New efficient vanishing point detection from a single road image based on intrinsic line orientation and color texture properties

Xiqun Lu

New efficient vanishing point detection from a single road image based on intrinsic line orientation and color texture properties

Xiqun Lu
Zhejiang University
College of Computer Science
Yu Quan Campus, Hangzhou 310027, China
E-mail: xqlu@zju.edu.cn

Abstract. Detecting the vanishing point from a single road image is a challenging problem because there is very limited information in the input image that can help the computer to deduce the genuine location of vanishing point. Besides, the cluttered ambient environment in a real road image sometimes will hinder rather than assist the detection. Learning both the advantages and the limitations of current edge-based and texture-based approaches motivates us to propose a new vanishing point detection method that exploits the intrinsic geometric line structures and color texture properties of general roads. Our approach integrates the efficiency of line segments of edge-based methods, and the orientation coherence concept that is frequently applied in texture-based methods, which can be of great help to improve the accuracy of selecting the right line segments for vanishing point detection. The proposed method has been implemented and tested on over 1000 various road images. These road images exhibit large variations in color, texture, illumination condition, and ambient environment. The experimental results demonstrate that this new method is both efficient and effective in detecting vanishing point when compared to the state-of-the-art edge-based and texture-based methods. © *2012 Society of Photo-Optical Instrumentation Engineers (SPIE).* [DOI: 10.1117/1.OE.51.3.037001]

Subject terms: road image; vanishing point; detection; orientation coherence; color textures.

Paper 111348 received Oct. 27, 2011; revised manuscript received Jan. 3, 2012; accepted for publication Jan. 26, 2012; published online Mar. 23, 2012.
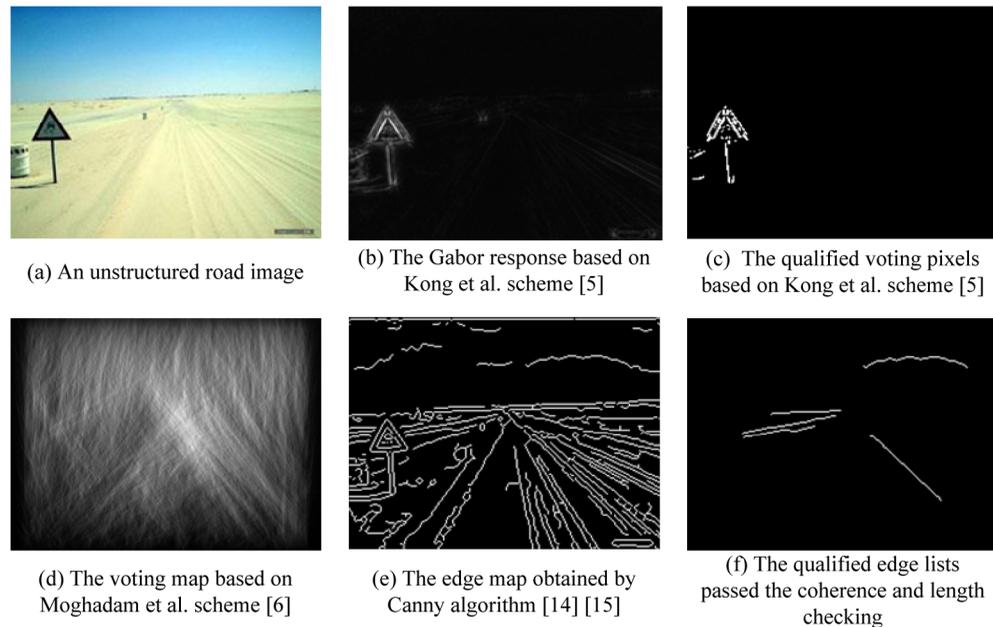
1 Introduction

Over the past few decades, there have been numerous research works about developing autonomous vehicle navigation systems in either structured,[1,2] urban environments[3] or unstructured roads.[4–7] 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 more likely 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 genuine location of vanishing point.

The majority of vision-based vanishing point detection methods in the literature can be grouped into three main categories: edge-based methods,[1,2,8] texture-based methods,[4,6,7] and prior-based methods.[9,10] Edge-based methods, e.g., in Ref. 1, edge pixels are extracted by the Canny detector,[11] and then the straight lines can be detected by the Hough transform, finally the intersections of any pair 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 are appropriate for structured roads with well-painted parallel road lanes, while they are sensitive to spurious straight lines caused by the cluttered ambient objects except the road lanes in the scene, and may not perform well in unstructured roads without strong edges or contrasting local characteristics. Textured-based methods, on the other hand, search for local oriented textures and then make them vote for the location of the road's vanishing point.[4,6,7] The local soft voting scheme proposed in Ref. 6, as well as the global voting schemes proposed in Refs. 4 and 7, are time-consuming and cannot meet the requirements of real-time applications. At the same time, these approaches are sensitive to noise: if a road scene exists some obstacles with strong edges than the tracks left by previous passed vehicles, then these strong boundaries will induce the voter to an incorrect estimation of vanishing point (see the image in Fig. 1, and the images in the 1st and 2nd rows of Fig. 2).

In order to overcome the limitations of these low-level feature-based detection methods, prior-based techniques have been proposed recently. In Ref. 9, Alvarez et al. suggested integrating contextual three-dimensional information with low-level features to improve the detection performance. Such weak contextual cues include three-dimensional scene layout, three-dimensional road stages and temporal road cues, and so on. In Ref. 10, Wu et al. studied a global perspective structure matching (GPSM) scheme based on an image retrieval technique to identify the best candidate images in an image database, then used the prelabeled vanishing points of the best candidates as the initial estimation of the input image's vanishing point, and finally used a probabilistic model of vanishing point to refine the location of vanishing point. For these machine learning—based methods, not only is a large-scale image or video training database necessary for making these

(a) An unstructured road image

(b) The Gabor response based on Kong et al. scheme [5]

(c) The qualified voting pixels based on Kong et al. scheme [5]

(d) The voting map based on Moghadam et al. scheme [6]

(e) The edge map obtained by Canny algorithm [14] [15]

(f) The qualified edge lists passed the coherence and length checking

Fig. 1 An example to illustrate the problems with current texture-based methods.

prior-based methods robust to various imaging conditions, road types, and scenarios, but the training algorithm is also very important, 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.

Learning both the advantages and the limitations of current edge-based and texture-based approaches motivates us to propose a new efficient vanishing point detection method in this paper that takes advantage of the intrinsic line orientation and color texture properties of roads. This method is implemented and tested on over 1000 various road images. 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 or weather conditions. The proposed method provides higher accuracy of vanishing point detection when compared to some state-of-the-art edge-based and textured-based methods.[1,6,7] This paper is arranged as follows. Section 2 will discuss some related vanishing point detection algorithms, and some interesting works of line-based geometric analysis of a three-dimensional human-made scene based on a single view image. In Sec. 3 we introduce the new efficient vanishing point detection method, which integrates the efficiency of line segments of edge-based methods and the orientation coherence concept that is frequently applied in texture-based methods which can be of great help to improve the accuracy of selecting the right line segments for vanishing point detection. In Sec. 4, we evaluate the performance of the proposed vanishing point detection algorithm quantitatively and qualitatively. Finally, some conclusions are drawn in Sec. 5.

2 Related Works

Under perspective projection, parallel lines in three-dimensional space project to converging lines in the image plane, and the common point of intersection is called the vanishing point. The vanishing point analysis can provide strong cues for inferring the three-dimensional structure of a scene from only a single view.[12]

A general human-made scene may have two or more vanishing points, which correspond to different sets of parallel lines of regular geometric structures in the scene such as windows, walls, and buildings. Recently two new approaches have been proposed for detecting vanishing points in human-made environments: one is based on the J-Linkage (the simultaneous estimation of multiple models),[8] and the other is based on a property of one-dimensional affine-similarity between parallel crosssections of a concurrent line set.[13]

However, these approaches have difficulties in handling complex real-road scenes, where the clutter ambient environment will induce many spurious lines that are prone to deduce the incorrect vanishing point, and at the same time it is difficult for one to identify self-similar structures or repeating patterns in these real-road images.

We all believe that the vanishing point plays an important role as a global constraint for detecting road direction, since all parallel road lanes on structured roads, or ruts and tracks left by previous passed vehicles on unstructured roads, appear to converge into a single vanishing point.

