Fast and Robust Vanishing Point Detection for Unstructured Road Following

Jinjin Shi, Jinxiang Wang, and Fangfa Fu

***Abstract*—Vision-based unstructured road following is a challenging task due to the nature of the scene. This paper describes a novel algorithm to improve the accuracy and robustness of vanishing point estimation with very low computational cost. The novelties of this paper are three aspects: 1) We use joint activities of four Gabor filters and confidence measure for speeding up the process of texture orientation estimation. 2) Misidentification chances and computational complexity of the algorithm are reduced by using a particle filter. It limits vanishing point search range and reduces the number of pixels to be voted. The algorithm combines the peakedness measure of vote accumulator space with the displacements of moving average of observations to regulate the distribution of vanishing point candidates. 3) Attributed to the design of a noise-insensitive observation model, the proposed system still has high detection accuracy even when less than 60 sparsely distributed vanishing point candidates are used for voting as the influence introduced by the stochastic measurement noise of vote function and the sparsity of the vanishing point candidates is reduced. The method has been implemented and tested over 20 000 video frames. Experimental results demonstrate that the algorithm achieves better performance than some state-of-the-art texture-based vanishing point detection methods in terms of detection accuracy and speed.**

***Index Terms*—Road following, particle filter, vanishing point tracking, unstructured road detection.**

I. Introduction

USING computer vision techniques to develop perceptual algorithm for autonomous ground vehicle to follow unstructured roads is a challenging task as there are hardly any invariant features that can characterize the road. Over the past few decades, numerous approaches have been developed. The focus of much of early systems has been the algorithms of paved road following based on specific visual characteristics of the road surface, such as color contrasts or edges in the input image. Lane detection on structured environments is largely considered a "solved" problem and more recent research attempts to handle unstructured road conditions. The difficulty in unstructured road detection using monocular vision is that the detection algorithm must be able to deal with complex real-road scenes, such as the presence of pedestrians or vehicles on the road, different road types with varieties of colors, varying illumination and changing weather conditions etc. Furthermore, intelligent transportation systems generally require fast processing since vehicle speed is bounded by the processing rate [1]. A host of researchers utilized vanishing point for road following or as a directional constraint for defining the path boundaries. Vanishing point is defined as the point of intersection of the perspective projections of a set of parallel lines in 3D scene onto the image plane. For a straight road segment [Fig. 1(a)], vanishing point is obtained as the intersection point of the lines that characterize the lane. In the case of a curved road, vanishing point is approximated by the lane borders, markings etc. in the vicinity of the vehicle [Fig. 1(b)]. Its distance components to the image center yield the relative direction (yaw and pitch angles) of the vehicle with respect to the path. Therefore, estimated vanishing point can be utilized to steer the vehicle [2] or as a directional constraint for drivable region segmentation [3], lane marking detection [4] etc.

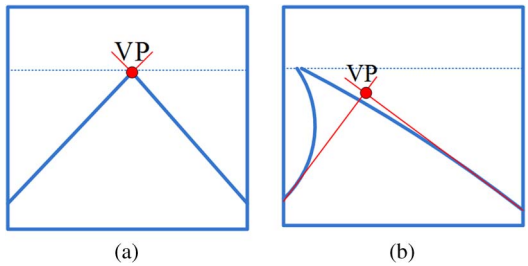


Fig. 1. Vanishing point in (a) straight and (b) curved roads.

In this paper we propose a novel algorithm to reduce the computational cost of vanishing point estimation and harden its robustness against strong edge interference. Specifically, we use joint activities of four Gabor filters and confidence measure to speed up texture orientation computation. The pixels with low confidence level are discarded and the remaining are used for vanishing point estimation. In addition, an efficient and effective vanishing point tracking algorithm is proposed. The proposed system simply uses the peakedness measure of VAS together with the displacement of moving average of vanishing point observations to regulate the distribution of vanishing point candidates. However, tracking the global vanishing point $V_{\max}$ of the entire image from frame to frame is unstable if only a small number of vanishing point candidates are used for voting because of the sparsity of the pixels to be voted and the measurement noise introduced by the winner-take-all mechanism of the vote function. To overcome this weakness, we take the moving average of vanishing point observations to reduce the influence. As a result, the proposed system still has high

Manuscript received October 22, 2014; revised January 28, 2015, April 7, 2015, June 3, 2015, and August 8, 2015; accepted October 8, 2015. Date of publication October 26, 2015; date of current version March 25, 2016. The Associate Editor for this paper was M. Bertozzi. *(Corresponding author: Jinxiang Wang.)*

The authors are with the Department of Microelectronics, Harbin Institute of Technology, Harbin 150090, China (e-mail: shijinjinhit@sina.com; jxwang@hit.edu.cn; fff1984292@163.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2015.2490556

detection accuracy even though less than 60 vanishing point candidates are used for voting. Thus the computational complexity is greatly reduced. The method has been implemented and tested over 20000 video frames. Qualitative and quantitative analyses demonstrate that the method provides higher accuracy and efficiency when compared with some state-of-the-art textured-based vanishing point detection methods. We have posted in http://www.adrive.com/public/DCX44b/video.zip six video clips of our detection results under six different scenarios.[1]

The remainder of this paper is organized as follows. We first review some relevant works in Section II. The proposed algorithm is then detailed in Section III. We evaluate the performance of the proposed algorithm in Section IV. Finally, we draw some conclusions in Section V.

II. Related Work

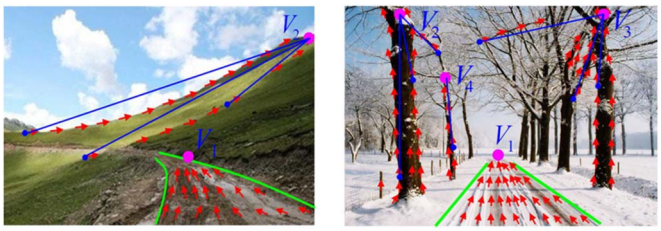
Generally, a road image can be classified into a structured or unstructured one. For structured roads (e.g., highway or city roads), there are distinct features, such as strong color contrasts or remarkable edges in the input image that can help computer to detect the road by many strategies. For example, region-based methods such as those presented by Alvarez and Lopez [5], Alon *et al.* [6], and Cheng *et al.* [7] are often used in urban roads to detect drivable area for autonomous vehicles. These methods use color contrast to group pixels or super-pixels together with similar color features to determine pixels belong to the road area or the background. A good contrast even aids these methods working well on unpaved rural roads. Edge-based approaches, such as those described in [8]–[12], typically work best on well-painted roads with high-contrast contours suited for edge detection. In contrast, unstructured roads (e.g., rural or desert) are not been well-paved, lack clear lane markings or boundaries, making road detection task very challenging. For unstructured roads, Lookingbill *et al.* [13] combined self-supervised learning and optical flow techniques to classify a drivable region for autonomous robot navigation, but the results were not robust enough on chaotic roads. In [14], [15], features were learned using convolutional neural network for road scene segmentation from a single image. However, the off-line learning needs a large amount of manually labeled training data, which is time-consuming in practice. Huang *et al.* [16] proposed a method based on HSV color space and road features. The system ROBOG [17] utilized uniformity measure to recognize road lanes. Although unstructured roads are not characterized by strong edges or distinct features, they often have texture cues parallel to the road direction in the form of edges, border lines, or ruts and tracks left by other vehicles on the road. These cues appear to converge into a single vanishing point. Therefore, many vanishing point based approaches have been proposed recently.

