detection without any a priori information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 4, pp. 502–507, 2003.

[8] Q. She, Z. Lu, and Q. Liao, "Vanishing point estimation for challenging road images," in *IEEE International Conference on Image Processing*, Oct 2014, pp. 996–1000.

[9] R. Duda and P. Hart, "Use of the Hough transform to detect lines and curves in pictures," *Community Associate Computing Machine*, vol. 15, pp. 1–15, 1972.

[10] R. G. von Gioi, J. Jakubowicz, J. M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 722–732, 2010.

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

[12] 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, pp. 425–430, 2012.

[13] O. Miksik, "Rapid vanishing point estimation for general road detection," in *IEEE International Conference on Robotics and Automation*, May 2012, pp. 4844–4849.

[14] H. Kong, S. E. Sarma, and T. Feng, "Generalizing Laplacian of Gaussian filters for vanishing-point detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 408–418, 2013.

[15] M. C. Le, S. L. Phung, and A. Bouzerdoum, "Pedestrian lane detection in unstructured environments for assistive navigation," in *International Conference on Digital lmage Computing: Techniques and Applications*, Nov 2014, pp. 1–8.

[16] J. Shi, J. Wang, and F. Fu, "Fast and robust vanishing point detection for unstructured road following," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 970–979, 2016.

[17] T. S. Lee, "Image representation using 2D Gabor wavelets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959–971, 1996.

[18] J. van de Weijer, T. Gevers, and A. W. M. Smeulders, "Robust photometric invariant features from the color tensor," *IEEE Transactions on Image Processing*, vol. 15, no. 1, pp. 118–127, 2006.

[19] J. Bigun, G. H. Granlund, and J. Wiklund, "Multidimensional orientation estimation with applications to texture analysis and optical flow," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 775–790, 1991.

[20] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. John Wiley & Sons, Inc., 2001.