Most of the edge-based vanishing point detection algorithms rely on three steps.[1,2] First, edge pixels are extracted by an edge detector. Usually the Canny detector is employed to obtain the edge map.[11] Then the straight lines can be determined by a Hough transform. Once all of the line segments are identified, a voting or a weighted voting procedure is applied to find the vanishing point based on the intersections of lines. Although these methods are simple and efficient, a big shortcoming of these methods is that they are very sensitive to spurious lines caused by the cluttered ambient objects except the road lanes. In order to improve the stability and the accuracy of vanishing point estimation, Suttorp and Bücher brought some well-known physical world constraints into the estimation.[2] For example, the orientation of line segments must not be horizontal or approximated
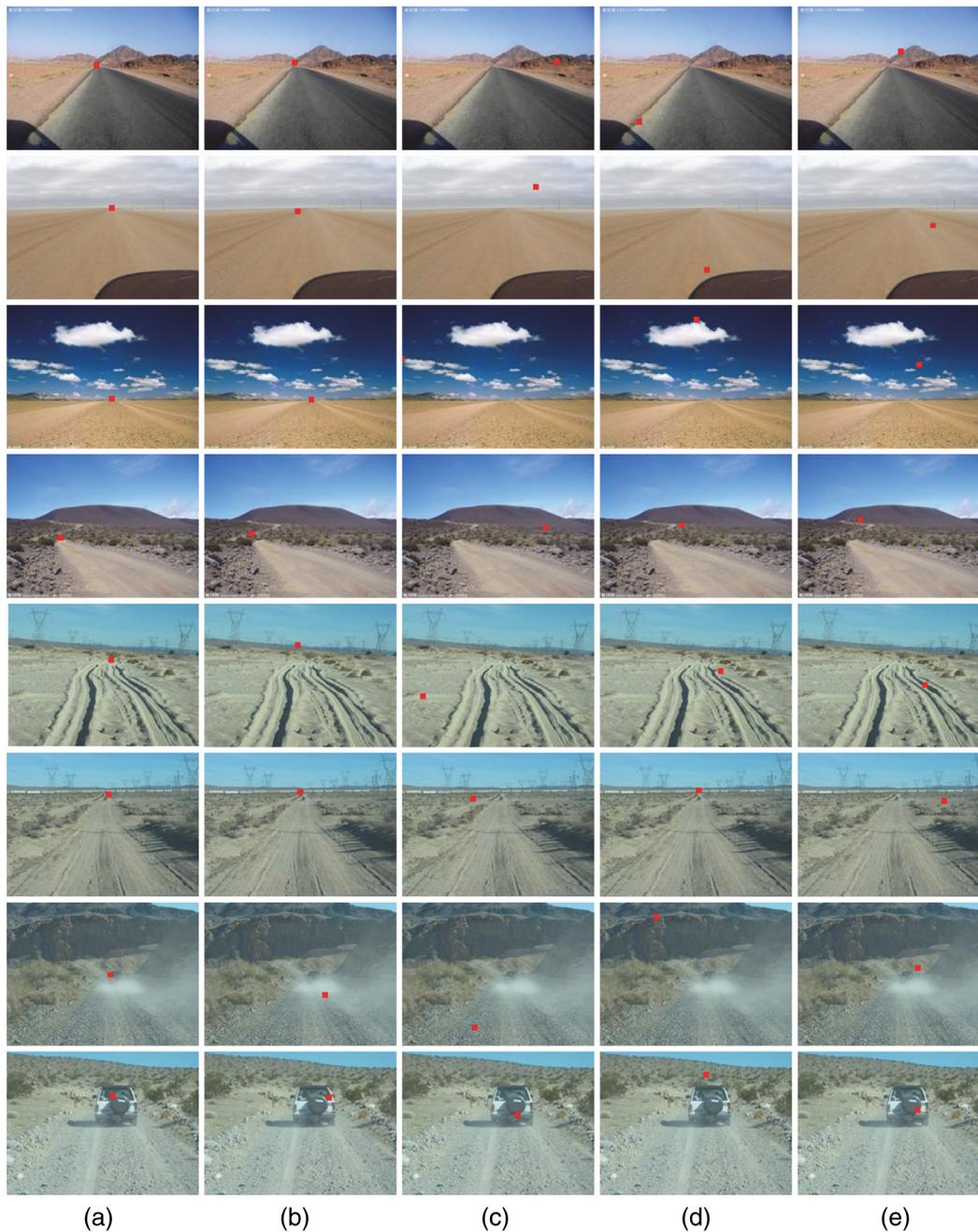
Fig. 2 Examples of vanishing point detection for unstructured roads. (a) The ground truth locations of vanishing points, (b) the estimated vanishing point locations of our proposed method, and (c) the estimated locations of the Hough transform-based method.[1] (d) and (e) The vanishing point locations estimated by Kong et al.[6] and by Moghadam et al.,[7] respectively.

horizontal, but the horizontal line which corresponds to the vanishing line of a three-dimensional scene in the projected image plane sometimes can help the computer to locate the vanishing point. They also assume that the displacement between the vanishing points in two consecutive frames should be small (because of Newton's Law of Mechanics), so a Kalman filter can be used to refine the vanishing point position achieved by a data-driven process. However, in this paper we mainly focus on the detection of vanishing point based on a single road image.

In order to address the problems introduced by unstructured roads, such as the fact that they have no apparent boundaries, Rasmussen proposed texture-based methods for vanishing point detection.[4] Texture-based approaches apply a bank of directional filters such as Gabor filter banks,[14] and choose the orientation with the maximum filter response as the dominant texture orientation $\hat{\theta}(\mathbf{p})$ at a pixel location $\mathbf{p}(x, y)$. Only those with higher confidence pixels (based on Gabor response magnitudes) are qualified for voting the vanishing point, and a location with maximum votes is considered as the vanishing point of the road. In order to achieve precise orientation estimation, one needs to apply a large number of oriented filters in all possible directions from 0 to 180 deg, and this is computationally expensive.

Moghadam et al. proposed a single scale 4-orientation Gabor filter bank with orientations {0, 45, 90, 135 deg} to speed up the texture orientation estimation process.[7] Actually using the fast Fourier transform can accelerate the computation of Gabor responses, therefore there is not too much difference between whether you use a 5-scale, 36-orientation Gabor filter bank[6] or just a single scale, 4-orientation Gabor filter bank with orientations {0, 45, 90, 135 deg}.[7] In the experiments, we find that the voting scheme plays a much more important role in the vanishing point detection than does the selection of directional filter banks. In the local soft voting scheme proposed by Kong et al.,[6] only those pixels with the confidence values higher than $0.3[\max_{\mathbf{p}} \text{Conf}(\mathbf{p}) - \min_{\mathbf{p}} \text{Conf}(\mathbf{p})]$ are qualified for voting. When there are some objects in the scene with strong boundaries compared with the ambient environment (a good example is the image shown in Fig. 1), the Gabor response of the index board is apparently much stronger than those of vehicle tracks. Hence the local soft voting scheme will prefer the pixels above the index board over other positions in the image, and will lead to an incorrect estimation of the vanishing point. From Fig. 1, we can see that the global weighted voting scheme proposed in Ref. 7 is much better than Kong's local soft voting scheme.[6] Moghadam et al. proposed to assign a weight to each ray $\mathbf{r_p}$ based on its dominant orientation $\sin[\hat{\theta}(\mathbf{p})]$. In order to reduce the bias that the pixels in the upper part of the image will receive more votes than lower image pixels because pixels only vote for pixels above them, they suggested a distance-based voting scheme, and this approach allows each dominant orientation to give higher vote to the points closer to it than to those points far away along its ray $\mathbf{r_p}$.

Many research works have pointed out that for unstructured roads,[4,6,7,10] edge-based vanishing point estimation methods may have some problems with edge detection because there are no apparent boundaries in unstructured road scenes. But for our own research experience, this depends on the selection of derivative operators. In Fig. 1(e), we compute the 1st derivatives of the input image using the 5-tap filters given by Farid and Simoncelli.[15] The Canny detector[11] can extract useful lines related to ruts or tracks for detecting the vanishing point in an unstructured road environment.