Texture-based method, on the other hand, utilizes texture orientations computed with steerable filters to vote for the location of road's vanishing point. Rasmussen [18] applied 72 oriented Gabor filter banks to achieve precise orientation estimation at each pixel. Then vanishing point was estimated by global hard-voting. However, this scheme tends to favor points that are high in the image, leading pixels in the upper part of the image to receive more votes than the lower ones. To overcome this weakness, Kong *et al.* [19] proposed an adaptive soft voting scheme which takes into account the distance between vanishing point candidate and voter. But in order to achieve precise orientation estimation, they utilized 5-scale and 36-orientation Gabor filters to convolute with the input image. Moghadam *et al.* [20] suggested using single scale 4-orientation Gabor filter banks to speed up the process of texture orientation computation. They estimated the orientations of all of the pixels regardless with or without apparent dominant orientations. In fact, there is not too much difference between using 5-scale, 36-orientation Gabor filters or just single scale, 4-orientation Gabor filters since the post-processing algorithm of [20] is time-consuming [21]. Apart from that, we show in this paper that some improvements still can be made (further discussed in Section III). In [22], a novel method for vanishing point detection and tracking is introduced by using adaptive steerable filter banks and linear road model. The algorithm updates the filter bank so that the preferred orientations of the filters that will be used in the next frame are the estimated orientations of the lane markings. More recently, Wang *et al.* [23] proposed a vanishing point detection approach which needs neither magnitude threshold selection for edge detection nor scale parameter turning for texture analysis. Miksik *et al.* [24] decomposed Gabor wavelets into a linear combination of Haar-like box filters to perform filtration in integral image domain and utilized super-pixels in voting scheme to provide a significant speed-up at the expense of precision loss. In [25], vanishing point is estimated by road-type-specific voting which focuses on adjusting weights to balance out the accumulative votes at higher area. Ebrahimpour *et al.* [26] imported minimum possible information to Hough space for fast vanishing point estimation. In [27], Wu *et al.* proposed an example-based vanishing point assisted path detection method. Vanishing point candidates were initially detected directly from neighboring images and then refined by an iterative process.

Although the accuracy of vanishing point detection is promising based on pixel-wise voting, it is computationally expensive during the voting stage because each pixel can be regarded as both a voter and a vanishing point candidate, i.e., each pixel of an image can vote for any pixels above it. Recently, Kong *et al.* [3], [19], [28] reduced the computational amount by incorporating confidence measure into vanishing point estimation. Similarly, Miksik *et al.* [24] performed a morphological dilation to include the pixels close to huge support region. However, vanishing point detection from single image is extremely sensitive to noise. Considering the case that a road scene exists some interference with more strong edges than the tracks left by other vehicles, then these edges may induce an incorrect vanishing point estimation [21] (see Fig. 2).

A large number of approaches for robust vanishing point tracking have been proposed. For example, Moghadam *et al.* [30] used Rao-Blackwellised particle filter to track the location of the vanishing points $V_{\max}$ over consecutive frames. Suttorp *et al.* [29] developed a data-driven and model-based filtering module for driver assistance systems. The robustness

[1]Or available from http://pan.baidu.com/s/1sjwxDYP



(a) (b)

Fig. 2. Influence of strong edges in voting. V_1 is the location of ground truth vanishing point. V_3, V_3, and V_4 are the locations of local maximum of votes. The red arrows denote the orientations of discrete points. Since the dominant orientations of vegetation or mountains etc. have strong points of convergence, V_2-V_4 may receive more votes than V_1, resulting in incorrect estimation if only single image is used.

of vanishing point estimation is increased by online adaptation of the parameters of both the data-driven as well as the model-driven processing units. The noise variance of the Kalman-Filter for a single measurement is determined using the confidence measure of actual vanishing point estimation. Wang *et al.* [31] performed vanishing point detection on horizontally divided image strips. Then image feature measurement together with road width likelihood are integrated for vanishing point tracking. A similar strategy to our algorithm also has been adopted by Rasmussen within the RoadCompass system [2], [18]. They partitioned voting procedure into strips parallel to the vanishing line and then tracked the points along vanishing point contour from frame to frame. Our approach is much efficient, which only needs peakedness and displacement information for tracking.

III. Methods

Our approach broadly consists of three significant components: (1) computation of dominant texture orientation at every image pixel with certain confidence level, (2) vanishing point candidates selection through particle filtering, and (3) vanish point voting. Below, we describe each stage in detail.

A. Texture Orientation Extraction

In this subsection, the dominant orientation estimation method in [20] is first reviewed and then ours is presented. The dominant orientation $\theta(p)$ at pixel $p=(x,y)$ of an image is the direction that describes the strongest local texture flow. 2-D Gabor filter banks are often used to get $\theta(p)$. First, four Gabor energy responses are calculated as in [20] with $\phi \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$. The Gabor filter with preferred orientation ϕ and radial frequency $\omega_0 = 2\pi/\lambda$ can be written as [32]

$$g_{\phi,\omega_0}(x,y) = \frac{\omega_0}{\sqrt{2\pi}c} e^{-{\omega_0}^2(4a^2+b^2)/8c^2} \left(e^{ia\omega_0} - e^{-c^2/2}\right) \quad (1)$$

where $a = x\cos\phi + y\sin\phi$, $b = -x\sin\phi + y\cos\phi$, $c = \pi/2$ is a constant, and the spatial frequency λ is set to $4\sqrt{2}$. The Gabor energy responses at each pixel are obtained through the convolution of a grayscale input image I with a bank of Gabor filters with predefined orientations, i.e.,

$$I_{\phi,\omega_0}(p) = I(p) \otimes g_{\phi,\omega_0}(p). \quad (2)$$

According to the relationship between convolution and Fourier transform, (2) becomes

$$I_{\phi,\omega_0}(p) = \mathscr{F}^{-1}\left\{\mathscr{F}\left\{I(p)\right\} \bullet \mathscr{F}\left\{g_{\phi,\omega_0}(p)\right\}\right\} \quad (3)$$

where $\mathscr{F}$ and $\mathscr{F}^{-1}$ denote Fourier and inverse Fourier transforms, respectively. Normally, I_{ϕ,ω_0} has two components, a real and an imaginary parts. Then, in order to provide an accurate dominant orientation estimation, the Gabor energy response E_{ϕ,ω_0} is calculated using the following equation:

$$E_{\phi,\omega_0}(p) = \sqrt{\mathrm{Re}\left(I_{\phi,\omega_0}(p)\right)^2 + \mathrm{Im}\left(I_{\phi,\omega_0}(p)\right)^2}. \quad (4)$$

To measure Gabor energy responses' peakedness near their optimum angle, the four energy responses obtained from (4) are first sorted based on their magnitudes in descending order $E^1_{\phi,\omega_0}(p) > E^2_{\phi,\omega_0}(p) > E^3_{\phi,\omega_0}(p) > E^4_{\phi,\omega_0}(p)$ and the notation ϕ_i is assigned to denote the preferred orientation of the Gabor filter for the computation of $E^i_{\phi,\omega_0}(p)$. If the values of $E^1_{\phi,\omega_0}(p)$ and $E^4_{\phi,\omega_0}(p)$ are significantly different, then the orientation of $p(x,y)$ is reliable, otherwise, there is not an apparent dominant orientation associated with it, which often appears in nonroad regions. We do not, like [20], compute the orientations of all of the pixels in a road image, but reduce the computational complexity by defining a confidence level:

$$\mathrm{Conf}(p) = \begin{cases} 1 - \frac{E^4_{\phi,\omega_0}(p)}{E^1_{\phi,\omega_0}(p)}, & E^1_{\phi,\omega_0}(p) > E_{\mathrm{th}} \\ 0, & E^1_{\phi,\omega_0}(p) \le E_{\mathrm{th}} \end{cases} \quad (5)$$