When writing this paper, we also notice that there is an increasing interest in the line-based geometric analysis of a human-made scene based on a single view image.[16–19] To reconstruct a three-dimensional scene from a single view is an ill-posed problem, and this problem becomes more tractable under the assumption of "Manhattan-World" that the surfaces of interest are rectangles aligned with a three-dimensional Cartesian frame.[16] Coughlan and Yuille based their estimation on a dense gradient map of the input image,[16] whereas Denis et al. argued that basing estimation on a sparse edge map permits more accurate statistical models and becomes more efficient.[17]

But, as pointed out by Tretyak et al. in Ref. 18, the edge map is always noisy and contaminated with spurious edge pixels not coming from straight lines. In order to address the ambiguity propagation between different vision layers with the step-by-step approach, Tretyak et al. proposed to integrate line detection, vanishing point location, and higher-level geometric estimation into a single optimization framework.[18]

Another very interesting work about generating plausible interpretation of a scene from a collection of line segments extracted from a single indoor image is proposed by Lee et al.[19] They proposed several physical valid structure hypotheses, and tried to find the best-fitting physical hypothesis to line segments.

All these works assume that a calibrated camera or some important parameters about the camera are known. Also, an image that comes from a human-made environment is highly likely to contain a certain number of straight lines, which correspond to different sets of parallels line of regular geometric structures in the scene. The presence of such lines and their parallelism are valuable cues to infer the three-dimensional structure from the input image.

However, most of the road images in our collection are taken in arbitrary attitudes, so it is not valid to assume a calibrated camera or assume that the camera's parameters for all these images are known. Due to the complexity of a real-road scene, there exist many spurious edge points that are not coming from straight lines. Even worse, the line detection step may detect some spurious lines that do not exist in the scene. Thus, an unavoidable problem we have to face is to select the right line segments for vanishing point detection.

In this paper we integrate the efficiency of line segments of edge-based methods and the orientation coherence idea applied widely in texture-based methods into a new efficient vanishing point detection scheme.

3 Framework for Vanishing Point Estimation

In this section we present a new framework for vanishing point detection that utilizes the intrinsic geometric line and color texture properties of roads. Although it is based on line segments, it is very different from common edge-based methods [1,2] that determine the vanishing point by searching a point that is close to most line segments. In a practical road image, the cluttered ambient environment and background (like clouds in the sky and trees on both sides of the road) will introduce many spurious edges [please see Figs. 1(e) and 3(b)] that will lead to an incorrect vanishing point estimation just based on a simple voting scheme; we therefore 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 midline of each intersected line pair as shown in Fig. 3(e) to determine which intersection point should be the vanishing point. Next we will describe each step in detail.

3.1 *Selection of Right Line Segments*

For estimating the vanishing point, only those line segments that correspond to the lane borders and the vanishing line in the world coordinate should be used. However, the cluttered ambient environment and background will induce many spurious edges, which will make edge-based methods prone to an incorrect estimation. The sensitive derivative operators will make things even worse, because they will extract more spurious edges. Therefore, the first problem we should face is the selection of right edge lists that will be used in the subsequent line fitting stage. We use the following criteria to ensure that only proper edge lists are selected:
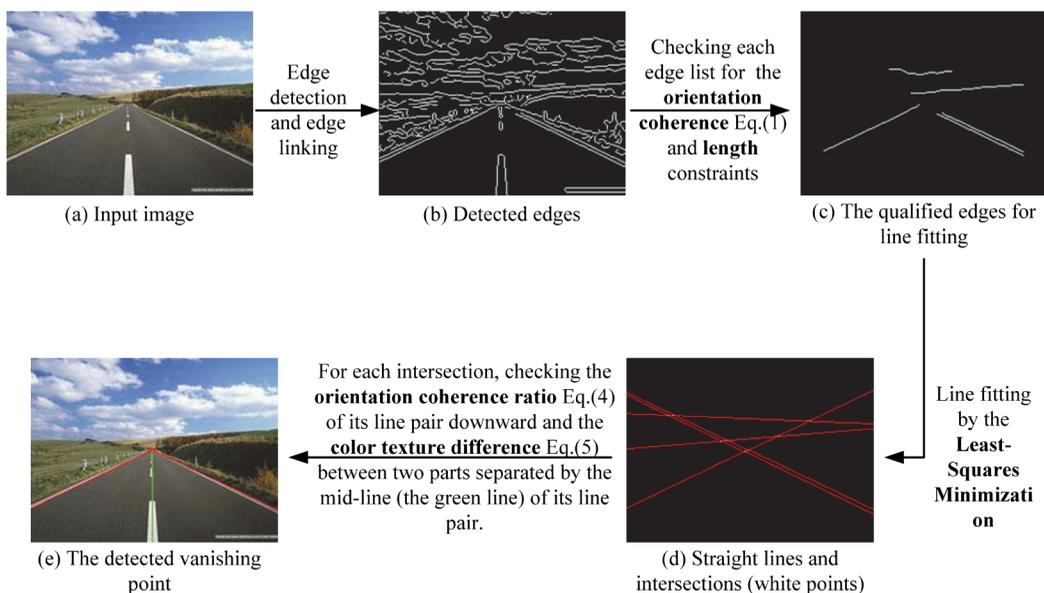
(a) Input image   (b) Detected edges   (c) The qualified edges for line fitting

(e) The detected vanishing point   (d) Straight lines and intersections (white points)

Fig. 3 The framework of vanishing point detection.

The edge lists must be longer than a minimal length. Generally in road images, the lanes of the road tend to be longer than other edges. However, to set a fixed minimal length for all road images is usually difficult because road images are very different. In the experiment, we find that one minimal length that works well for certain road images might not necessarily select enough edge lists for other road images. Hence, in our implementation we set the initial minimal length to be the half of the height of the input image. This minimal length can be reduced gradually to a smaller value when not enough edge lists can be selected under the current length. Each time, 5 pixels are reduced from the current minimal length until enough edge lists are selected or the minimal length reaches to 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 also set as 10. To reduce the influence of the image border on the measurements, in this paper, we assume that the vanishing point is located within the inside region of the input image (from 10% to 90% of the width and 10% to 90% of the height).

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

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

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 on the two points (computed by the 5-tap derivative filters proposed in Ref. 15). Then we select five of the most coherent edge lists which are with the smallest C values of Eq. (1) for the next line fitting stage, and throw away the rest of edge lists. The examples of the qualified coherent and long-enough edge lists are shown in Figs. 1(f) and 3(c).

3.2 Straight Lines Fitting

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

$$\min \left\| \mathbf{A} \mathbf{l}_i \right\|, \qquad (2)$$

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,

$$A = \begin{bmatrix} x_1^i & y_1^i & 1 \\ \vdots & \vdots & \vdots \\ x_{n_i}^i & y_{n_i}^i & 1 \end{bmatrix}. \qquad (3)$$

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

3.3 Determining the Vanishing Point

In Fig. 3(d), there are multiple intersection points (indicated with white color), and 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 determining the vanishing point just by searching a point close to most line segments may lead to an incorrect position.

So here we check an orientation coherent ratio along each line pair downward from their intersection point, respectively:

$$C(\mathbf{l}_i^s) = \frac{\#[|\hat{\theta}(\mathbf{p}_j) - \theta(\mathbf{l}_i)| < T, \mathbf{p}_j \in \mathbf{l}_i^s]}{\text{length}(\mathbf{l}_i^s)}, \qquad (4)$$

where $\mathbf{l}_i^s$ indicates a part of the line $\mathbf{l}_i$ which 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 image, and $\theta(\mathbf{l}_i)$ stands for the orientation of the line $\mathbf{l}_i$; and $\mathbf{p}_j$ denotes one of the points on line $\mathbf{l}_i^s$ which began with the considered intersection point downward, and $\hat{\theta}(\mathbf{p}_j)$ is the estimated orientation of the point $\mathbf{p}_j$ as in Eq. (1); # is the operator to count how many points on $\mathbf{l}_i^s$ have an angle error from the angle of line $\mathbf{l}_i$ that is less than the threshold T (in our implementation T is set as 3 deg). Thus the higher value $C(\mathbf{l}_i^s)$ is, the more coherent the line $\mathbf{l}_i^s$ 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 two parts separated by the mid-line of each intersected line pair as well.

As we know the angles of all these fitted straight lines (we have computed the parameters of each straight line $\mathbf{l}_i = (a_i \, b_i \, c_i)^T$ in the Sec. 3.2), so the angle of mid-line of each intersected line pair can be computed as $(\phi_1 + \phi_2)/2$ (where ϕ_1 and ϕ_2 are the angles of two intersected lines). Also, it is easy to determine which points are located within the area enclosed by two intersected lines as shown in Fig. 4: the angle of the line connected the intersection point and the considered point (which is always below the intersection 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:

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

$$(5)$$

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 of the area enclosed by line $\mathbf{l}_i$ and line $\mathbf{l}_j$ separated by their middle line; # is the operator to count how many points located in Ω_1 and in Ω_2; and $c(\mathbf{p})$ 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 by combining Eqs. (4) and (5) together:

$$\mathrm{cost}(\mathbf{l}_i^s, \mathbf{l}_j^s) = \mathrm{CTdiff}(\mathbf{l}_i^s, \mathbf{l}_j^s) - C(\mathbf{l}_i^s) - C(\mathbf{l}_j^s). \quad (6)$$
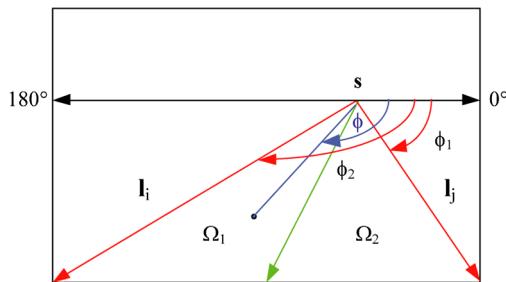
Fig. 4 Example of computing the color texture difference between the two parts enclosed by two intersected lines and their middle line (the two red arrowed lines represent the two lines intersected at **s**, and the green line is their middle line. Note that $\mathbf{l}_i$ and $\mathbf{l}_j$ represent any two intersected lines computed according to the method described in Sec. 3.2 not just the road lanes only, and we draw $\mathbf{l}_i$ and $\mathbf{l}_j$ like road boundaries only for illustration).

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

In this new vanishing point estimation framework, the two-stage orientation coherence checking can help to improve the performance of estimation: checking the orientation coherence and length of edge lists in the first stage can help to remove most of the spurious edge lists that are not related to the lanes of road for further consideration, and can reduce the computation complexity simultaneously, since there will be fewer intersection points left to be considered in the second stage. Unlike traditional edge-based methods,[1,2] we consider the orientation coherence of two intersected lines and the uniformity of the area enclosed by the two lines rather than considering the distance between the intersection point and all line segments, and this will greatly reduce the probability of incorrect estimation.

Although some three-dimensional scene geometric analysis methods are also based on line segments,[16–19] they put their accent on inferring the three-dimensional structure[19] or some high-level vision concepts (the zenith and the horizontal line)[17] of a scene from a single view indoor or city street image. Grouping the different levels of primitives, such as the edge pixels, the line segments, the zenith, and the horizontal vanishing points, into a unified framework may have some advantages over the presented bottom-up pipeline approach. But the energy function based on the likelihoods of edge pixels, line segments, and high vision level constraints is too complicated to be minimized, so they eventually turn to a discrete approximation of the original energy that is easier to be minimized. As done in Ref. 17, two steps of the bottom-up pipeline are performed—namely, the line detection and the vanishing point detection. For the line detection, Denis et al. usually adopt the Hough transform or its probabilistic version,[17] and choose the candidates for vanishing points using the J-Linkage procedure,[8] and then use a simple and brute-force optimization scheme.

There is a big difference between the images comes from indoors or on city streets and the images that come from general outdoor roads: the former are highly likely to contain a certain number of straight lines, which can be easily identified in the edge maps, but for the latter there exist many spurious edge points that are not coming from straight lines. Even worse, the line detection step may detect some spurious lines that do not exist in the scene. Thus, the orientation coherence meters explored in this paper are mainly used for removing spurious line segments for further consideration for vanishing point detection, not as constraints on three-dimensional structure reconstruction of a scene.

4 Experimental Results and Analysis

4.1 Image Database

Vanishing point detection is tested on over 1000 general road images. Most of these images have been used by Kong et al. in Ref. 5; the remainder were downloaded from the Internet by Google Image. In this collection over 550 images are unstructured roads, and about 350 images are structured roads. Some of the images in this collection are shown in Fig. 5. We can see that some of these images feature well-painted roads, and some of them are unstructured roads, like vehicle tracks in
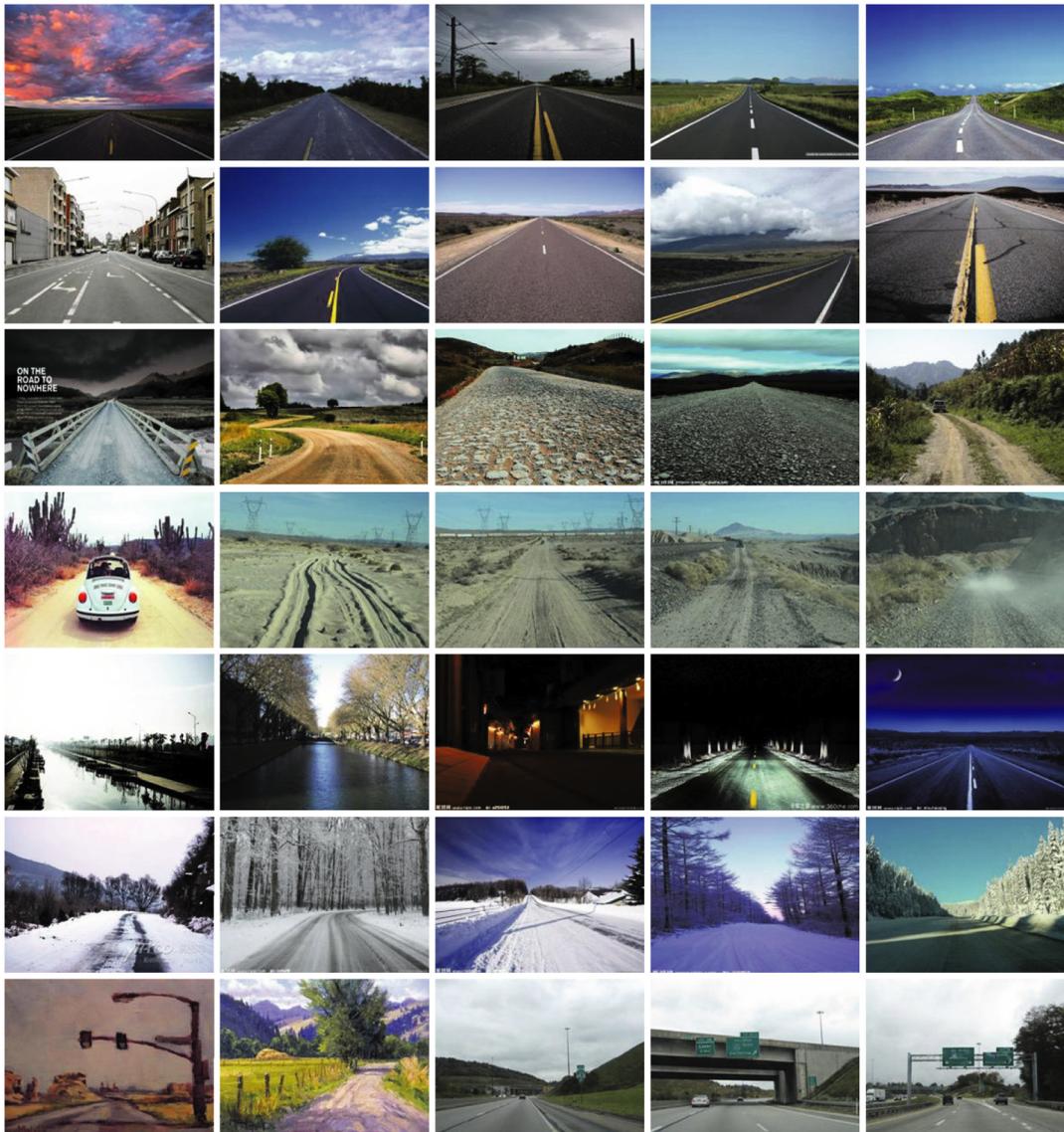
Fig. 5 Different types of roads with varying colors, textures, and ambient environments.