where E_{th} is Gabor energy response threshold. We discard all of the pixels having confidence scores lower than T_c, and the remaining, also called reliable pixels, are kept as voters that are used to detect the vanishing point. The thresholds E_{th} and T_c are empirically determined through experiments on our test images, and $E_{\mathrm{th}} = 0.1$, $T_c = 0.85$ result in highest detection accuracy. By comparison, our confidence level definition is much simpler than the one described in [19] and [33] which choose the average of the responses from $E^5_{\phi,\omega_0}(p)$ to $E^{15}_{\phi,\omega_0}(p)$ as the mean of local maximum responses. Although using joint activities of four Gabor filters to estimate texture dominant orientations has been studied by Moghadam [20], the algorithm still needs some improvements. In [20], the two most dominant filter activation strengths $E^1_{\phi,\omega_0}(p)$ and $E^2_{\phi,\omega_0}(p)$ are used to calculate the local texture orientation $\hat{\theta}'(p)$ as follows:

$$\mathbf{V}(p) = V_x(p) + jV_y(p) = \sum_{i=1}^{2} E^i_{\phi,\omega_0}(p) e^{j\phi_i(p)} \quad (6)$$

$$\hat{\theta}'(p) = \tan^{-1}\left(\frac{V_y(p)}{V_x(p)}\right). \quad (7)$$

Consider the case that the orientations of $E^1_{\phi,\omega_0}(p)$ and $E^2_{\phi,\omega_0}(p)$ are 135° and 0°. Fig. 3 depicts the dominant orientation $\hat{\theta}'(p)$ estimated by (6) and (7). In fact the real one is in between 135° and 180°, according to the symmetry of Gabor

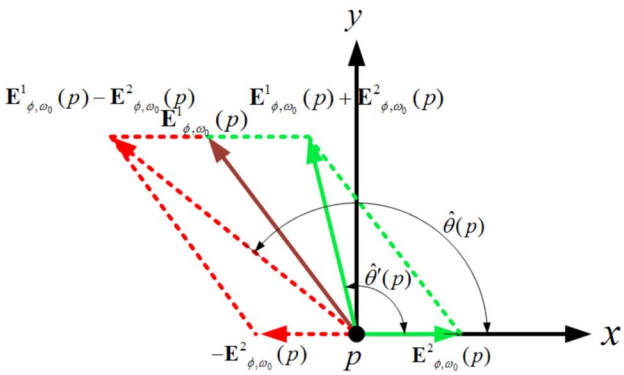


Fig. 3. Illustration of texture dominant orientation estimation using two different algorithms. $E^1_{\phi,\omega_0}(p)$ and $E^2_{\phi,\omega_0}(p)$ are the two most dominant Gabor energy responses. The texture orientations computed by our algorithm and the algorithm presented in [20] are denoted as $\hat{\theta}(p)$ and $\hat{\theta}'(p)$, respectively.

wavelets. Here, we propose to use the following equations for dominant orientation estimation.

(a) If $\phi_1 = 135^\circ$ and $\phi_2 = 0^\circ$,

$$\begin{cases} V_x(p) = E^1_{\phi,\omega_0}(p)\cos\phi_1 - E^2_{\phi,\omega_0}(p) \\ V_y(p) = E^1_{\phi,\omega_0}(p)\sin\phi_1. \end{cases} \tag{8}$$

(b) Otherwise

$$\begin{cases} V_x(p) = E^1_{\phi,\omega_0}(p)\cos\phi_1 + E^2_{\phi,\omega_0}(p)\cos\phi_2 \\ V_y(p) = E^1_{\phi,\omega_0}(p)\sin\phi_1 + E^2_{\phi,\omega_0}(p)\sin\phi_2. \end{cases} \tag{9}$$

Then the dominant orientation $\hat{\theta}(p)$ can be obtained by (7). To speed up the process, a lookup table is used for (7). Meanwhile, the computational amount is greatly reduced as many pixels with low confidence scores have been discarded.

B. Vanishing Point Tracking

Once texture orientations are estimated, each reliable pixel $p(x, y)$ of an image with a texture orientation of $\hat{\theta}(p)$ can vote for any pixels above it, thus the voting stage is still time consuming [28]. Apart from that, traditional voting schemes (e.g., [19], [20]) are sensitive to strong edge interference as the pixel which receives the largest number of votes is selected as road vanishing point. In this subsection, we propose a novel vanishing point tracking algorithm that is based on peakedness measurement and particle filtering technique to reduce the computational cost and improve the accuracy and stability of vanishing point estimation.

1) Particle Filtering: Let the location of vanishing point be a stochastic process. The state of vanishing point in terms of discrete time k is denoted as $\mathbf{x}_k$ and its history $\mathbf{x}_{0:k} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k\}$. Similarly, the observation obtained through voting (discussed in next subsection) is $\mathbf{z}_k$ with history $\mathbf{z}_{1:k} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k\}$. Given the states $\mathbf{x}_{0:k}$ and the observations $\mathbf{z}_{1:k}$ up to time k, vanishing point tracking can be interpreted as a posterior density estimation $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$, where $\mathbf{x}_{0:k}$ is hidden and can only be estimated through the observations $\mathbf{z}_{1:k}$. Particle filtering is an on-line posterior density estimation algorithm that obtains an approximation to the posterior distribution at time $k-1$ by a set of weighted samples $\{\mathbf{x}^{(i)}_{0:k-1}, w^{(i)}_{k-1}\}^N_{i=1}$, also called particles, and recursively updates these particles to approximate the posterior distribution at the next time step $p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k})$. A particle with large weight is likely to be drawn multiple times and conversely a particle with very small weight is almost not likely to be drawn. In vanishing point tracking, particle weights will have great changes if stochastic observation noise is increased or decreased dramatically and suddenly, which is usually the case if only less than 60 vanishing point candidates are used for voting, and the raw maximum of vote function is noisy, leading to low tracking accuracy. Although the observation noise could easily be handled in the sensor model $p(\mathbf{z}_k|\mathbf{x}^{(i)}_k)$ by assuming a higher-variance on measurement dynamics, vanishing point tracking accuracy and stability will be reduced and the chances of misidentification will be increased accordingly. An effective strategy to reduce the influence of the stochastic observation noise is to take the moving average of vanishing point observations based on the fact that genuine vanishing points shift only slightly between frames as vehicle moves. In this paper, the observation which relates to the state vector $\mathbf{x}_k$ at time k is modeled in the form:

$$\overline{\mathbf{z}}_k = \frac{1}{n}\sum_{j=k-l}^{k}\mathbf{z}_j = \mathbf{x}_k + \mathbf{r}_k. \tag{10}$$

Here $\mathbf{r}_k$ is moving average measurement noise vector. l represents the number of discrete time periods, it is inversely proportional to the frame rate of road images as well as the running speed of the ego vehicle. Since they are unknown for the test video images, $l = 20$ is empirically determined through our experiments. According to Bayes' rule, the posterior density at k can be approximated as [34], [35]

$$p(\mathbf{x}_{0:k}|\overline{\mathbf{z}}_{1:k}) \approx \sum_{i=1}^{N} w^{(i)}_k \delta\left(\mathbf{x}_{0:k} - \mathbf{x}^{(i)}_{0:k}\right) \tag{11}$$

where

$$w^{(i)}_k \propto w^{(i)}_{k-1}\frac{p\left(\overline{\mathbf{z}}_k|\mathbf{x}^{(i)}_k\right)p\left(\mathbf{x}^{(i)}_k|\mathbf{x}^{(i)}_{0:k-1}\right)}{q\left(\mathbf{x}^{(i)}_k|\mathbf{x}^{(i)}_{0:k-1}, \overline{\mathbf{z}}_k\right)}. \tag{12}$$