**Table 1** Performance of different vanishing point estimation algorithms.

	Methods			
Road Types	Our proposed method	Classical Hough-based[1]	Kong et al[6]	Moghadam et al.[7]
Total Ave.	**0.15265** (±0.12175)	0.27131 (±0.17535)	0.19374 (±0.12051)	0.15280 (±0.11027)
Structured	**0.13374** (±0.12840)	0.23990 (±0.16642)	0.18913 (±0.12583)	0.17985 (±0.11757)
Unstructured	0.17069 (±0.11378)	0.30242 (±0.18023)	0.19088 (±0.11614)	**0.13107** (±0.10051)
Snow Roads	0.14930 (±0.12451)	0.25811 (±0.16514)	0.21184 (±0.10360)	**0.13930** (±0.10021)
Dark Roads	0.15260 (±0.12884)	0.21847 (±0.14263)	**0.15067** (±0.12127)	0.15321 (±0.11280)
Highway	**0.14104** (±0.14797)	0.21661 (±0.15935)	0.24663 (±0.10162)	0.22603 (±0.09174)
Painting Roads	0.18663 (±0.14434)	0.34874 (±0.18581)	0.25922 (±0.12995)	**0.16608** (±0.11595)
Est. Time (image/second)	~0.5 s	~0.2 s	12–15 s	13–15 s

The numbers in the bold face indicate the best performance (with the minimum normalized Euclidean distance error) among the four algorithms when applied them to different kind of road images.

desert or snow, and some of them are water roads. This road image collection also contains over 10 oil painting road images, and it will be very interesting to test different vanishing point detection algorithms on unnatural road images. From Fig. 5 we can see that the images in this collection exhibit large variations in color, texture, illumination, and ambient environment. In order to analyze the performances of different vanishing point detection algorithms on different types of roads carefully, we divide this image collection into structured roads, unstructured roads, night or dark ambient roads, snow covered roads, oil painting roads, and highways. Although highway roads are well painted, some of them have large interference objects, like road instruction boards, overpasses, or advertisement boards overhead. These are considered independently as a special case of roads.

Since these images are of very different spatial sizes, all images are normalized to the same size (with the height of 180 pixels and the width of 240 pixels) by using the bicubic image interpolation method.[20] To assess the algorithm's performance versus human perception of the vanishing point location, we invited five students in our college to manually mark the vanishing point location in each image in this collection 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.

(a) (b) (c) (d) (e)
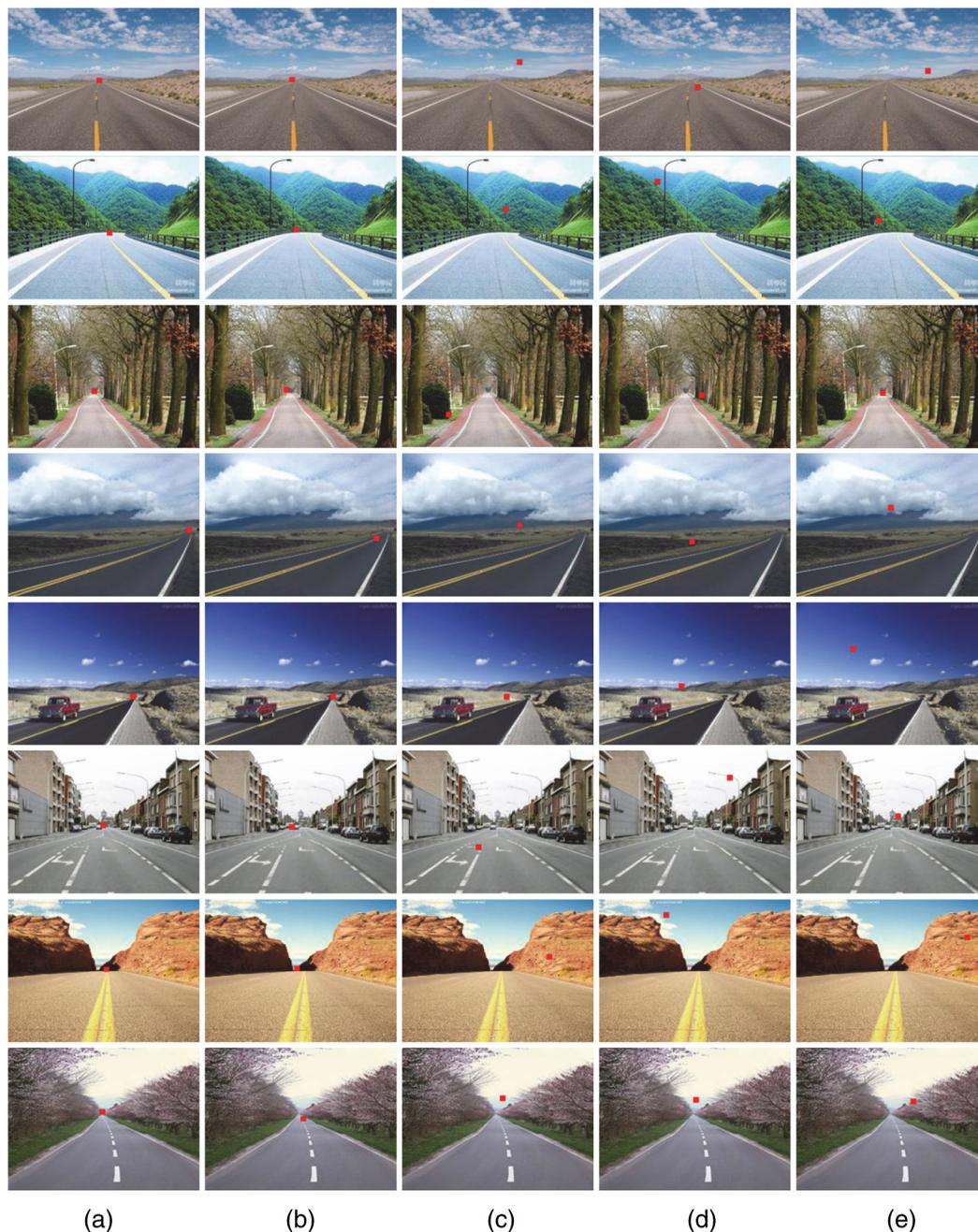
Fig. 6 Examples of vanishing point detection for structured roads. (a) The ground truth locations of vanishing points, (b) the estimated vanishing point locations of our proposed method, and (c) the estimated locations of the Hough transform-based method.[1] (d) and (e) The vanishing point locations estimated by Kong et al.[6] and by Moghadam et al.,[7] respectively.

4.2 Performance Metric

To measure the accuracy of vanishing point estimation method, we use the normalized Euclidean distance as suggested in Ref. 7, where the Euclidean distance between the estimated vanishing point and the ground truth is normalized by the diagonal length of the input image as follows:

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

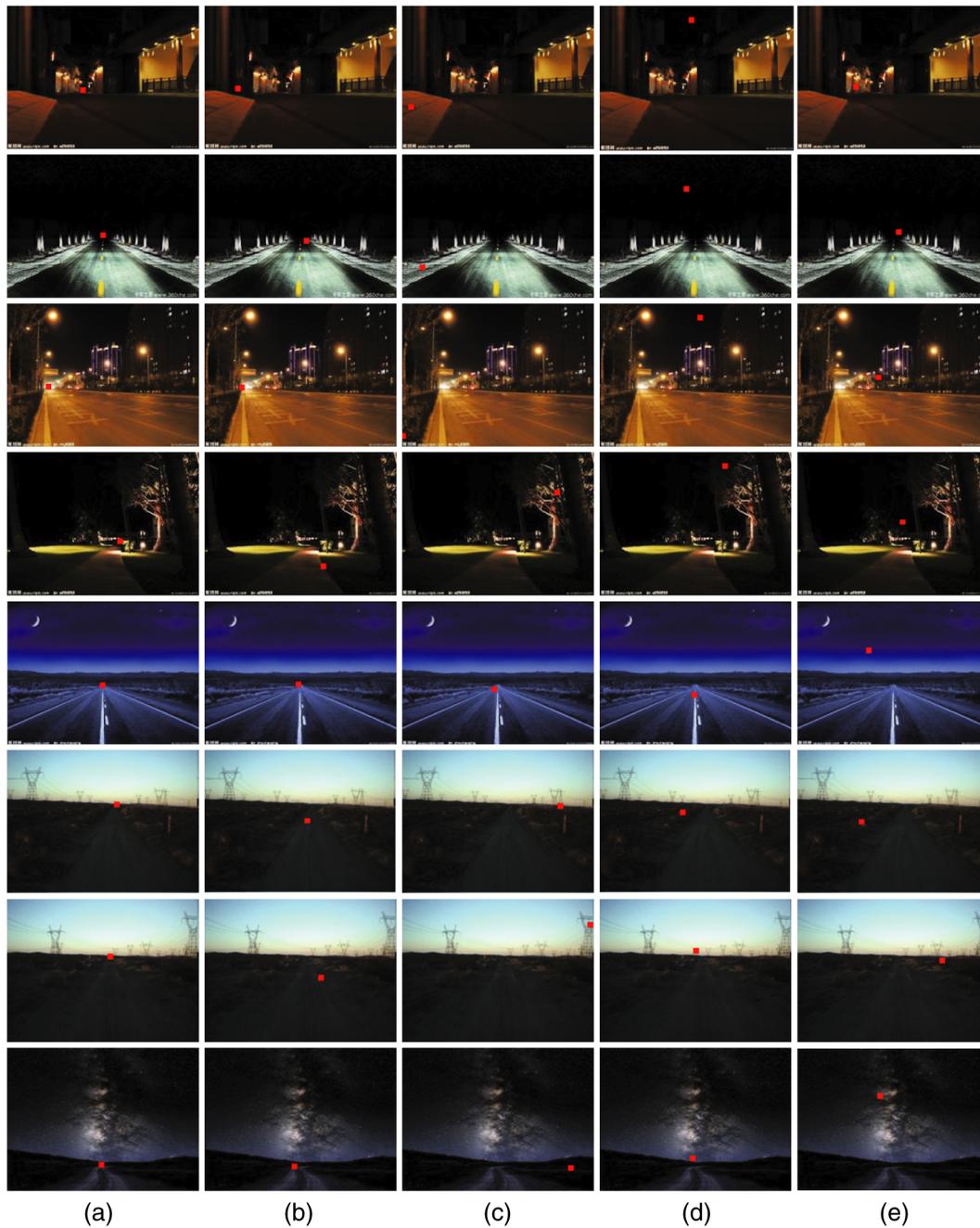where $v_E(x_e, y_e)$ is the estimated vanishing point location and $v_T(x_t, y_t)$ is the location of ground truth vanishing point, and "Diag image" is the diagonal length of the input image. Using this meter, a value near 1 would correspond to an incorrectly estimated vanishing point location. A value near 0 means the estimated vanishing point location is close to the location of ground truth.

4.3 Evaluation Results

In this section, we evaluate the performances of four vanishing point detection algorithms quantitatively and qualitatively. One is based on the Hough transform,[1] and another is our proposed method. Both of these methods belong to the edge-based methods. The other two are the texture-based
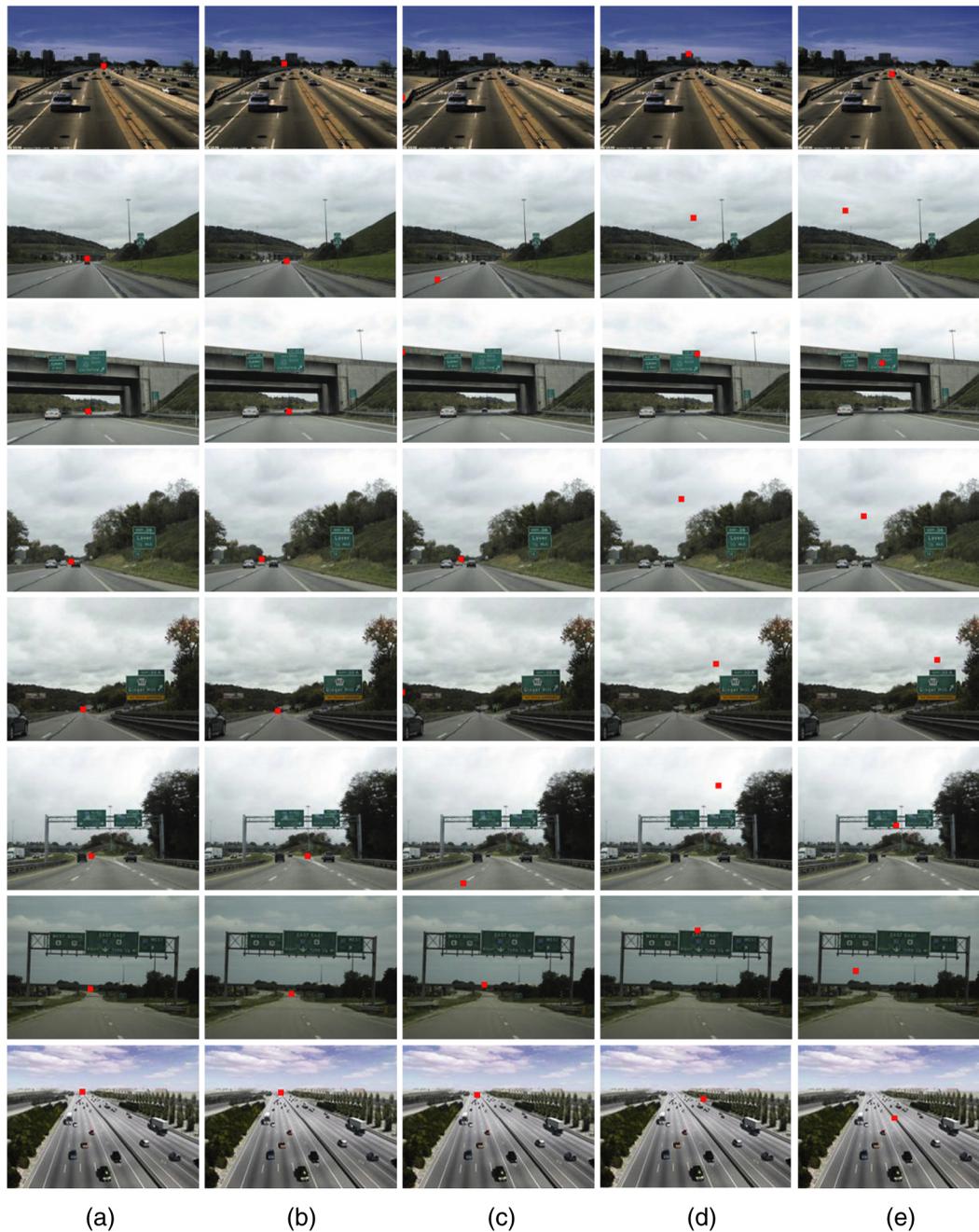

(a) (b) (c) (d) (e)
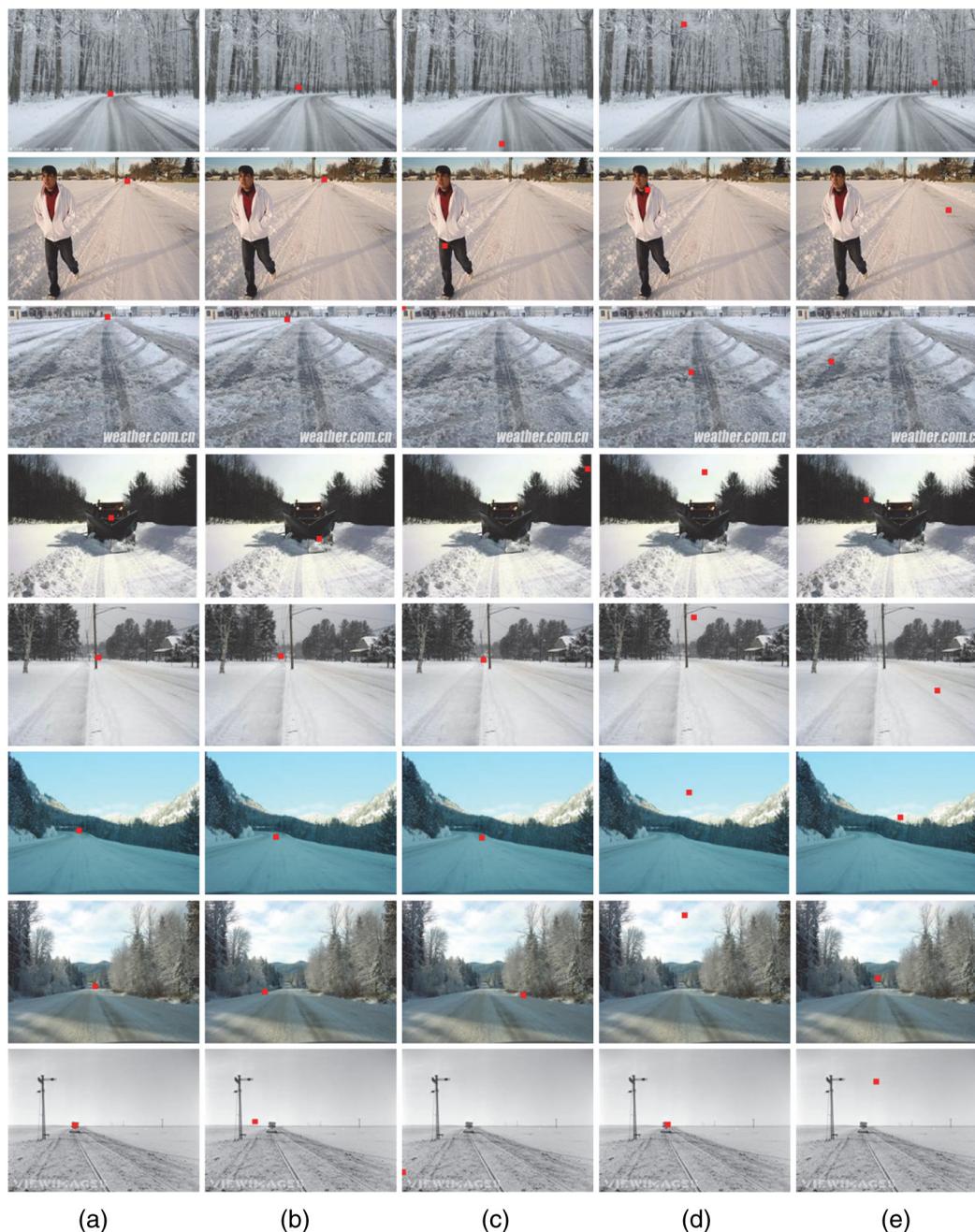
Fig. 7 Examples of vanishing point detection for snow-covered roads. (a) The ground truth locations of vanishing points, (b) estimated vanishing point locations of our proposed method, and (c) the estimated locations of the Hough transform-based method.[1] (d) and (e) The vanishing point locations estimated by Kong et al.[6] and by Moghadam et al.,[7] respectively.