Assume that the measurement noise $p(\mathbf{r}_k|\mathbf{x}^{(i)}_k)$ is a zero mean Gaussian white noise with variance $\boldsymbol{\sigma}_{r,k}$, we could model the measurement density as:

$$\begin{aligned} p\left(\overline{\mathbf{z}}_k|\mathbf{x}^{(i)}_k\right) &= p_r\left(\overline{\mathbf{z}}_k - \mathbf{x}^{(i)}_k|\mathbf{x}^{(i)}_k\right) \\ &= \exp\left(-\frac{1}{2\boldsymbol{\sigma}^2_{r,k}}\left|\overline{\mathbf{z}}_k - \mathbf{x}^{(i)}_k\right|^2\right). \end{aligned} \tag{13}$$

To deal with degeneracy, the samples $\{\mathbf{x}^{(i)}_k\}_{i=1:N}(N = 60)$ are then resampled to generate an unweighted new particle set $\{\widetilde{\mathbf{x}}^{(i)}_k\}_{i=1:N}$ according to their importance weights. After resampling, (12) reduces to:

$$w^{(i)}_k \propto \exp\left(-\frac{1}{2\boldsymbol{\sigma}^2_{r,k}}\left|\overline{\mathbf{z}}_k - \widetilde{\mathbf{x}}^{(i)}_k\right|^2\right)w^{(i)}_{k-1}. \tag{14}$$

Finally, an estimate of vanishing point location is given by:

$$\hat{\mathbf{x}}_k = \frac{1}{N}\sum_{i=1}^{N}\widetilde{\mathbf{x}}_k^{(i)}. \tag{15}$$

The averaging effect of (15) further reduces the influence of the stochastic observation noise introduced by vote function. In the next time step $k+1$, a new sample $\{\mathbf{x}_{k+1}^{(i)}\}_{i=1}^{N}$ is drawn around $\hat{\mathbf{x}}_k$ according to the following nonlinear function:

$$\mathbf{x}_{k+1}^{(i)} = \hat{\mathbf{x}}_k + \mathbf{q}_k \tag{16}$$

Where $\mathbf{q}_k \sim N(\mathbf{0}, \boldsymbol{\sigma}_{q,k})$, $\boldsymbol{\sigma}_{q,k}$ is process noise vector controlling the distribution of a new sample. Note that, $\{\mathbf{x}_{k+1}^{(i)}\}_{i=1}^{N}$ are locations of the pixels to be voted.

2) Particle Distribution Regulation: Regulation of particle distribution is a crucial step within the vanishing point tracking process. In this paper, particles of the filter are tuned by measuring the sharpness of the peak in VAS and the magnitude of vanishing point displacement between frames. If the peakedness of VAS is low or the magnitude of vanishing point displacement is large (e.g., tracking a curved road), then their search range should be increased. Thereby, vanishing point can be tracked timely or a wrong estimation can be recovered. In contrast, high sharpness and small displacement indicate that estimated vanishing point is reliable. Limiting search range appropriately can reduce the computational complexity and the chances of misidentification of vanishing point greatly. The distribution of particles at time $k+1$ is controlled by the process and measurement noise variances $\boldsymbol{\sigma}_{q,k}$ and $\boldsymbol{\sigma}_{r,k}$, which are determined by the peakedness measure and the displacement of moving average of vanishing point observations

$$\begin{aligned}\boldsymbol{\sigma}_{q,k} = \boldsymbol{\sigma}_{r,k} = \max\big(\min\big((a|\overline{\mathbf{z}}_{k-1} - \overline{\mathbf{z}}_{k-2}| \\ + (1-a)b_1^n\mathbf{I}_{2\times 1})\boldsymbol{\sigma}_0, \boldsymbol{\sigma}_{\max}\big)\big), \boldsymbol{\sigma}_{\min}\big)\end{aligned} \tag{17}$$

where b_1, $\boldsymbol{\sigma}_0$, $\boldsymbol{\sigma}_{\min}$, $\boldsymbol{\sigma}_{\max}$ are constants, they are empirically set to 1.5, 44, 10, and 1000 accordingly. $\min(\mathbf{X})$ and $\max(\mathbf{X})$ return the minimum and maximum elements in $\mathbf{X}$, respectively. The constant parameter $a(=0.91)$ balances the importance between the displacement and the peakedness of vanishing point observations. n is the number of images with peakedness lower than a given threshold $T_U(=10^{-3})$ over consecutive frames. The reason of using b_1^n rather than a direct measurement of the peakedness (e.g., $1/KL$) is that it is noisy and the performance of particle filtering would be affected. The peakedness served as an indicator of the existence of a reliable vanishing point. One way to assess the peakedness is to compute KL divergence, which is a measure of the distance between two probability distributions on a random variable. The KL divergence was also used in [2], but it is just for failure detection, not the same purpose described in this paper. Formally, given two probability distributions $g(x)$ and $q(x)$ over a discrete random variable X, the KL divergence given by $KL(g/q)$ is defined as follows:

$$KL\left(\frac{g}{q}\right) = \sum_{x\in X} g(x) ln\left(\frac{g(x)}{q(x)}\right). \tag{18}$$

Here $g(x)$ is taken from the vote function which is discussed in next subsection, $q(x) = \sum_{(u,v)\in\{X_{k-1}^{(i)}\}_{i=1}^{N}} \text{Votes}(c(u,v))/N$ is the average of the votes received by vanishing point candidates. However, the information of vanishing point candidate distribution is not incorporated in (18). Note that the surface of our VAS is only spanned by 60 sparsely distributed vanishing point candidates. Normally, the wider the distribution is, the lower the peakedness would be. In order to incorporate distribution information into peakedness measure, the following equation is proposed

$$U\left(\frac{g}{q}\right) = \frac{1}{\sqrt{||\boldsymbol{\sigma}_{q,k-1}||_1}}\sum_{x\in X} g(x) ln\left(\frac{g(x)}{q(x)}\right) \tag{19}$$

where $||\mathbf{X}||_1$ is the 1-norm of $\mathbf{X}$. Finally, the measured vanishing point $\mathbf{z}_k$ is considered reliable when $U(g/q)$ is larger than a given threshold T_U determined through experiments. The effect of the proposed noise-insensitive observation model on improving vanishing point tracking quality can be further understood from the equations (14), (16) and (17). Considering the case that $\overline{\mathbf{z}}_k$ is replaced with $\mathbf{z}_k$, the particle weight $w_k^{(i)}$, $\boldsymbol{\sigma}_{q,k}$ and $\boldsymbol{\sigma}_{r,k}$ will have great changes between frames. As a result, the estimated vanishing point tends to lose or the chances of misidentification will be increased, leading to low tracking accuracy and stability.

C. Voting Scheme

With texture dominant orientations and the sample set $\{\mathbf{x}_{k+1}^{(i)}\}_{i=1:N}$, a raw vanishing point is estimated by voting according to the method presented in this subsection. Initially, the samples are distributed uniformly in order to coarsely localize a vanishing point. We round the value of each sample $\mathbf{x}_{k+1}^{(i)}|_{i=1:N}$ to its nearest integer value. For each potential vanishing point location, we only use reliable pixels below it to vote for it. Note that the number of pixels to be voted is normally less than the total number of the sample set $\{\mathbf{x}_{k+1}^{(i)}\}_{i=1:N}$ since there are several identical copies. The resampling algorithm used in this paper are based on multinomial selection of the particles from the original set with $p(\widetilde{\mathbf{x}}_k^{(i)} = \mathbf{x}_k^{(i)}) = w_k^{(i)}$. The resampling operation biases the more heavily weighted particles, therefore some vanishing point candidates may well be chosen repeatedly. In fact, the smaller the $\boldsymbol{\sigma}_{q,k}$ and $\boldsymbol{\sigma}_{r,k}$ are, the greater the chances of repetition and the more efficient the voting will be.

The vote function presented in [2] and [18] tends to favor points that are high in the image, leading those in the upper part of an image receiving more votes than the lower ones. In order to overcome this bias, the distance-based voting scheme proposed by Moghadam [20] is utilized. This vote function gives more votes to the candidates that are closer to it along its ray. The Euclidean distance between the pixel $p(x,y)$ and the vanishing point candidate $c(u,v)$ is $d(p) = \sqrt{(x-u)^2+(y-v)^2}$. Here, (u,v) is the location of a vanishing point candidate in the image. The Euclidean distance $\hat{d}(p)$ is then normalized by the maximum possible distance $D(p)$ between the voter $p(x,y)$ and the intersection point of the

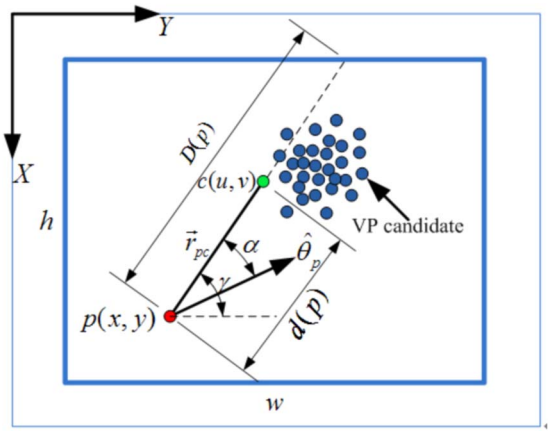


Fig. 4. Diagram of our voting scheme. Each reliable pixel of the image can vote for any vanishing point candidates above it. The locations of the vanishing point candidates are specified by the value of the sample set $\{\mathbf{x}_{k+1}^{(i)}\}_{i=1:N}$ after a round operation, and their distribution is controlled by $\boldsymbol{\sigma}_{q,k}$.

extension of the ray $\vec{r}_{pc}$ with the image boundaries (depending on the angle γ), as Fig. 4 depicts

$$l(p) = \begin{cases} \frac{(w-y)}{\cos(\gamma)}, & 0^\circ \le \gamma < 90^\circ \\ x, & \gamma = 90^\circ \\ -y/\cos(\gamma), & 90^\circ < \gamma < 180^\circ \end{cases} \tag{20}$$

$$D(p) = \begin{cases} l(p), & l(p)\sin(\gamma) \le x \\ \frac{x}{\sin(\gamma)}, & l(p)\sin(\gamma) > x \end{cases} \tag{21}$$

$$\hat{d}(p) = \frac{d(p)}{D(p)}. \tag{22}$$

Where γ is the angle the ray $\vec{r}_{pc}$ makes with horizontal line, w denotes the width of the road image. If the angle (α) between the texture dominant orientation $\hat{\theta}_p$ of the pixel p and its ray $\vec{r}_{pc}$ is smaller than a given threshold α_T $(= 15^\circ)$, then the vanishing point candidate $c(u, v)$ gains score given by

$$\text{Vote}\,(c(u,v)) = \exp\left(-\hat{d}(p)/2\sigma^2\right)\sin(\hat{\theta}_p). \tag{23}$$

The $\sin(\hat{\theta}_p)$ could be viewed as a coefficient that penalizes the vanishing point candidates who tend to receive votes from horizontal texture orientations. The distance function $\exp(-\hat{d}(p)/2\sigma^2)$ is designed to give more votes to the vanishing point candidates that are closer to them. Finally, the candidate which receives the largest votes is selected as a raw vanishing point, and its location corresponds to the observation $\mathbf{z}_k$. Although the vanishing point candidates to be voted are sparsely distributed, the proposed algorithm still has high detection accuracy, which is attributed to the design of the noise-insensitive observation model (10). It is worth noting that tracking of a vanishing point is lost if there is no road or a road is outside the camera field of view temporally. In this case, vanishing point search range is automatically enlarged until it is reacquired.

IV. Experimental Results

In this section, we evaluate the performance of our vanishing point tracking method quantitatively and qualitatively. The test videos come from a variety of uncalibrated cameras mounted with unknown height and tilt. For comparison purposes, the three best known texture-based vanishing point detection methods, i.e., Rasmussen [18], Kong [19] and Moghadam [20], are re-implemented using MATLAB. It is worth mentioning that some of them (e.g., [19], [20]) are based on single image. In addition, although the method in [18] is based on tracking, its naive implementation, i.e., detecting vanishing point solely relies on single image, is also investigated. This allows comparing in depth the performance of different kinds of voting strategies. Meanwhile, we did not tune any parameters as we are only interested in investigating the adaptation of these algorithms to different types of roads. In addition, we use (8) and (9) for [20] in calculating the texture of the pixels with apparent dominant orientations. Based on empirical observation, images with 61×81 resolution gave enough good results; hence all the images are first normalized to the same fixed size of 61×81 pixels for computational consideration and then passed to the algorithms. Finally, locations are projected back to the original images for evaluation. All of the results are tested on a 1.8 GHz Pentium Dual machine with 1 GB RAM. The estimated vanishing points are compared with the ground-truth which are manually marked by 6 adults who have the experience of driving after a brief description of the basic concept of road vanishing point. The method is tested over 20000 frames. These videos include desert, rural and snow-covered or well-painted road with large variations in illumination, color, and texture. Among them, more than 6000 frames contain strong-edge interference, such as vegetation, barriers, mountains and surrounding buildings. Each participant was told to select 80 representative challenging frames from one of the six videos. Total 480 were selected. There are 480×6 points in total marked by the participants. A GUI program recorded their marked coordinates. A median filter is applied to these coordinates (x and y respectively) for each image and the median As there are an even number of marked points, the median is the average of the middle two annotations) are used as an initial ground-truth position $(G(g_x, g_y))$. The number of manually marked points on either side of g_x is counted. Then a median filter is applied again to the x coordinate of the marked points on the side with larger amount (If their amount is equal, the more compact (i.e., lower variance) side is used), and the median is used as q_x. The same process is applied for q_y. A red circle centered at $Q(q_x, q_y)$ is overlaid on the road image. The radius of the circle $R = 0.1L$, where L is diagonal length of the road image. If there were points outside the circle, then we removed them. 6.1% of marked points were removed. The horizontal and vertical mean $(\overline{\sigma}_x, \overline{\sigma}_y)$ and maximum $(\sigma_{x\max}, \sigma_{y\max})$ standard deviations of the normalized distance errors of the remaining human annotators are (0.012, 0.017) and (0.064, 0.080) respectively, which indicates that most of them are fairly tight. For each image, a median filter is applied to the rest coordinates again, and the median is used as a ground-truth position.

A. Accuracy Comparison

Fig. 5 shows the basic steps of vanishing point localization on a road image. Fig. 6 illustrates the evolution of vanishing point

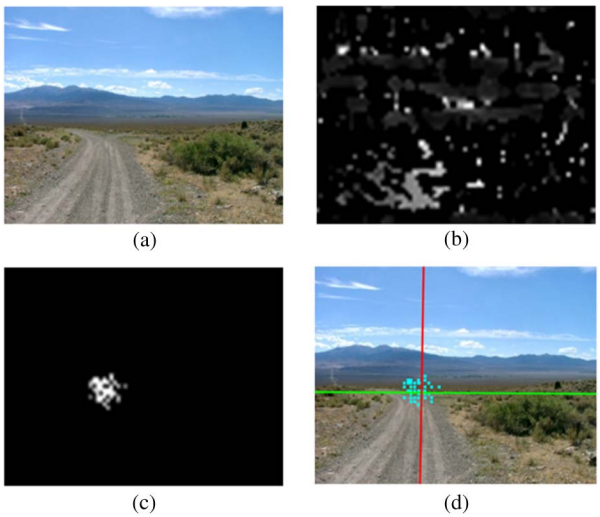


Fig. 5. Steps of vanishing point localization on a road image. (a) Original road image. (b) Extracted dominant orientations ($[0, \pi]$ radians mapped to [0, 255] intensity values), the orientations of pixels with low confidence scores are set to zero based on the knowledge discussed in Section III. (c) Voting accumulator space. The intensity is proportional to the votes of vanishing point candidates received. (d) Distribution of vanishing point candidates and estimated vanishing point location, its vertical position marks the horizon line of road plane and its horizontal position indicates the road direction. Note that the images in (b) and (c) have been resized to the same size as (a).