Fig. 8 Examples of vanishing point detection for night or dark ambient roads. (a) The ground truth locations of vanishing points, (b) the estimated vanishing point locations of our proposed method, and (c) the estimated locations of the Hough transform-based method.[1] (d) and (e) The vanishing point locations estimated by Kong et al.[6] and by Moghadam et al.,[7] respectively.

vanishing point detection methods for unstructured roads.[6,7] Table 1 shows the numerical results in terms of the average normalized Euclidean distance error for the test image dataset, and the numbers in the parentheses in Table 1 are the variances of normalized Euclidean distance errors of these four algorithms. Although this image collection contains many more unstructured roads than structured roads, the average normalized Euclidean distance error of our proposed method is still slightly higher than that of the method of Moghadam et al.,[7] and much better than that of Kong et al.[5] and the Hough-based method.[1]

The image dataset is divided into six types of roads (structured, unstructured, snow-covered, night or dark ambient roads, highways, and oil painting roads; please see the results presented in Figs. (2 and 6–10), and the performances of different vanishing point estimation algorithms for different kinds of roads are also outlined in Table 1. As we are only interested in investigating the adaptation of the four vanishing point detection algorithms to different types of roads, we did not tune any parameters when applying them to different types of roads. From Table 1, it can be seen that our method outperforms the other three methods on structured roads and highways, and the method proposed
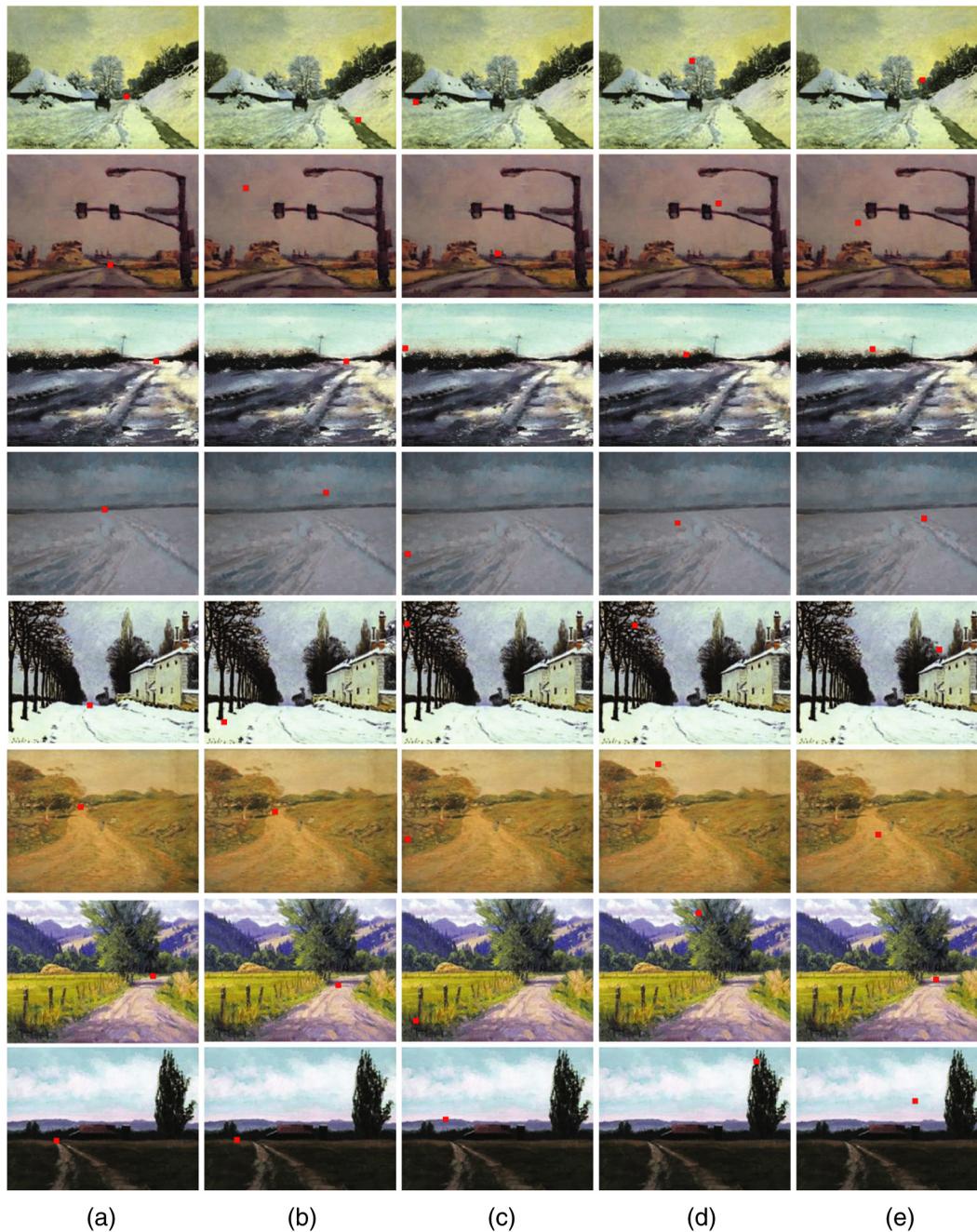
Fig. 9 Examples of vanishing point detection for highways. (a) The ground truth locations of vanishing points, (b) the estimated vanishing point locations of our proposed method, (c) the estimated locations of the Hough transform-based method,[1] and (d) and (e) the vanishing point locations estimated by Kong et al.[6] and by Moghadam et al.,[7] respectively.

by Moghadam et al. is excellent in detecting vanishing points for unstructured roads.[7] Though the snow-covered images are separated as a specific class, most of them are unstructured roads, so it is expected that the method of Moghadam et al. is the best one for snow-covered roads. The stroke textures make the texture-based method proposed by Moghadam et al.[7] more suitable for detecting vanishing points in oil painting road images, but texture-based methods are very sensitive to obstacles with strong boundaries in the image, and this can be seen clearly for detecting the vanishing points for highways in Fig. 9. Since the Hough-based

method[1] is very sensitive to spurious edges, for some images with cluttered ambient environment, the estimated vanishing points will be far from the genuine ones (some the estimated ones will be close to the boundaries or corners; please see examples in Figs. 2 and 7–10).

The last line listed in Table 1 is the estimation time for each input image of each vanishing point detection algorithm. Among the four vanishing point estimation algorithms, the Hough-based method[1] is the fastest, with about 0.2 sec. for each input image; our proposed method ranks second, at about 0.5 sec. for each image. But
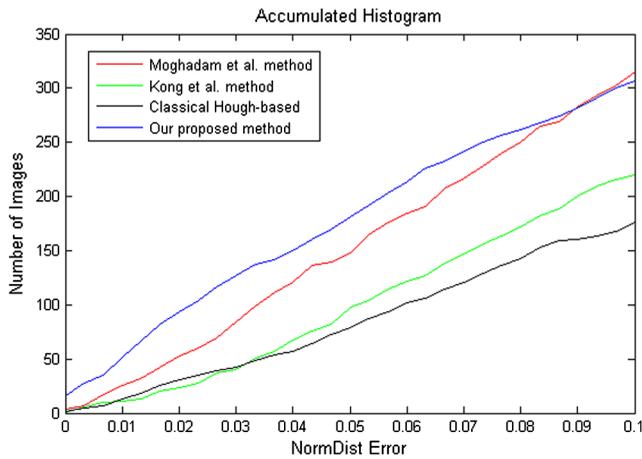
Fig. 10 Examples of vanishing point detection for oil painting roads. (a) The ground truth locations of vanishing points, (b) the estimated vanishing point locations of our proposed method, and (c) the estimated locations of the Hough transform-based method.[1] (d) and (e) The vanishing point locations estimated by Kong et al.[6] and by Moghadam et al.,[7] respectively.

for the texture-based methods,[6,7] the estimation time for vanishing point is around 13 to 15 sec., which depends on the input image because different images have different amount of texture features, and above 95% of the estimation time is spent on the voting stage. The four algorithms' simulations are implemented on Matlab, running on a regular Pentium 3 GHz (2-GB RAM) machine.

For better comparison, we also evaluate the results of our proposed method versus the classical Hough-based method,[1] the method of Kong et al.,[6] and the method of Moghadam et al.[6] in an accumulated histogram. Since the input image is sized 180×240 pixels, the normalized Euclidean distance

of 0.1 in Eq. (7) means that the location of the estimated vanishing point is about 30 pixels away from that of the ground truth, and this is a very large error. Therefore, we only consider that the normalized Euclidean distance is less than 0.1 in an accumulated histogram, as shown in Fig. 11. The accumulated histogram of a perfect vanishing point detection method should like a step function, that is, there is no error between the locations of estimated vanishing points and those of the ground truth ones, so the ideal accumulated histogram line will reach to the total number of image dataset at zero of "NormDist" error, and will keep it for all nonzero "NormDist" errors. Therefore, the higher lines in the

Fig. 11 Comparison of vanishing point detection methods in accumulated histograms.

accumulated histogram represent better results. From Fig. 11, we can see that the proposed method's accumulated histogram line is higher above the other three vanishing point detection methods' lines until the normalized Euclidean distance is close to 0.09; after this point, the method of Moghadam et al. is slightly higher than ours. This is because there are many more unstructured roads than structured roads in this test image dataset.

Most of the vanishing point detection methods only face to 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 fourth and fifth rows of Fig. 6) because it depends on the orientation coherence of each intersecting line pair and the color texture difference between the two parts separated by the middle line of each intersecting line pair (in the implementation we only consider line pairs in which the angles between the two lines are larger than 10 deg, and this is reasonable for general road images).

Although the method of Kong et al.[6] can do a pretty good job for unstructured road images, if the input image has some obstacles with much stronger boundaries than vehicle tracks, then the estimated location of the vanishing point will be distracted from the genuine one, for example, the rear window of the car in the image on the 2nd row in Fig. 2 and the index board in the image shown in Fig. 1. Because in their local soft voting scheme, only those pixels whose Gabor responses are larger than a threshold are qualified for voting, the voter is prone to select an incorrect vanishing point.

In order to meet the requirements of real-time applications, Moghadam et al.[7] suggested that first downsampled all of the input images to 80×60 pixels by a Gaussian filter and a downsampling operator, then detect the vanishing points on this small sized images, and finally project the positions of the detected vanishing points back to the original sized images. However, the processes of downsampling 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 images at their original sizes ($180 \times 240\ pixels$).

5 Conclusions

Detecting the vanishing point from a single road image is a very 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[1,2] and texture-based methods.[4,6,7] Edge-based methods are fit for structured road images, while texture-based methods are more suitable for unstructured road conditions 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 detecting the vanishing point than does the selection of filter banks.

Based on the analysis of both the advantages and the limitations of edge-based methods and texture-based methods, we proposed a new vanishing point estimation method in this paper that integrates the efficiency of line segments of edge-based methods and the orientation coherence concept frequently used in 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 that contains over 1000 road images. 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 in detecting the vanishing point when compared to the state-of-the-art edge-based and texture-based methods, especially for structured roads.

References

1. Y. Wang, E. K. Teoh, and D. Shen, "Lane detection and tracking using B-snake," *Image Vision Comput.* **22**, 269–280 (2004).
2. T. Suttorp and T. Bucher, "Robust vanishing point estimation for driver assistance," in *Proc. of the IEEE Intelligent Transportation Systems Conference* (2006).
3. Y. He, H. Wang, and B. Zhang, "Color-based road detection in urban traffic scenes," *IEEE T Int Trans Syst* **5**(4), 309–318 (2004).
4. C. Rasmussen, "Grouping dominant orientations for ill-structured road following," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition* (2004).
5. Y. Alon, A. Ferencz, and A. Shashua, "Off-road path following using region classification and geometric projection constraints," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition* (2006).
6. H. Kong, J.-Y. Audibert, and J. Ponce, "General road detection from a single image," *IEEE Trans. Image Process.* **19**(8), 2211–2220 (2010).
7. P. Moghadam, J. A. Starzyk, and W. S. Wijesoma, "Fast vanishing-point detection in unstructured environments," *IEEE Trans. Image Process.* **21**(1), 425–430(2012).
8. J.-P. Tardif, "Non-iterative approach for fast and accurate vanishing point detection," in *Proc. of IEEE International Conf. on Computer Vision* (2009).
9. J. M. Alvarez, T. Gevers, and A. M. Lopez, "3D scene priors for road detection," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition* (2010).
10. Q. Wu, W. Zhang, and T. Chen, "Prior-based vanishing point estimation through global perspective structure matching," in *Proc. of IEEE International Conference on Acoustic Speech and Signal Processing* (2010).
11. J. Canny, "A computational approach to edge detection," *IEEE Trans. On Pattern Analysis and Machine Intelligence* **PAMI-8**(6), 679–698 (1986).
12. R. T. Collins and R. S. Weiss, "Vanishing point calculation as a statistical inference on the unit sphere," in *Proc. of IEEE International Conference on Computer Vision* (1990).
13. H. Kogan, R. Maurer, and R. Keshet, "Vanishing points estimation by self-similarity," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition* (2009).
14. T. S. Lee, "Image representation using 2D Gabor wavelets," *IEEE Trans. Pattern Anal. Mach. Intell.* **18**(10), 959–971 (1996).
15. H. Farid and E. P. Simoncelli, "Optimally rotation-equivariant directional derivative kernels," in *Proc. of Intl. Conf. Computer Analysis of Images and Patterns*, pp. 207–214, University of Pennsylvania, Philadelphia PA (1997).

16. J. M. Coughlan and A. L. Yuille, "Manhattan world: compass direction from a single image by Bayesian inference," in *Proc. of IEEE International Conference on Computer Vision*, pp. 941–947 (1999).
17. P. Denis, J. H. Elder, and F. J. Estrada, "Efficient edge-based methods for estimating manhattan frames in urban imagery," in *Proc. of European Conference on Computer Vision* (2), pp. 197–210 (2008).
18. O. Barinova et al., "Geometric image parsing in man-made environments," in *Proc. of the 11th European Conference on Computer Vision* (2010).
19. D. C. Lee, M. Hebert, and T. Kanade, "Geometric reasoning for single image structure recovery," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition* (2009).
20. R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE Trans. Acoust. Speech Signal Process.* **29**(6), 1153–1160 (1981).

Xiqun Lu received her BS and MS degrees in electrical and electronics engineering from Hangzhou University, Hangzhou, in 1991 and 1994, respectively, and her PhD degree in electrical and electronics engineering from South China University of Technology, Guangzhou, 1997. In 1997, she joined Zhejiang University, Hangzhou, where she is currently an associate professor with College of Computer Science. Her research interests are in image processing and computer vision.