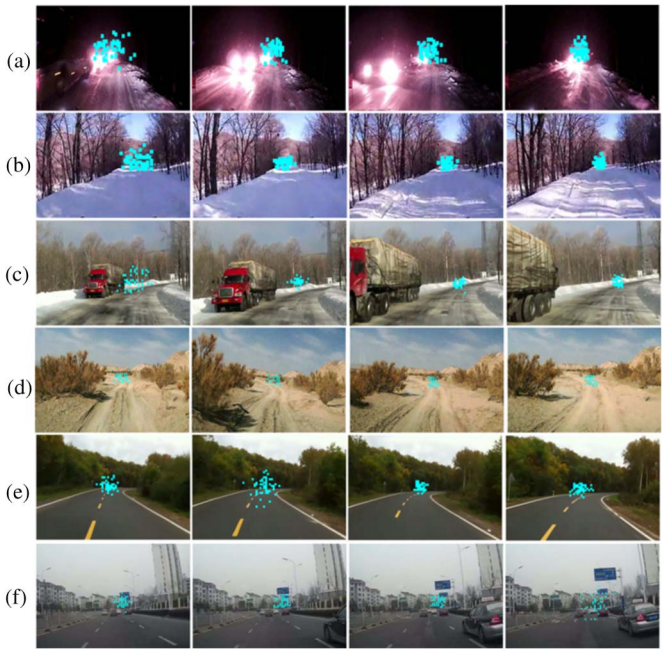


Fig. 6. Evolution of vanishing point candidates on unstructured roads with week (a), (b) or strong texture (c), (d), and paved roads with noticeable (e) or sparser lane markers (f).

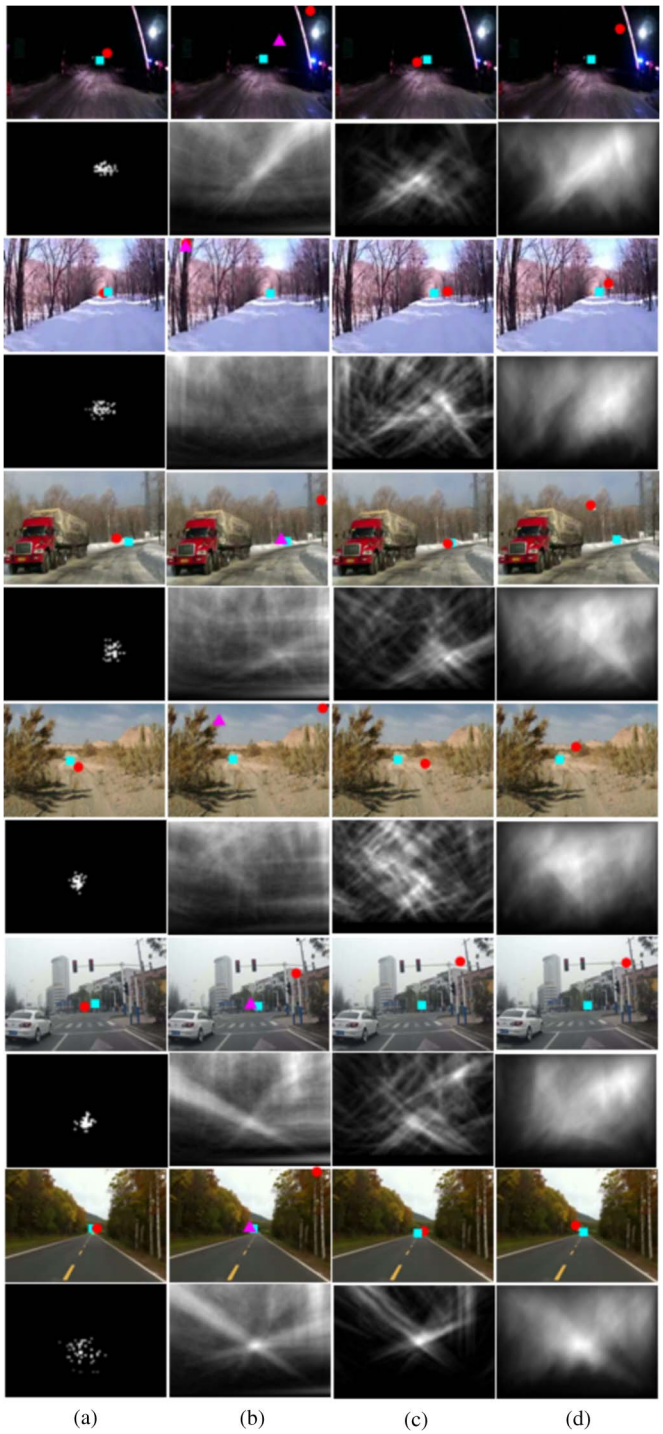


Fig. 7. Examples of vanishing point estimation. Cyan boxes are ground truths and red circles are estimated vanishing points. The pink triangles in the second column are the detection results of [18] with vanishing point tracking algorithm enabled. Odd rows: Locations of vanishing points; even rows: accumulator spaces. (a)–(d) The locations of estimated vanishing points using our method, and the methods in [18]–[20], respectively.

candidates on unstructured roads with week (a), (b) or strong texture (c), (d), and paved roads with noticeable (e) or sparser lane markers (f). It is clearly that the texture of Fig. 6(a) and (b) becomes more noticeable from the left to the right, while Fig. 6(f) is the opposite. Thus the points in the last column of Fig. 6(a) and (b) are more compact. Comparing with Fig. 6(a) and (b), the road images [Fig. 6(c) and (d)] whose texture are stronger tend to have a sharper vote function. In addition, if the displacement $|\overline{\mathbf{z}}_{k-1} - \overline{\mathbf{z}}_{k-2}|$ is large, the distribution of vanishing point candidates will be expanded, thus the actual vanishing point would fall in this range (the second image of Fig. 6(e)). As a result, it can track rapid changes or a wrongly estimated vanishing point can be recovered quickly. As shown, the number of vanishing point candidates to be voted changes with their distribution. The more compact the candidates are, the greater the chances of repetition will be. Fig. 7 illustrates

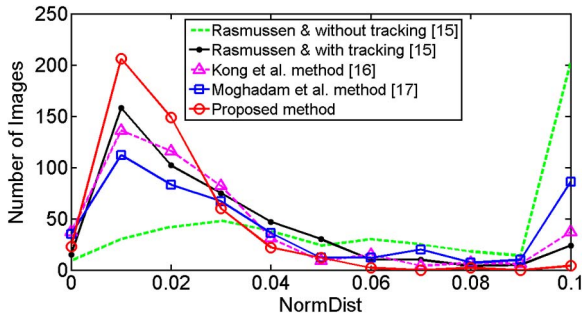


Fig. 8. Normalized Euclidean distance error comparison of vanishing-point detection methods in an 11-bins histogram.

TABLE I
ACCURACY COMPARISON

Method	Rasmussen [18]	Kong [19]	Moghadam [20]	Proposed method
NormDist Error	0.0291 (0.0885*)	0.0316	0.0424	0.0189

* Detected result based on single image

the detection results with estimated vanishing points overlaid on the original images. It is clear that the algorithm works perfectly well for both structured and ill-structured road conditions. For better comparison and in-depth analysis, the estimation errors are measured by comparing the results of the four algorithms against the ground-truth using the normalized Euclidean distance [20]: NormDist $= ||P(x_p, y_p) - P_0(x_0, y_0)||/L$, where $P(x_p, y_p)$ and $P_0(x_0, y_0)$ are the locations of estimated and ground-truth vanishing points respectively. A value toward 0 means the increase of estimation precision. We evaluated the results in an 11-bin statistical histogram. The normalized Euclidean distance that is larger than 0.1 is placed into the last bin of the histogram shown in Fig. 8. Table I shows the average NormDist error of the algorithms. The number 0.0885 inside the parentheses refers to the NormDist error of the vanishing point detection results in [18] based on a single image. As Fig. 8 indicates, our method has largest number of images in the left part of the histogram and thus achieves highest detection accuracy. In particular, the method [18] has 25 (with tracking) and 202 (without tracking) images with errors larger than 0.1. The methods [19] and [20] show 37 and 86 images with large error, whereas in our method, there are only less than 4 images. Besides, the NormDist error that is less than 0.01 occurs in 173 and 147 images obtained using the methods [19] and [20], respectively. But that number decreases to 39 images in [18] (without tracking). Using vanishing point tracking technique, the performance of [18] is greatly enhanced, with 158 images having NormDist error less than 0.01, whereas there are 229 images in our method. It is obvious that our method outperforms the others significantly.

B. Efficiency Comparison

Table II lists the average elapsed time of the four algorithms on an image. As indicated, the Rasmussen [18] (without tracking) and Moghadam [20] methods are the slowest, with about 7.14 sec and 7.39 sec respectively, The method [19] takes 2.14 sec on average. By using vanishing point tracking technique, the running speed of [18] is improved, which takes 1.73 sec on average. However, our method only consumes 0.027 sec. The reason why our method outperforms [18] is mainly two aspects: 1) our texture orientation estimation algorithm is much efficient; and 2) our method needs less extra cost than [18] for tracking the vanishing points. We want to further point out that Kong *et al.* [28] also have implemented their novel gLoG filter based vanishing point detection algorithm in MATLAB. The average time cost on a 240 × 180 image is 55 sec for their slow gLoG method and 0.98 sec for their efficient gLoG method. Rasmussen *et al.* [2] have implemented their algorithm in Graphics hardware and it runs at roughly 30+ fps on a 60 × 80 image. The proposed texture orientation estimation and sparse voting algorithms are parallelizable and can be further speed up by parallel computing devices.

TABLE II
EFFICIENCY COMPARISON

Method	Rasmussen [18]	Kong [19]	Moghadam [20]	Proposed method
Time Ellasped (s)	1.73 (7.14*)	2.14	7.39	0.027

* Detected result based on single image

C. Failure Case Study and Future Work

There are a number of situations that can cause our algorithm to fail. We have collected a large amount of videos from Internet to list as many as possible the failure cases. Based on our observation, these situations can be classified into three general categories:

1) Poor Road Conditions: We found that the estimator shows jumping vanishing points or even fails when the vehicle runs on bumpy roads or goes up or down an abrupt slope [Fig. 9(a)]. There are two reasons for the fails on these situations. First, if the peakedness of VAS is high, the algorithm may predict that the vanishing point resides in a small region in the road image plane, while the actual one might not be inside because of an abrupt change. However, once the peakedness measure becomes low, the search range would be enlarged exponentially, thereby, the actual vanishing point can be reacquired quickly. In fact, our method can have better results if an ego-motion compensation is available. Second, in this paper, we fix the value of $l\,(= 20)$ to smooth the measurements for all of the test images to reduce the stochastic observation noise influence. This assumes that the vehicle is running at constant speed and therefore there is a lag or advance in the tracking. By dynamically adjusting l according to the velocity of the ego vehicle, it is expected that our results can be further improved.

2) No-Road Images: The second type of failures occur when the vehicle approached road dead-ends, T-intersection [Fig. 9(b)], or left/right-angled turn [Fig. 9(c)], or the road continuation is outside the camera field of view [Fig. 9(d)] etc. An approach to sense when the estimator has failed is to threshold the VAS peakedness and then switch to other detection modules if available. In addition, the vanishing point search space is restricted due to the fixed camera mounting. To keep the road in a given look ahead distance within the visible

Fig. 9. Illustrations of how the vanishing point fails under poor road conditions (a), no-road situations (b)–(d), or when there are strong shadows or sun glare in the road images (e), (f).

area, a simple method is using gaze control to direct the camera toward the road vanishing point direction [36].

3) Shadows or Sun Glare: Other failure cases are the situations when shadows are cast nearly along the road [Fig. 9(e)] or sun glare causes many road pixels on the camera to saturate [Fig. 9(f)]. If the shadow edges are strong enough, and the road vanishing point is close enough to the phantom vanishing point, the tracker may be distracted. Fig. 9(e) shows how a vanishing point "hop" from the actual to a phantom position. There are many approaches can be used to mitigate this problem. For example, such shadows can be explicitly detected and removed using Yang's algorithm [37] as a pre-processing step before the dominant orientation computation. Another possible approach for the former case is to predict the shadow region in the image by calculating the sun altitude and azimuth as the method presented in [2].

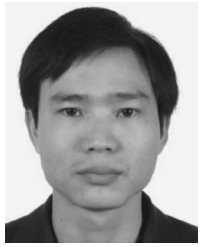
V. Conclusion

We have presented a novel algorithm that focuses on improving vanishing point tracking accuracy and robustness. The algorithm uses joint activities of four Gabor filters and confidence measure to speed up dominant orientation computation. After the dominant orientations have been estimated, a local soft voting scheme is utilized to locate the vanishing point. The vanishing point is traced from frame to frame in order to reduce the computational complexity of voting and the chances of misidentification due to a false peak elsewhere in the image. The distribution of vanishing point candidates is tuned based on the peakedness measure and the displacement of moving average of vanishing point observations. The proposed algorithm works for both structured and ill-structured road conditions. The method has been implemented and tested over 20000 general road video frames including desert, rural, snow covered and well painted roads with varieties of colors and varying illuminations. Evaluation made on 480 frames demonstrated that it outperforms some state-of-the-art texture-based algorithms in terms of detection accuracy and speed. Despite of this, there are still many situations that the algorithm may fail, which necessitate modifications in the future work to make it more reliable and effective. Firstly, improvement can be made by incorporating car-road geometry and odometry information into the particle filtering. Secondly, the failures in non-road images could be prevented by using an active-pointed camera. To harden the algorithm against strong shadows or sun glare, possible solutions include shadow region prediction by sun altitude and azimuth calculation and explicit shadow removal before the dominant orientation computation. Finally, by going to a higher resolution for vanishing point voting, additional precision can be achieved.

References

[1] M. Bertozzi *et al.*, "Artificial vision in road vehicles," *Proc. IEEE*, vol. 90, no. 7, pp. 1258–1271, Jul. 2002.
[2] C. Rasmussen, "RoadCompass: Following rural roads with vision+ladar using vanishing point tracking," *Auton. Robot.*, vol. 25, no. 3, pp. 205–229, Oct. 2008.
[3] H. Kong, J. Audibert, and J. Ponce, "Vanishing point detection for road detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 96–103.
[4] H. J. Liu, Z. B. Gao, J. F. Lu, and J. Y. Yang, "A fast method for vanishing point estimation and tracking and its application in road images," in *Proc. IEEE 6th Int. Conf. Telecommun.*, 2006, pp. 106–109.
[5] J. M. A. Alvarez and A. M. Lopez, "Road detection based on illuminant invariance," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 1, pp. 184–193, Mar. 2011.
[6] Y. Alon, A. Ferencz, and A. Shashua, "Off-road path following using region classification and geometric projection constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2006, vol. 1, pp. 689–696.
[7] H. Y. Cheng *et al.*, "Environment classification and hierarchical lane detection for structured and unstructured roads," *IET Comput. Vis.*, vol. 4, no. 1, pp. 37–49, Mar. 2010.
[8] Y. Wang, D. Shen, and E. K. Teoh, "Lane detection using spline model," *Pattern Recognit. Lett.*, vol. 21, no. 8, pp. 677–689, 2000.
[9] P. Wu, C. Chang, and C. Lin, "Lane-mark extraction for automobiles under complex conditions," *Pattern Recognit.*, vol. 47, no. 8, pp. 2756–2767, 2014.
[10] Y. Kuo, N. Pai, and Y. Li, "Vision-based vehicle detection for a driver assistance system," *Comput. Math. Appl.*, vol. 61, no. 8, pp. 2096–2100, 2011.
[11] G. Chen *et al.*, "Lane detection based on improved canny detector and least square fitting," *Adv. Mater. Res.*, vol. 765, pp. 2383–2387, Sep. 2013.
[12] Y. Wang, E. K. Teoh, and D. Shen, "Lane detection and tracking using b-snake," *Imag. Vis. Comput.*, vol. 22, no. 4, pp. 269–280, Apr. 2004.
[13] A. Lookingbill, J. Rogers, D. Lieb, J. Curry, and S. Thrun, "Reverse optical flow for self-supervised adaptive autonomous robot navigation," *Int. J. Comput. Vis.*, vol. 74, no. 3, pp. 287–302, 2007.
[14] J. Alvarez, Y. LeCun, and A. Lopez, "Road scene segmentation from a single image," in *Computer Vision-ECCV*, Berlin, Germany: Springer-Verlag, 2012, pp. 376–389.
[15] T. Li, C. Xu, and Y. Cai, "A fast and robust heuristic road detection algorithm," *J. Inf. Technol*, vol. 13, no. 8, pp. 1555–1560, 2014.
[16] J. Huang, B. Kong, B. Li, and F. Zheng, "A new method of unstructured road detection based on HSV color space and road features," in *Proc. Int. Conf. Inf. Acquis.*, 2007, pp. 596–601.
[17] R. Kumar, C. Jada, M. Feroz, V. A. Kumar, and H. Yenala, "ROBOG an autonomously navigating vehicle based on road detection for unstructured road," in *Proc. IEEE Int. Conf. Signal Process. Commun. Eng. Syst.*, 2015, pp. 29–33.
[18] C. Rasmussen, "Grouping dominant orientations for ill-structured road following," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2004, pp. 470–477.

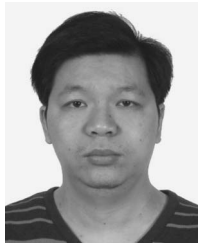
[19] H. Kong, J. Y. Audibert, and J. Ponce, "General road detection from a single image," *IEEE Trans. Image Process.*, vol. 19, no. 8, pp. 2211–2220, Aug. 2010.
[20] P. Moghadam, J. A. Starzyk, and W. S. Wijesoma, "Fast vanishing-point detection in unstructured environments," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 425–430, Jan. 2012.
[21] X. Lu, "New efficient vanishing point detection from a single road image based on intrinsic line orientation and color texture properties," *Optical Eng.*, vol. 51, no. 3, pp. 1–15, Mar. 2012.
[22] M. Nieto and L. Salgado, "Real-time vanishing point estimation in road sequences using adaptive steerable filter banks," in *Advanced Concepts for Intelligent Vision Systems*. Berlin, Germany: Springer-Verlag, 2007, pp. 840–848.
[23] H. Wang, F. Li, and M. Ren, "Vanishing point detection based on infrared road images for night vision navigation," in *Advanced Concepts for Intelligent Vision Systems*. Berlin, Germany: Springer-Verlag, 2013, pp. 611–617.
[24] O. Miksik, "Rapid vanishing point estimation for general road detection," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 4844–4849.
[25] Y. Sun, H. Lu, and Z. Zhang, "Road type estimation and hierarchical real-time vanishing point detection," in *Proc. IEEE 7th Int. Conf. Image Graph.*, 2013, pp. 332–337.
[26] R. Ebrahimpour, R. Rasoolinezhad, and Z. Hajiabolhasani, and M. Ebrahimi, "Vanishing point detection in corridors: Using Hough transform and K-means clustering," *IET Comput. Vis.*, vol. 6, no. 1, pp. 40–51, Jan. 2012.
[27] Q. Wu, W. Zhang, and B. V. Kumar, "Example-based clear path detection assisted by vanishing point estimation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1615–1620.
[28] H. Kong, S. E. Sarma, and F. Tang, "Generalizing Laplacian of Gaussian filters for vanishing-point detection," *IEEE Trans. Intell. Trans. Syst.*, vol. 14, no. 1, pp. 408–418, Mar. 2013.
[29] T. Suttorp and T. Bucher, "Robust vanishing point estimation for driver assistance," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2006, pp. 1550–1555.
[30] P. Moghadam and J. Dong, "Road direction detection based on vanishing-point tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2012, pp. 1553–1560.
[31] Y. Wang, L. Bai, and M. Fairhurst, "Robust road modeling and tracking using condensation," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 4, pp. 570–579, Dec. 2008.
[32] T. Lee, "Image representation using 2D Gabor wavelets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 10, pp. 959–971, Oct. 1996.
[33] T. H. Bui and E. Nobuyama, "A local soft voting method for texture-based vanishing point detection from unstructured road images," in *Proc. IEEE SICE Annu. Conf.*, 2012, pp. 396–401.
[34] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.
[35] P. Pan and D. Schonfeld, "Video tracking based on sequential particle filtering on graphs," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1641–1651, Jun. 2011.
[36] M. Manz, F. von Hundelshausen, and H. J. Wuensche, "A hybrid estimation approach for autonomous dirt road following using multiple clothoid segments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 2410–2415.
[37] Q. Yang, K. Tan, and N. Ahuja, "Shadow removal using bilateral filtering," *IEEE Trans. Image Process.*, vol. 21, no. 10, pp. 4361–4368, Oct. 2012.



Jinjin Shi received the B.S. and M.S. degrees in mechatronic engineering from Fuzhou University, Fuzhou, China, in 2009 and 2012, respectively. He is currently working toward the Ph.D. degree with the Department of Microelectronics, Harbin Institute of Technology, Harbin, China. His current research interests are in robotics, image processing and its applications in obstacle detection, automated highway, and driving assistance systems.

Jinxiang Wang received the B.S. and M.S. degrees in semiconductor physics and the Ph.D. degree in communication and information engineering from Harbin Institute of Technology, Harbin, China, in 1990, 1993, and 1999, respectively.

He is currently a Professor with the Department of Microelectronics, Harbin Institute of Technology. His research interests are very large-scale integration design, wireless communication, signal processing, and computer vision.



Fangfa Fu received the M.S. and Ph.D. degrees in electronic science and technology from Harbin Institute of Technology, Harbin, China, in 2008 and 2012, respectively.

Since 2012, he has been giving lectures with the Department of Microelectronics, Harbin Institute of Technology. His research interests include the analysis and interpretation of images and videos from onboard cameras, very large-scale integration design, and digital signal processing.