



Camera orientation estimation using voting approach on the Gaussian sphere for in-vehicle camera

YOUNGRAN JO,^{1,2} JINBEUM JANG,^{1,2} MINWOO SHIN,¹ AND JOONKI PAIK^{1,*}

¹Department of Image, Chung-Ang University, 84 Heukseok-ro, Dongjak-gu, Seoul 06974, Korea

²These authors equally contributed to this work.

*paikj@cau.ac.kr

Abstract: Calibration of a vehicle camera is a key technology for advanced driver assistance systems (ADAS). This paper presents a novel estimation method to measure the orientation of a camera that is mounted on a driving vehicle. By considering the characteristics of vehicle cameras and driving environment, we detect three orthogonal vanishing points as a basis of the imaging geometry. The proposed method consists of three steps: i) detection of lines projected to the Gaussian sphere and extraction of the plane normal, ii) estimation of the vanishing point about the optical axis using linear Hough transform, and iii) voting for the rest two vanishing points using circular histogram. The proposed method increases both accuracy and stability by considering the practical driving situation using sequentially estimated three vanishing points. In addition, we can rapidly estimate the orientation by converting the voting space into a 2D plane at each stage. As a result, the proposed method can quickly and accurately estimate the orientation of the vehicle camera in a normal driving situation.

© 2019 Optical Society of America under the terms of the [OSA Open Access Publishing Agreement](#)

1. Introduction

Autonomous driving systems need various types of sensors such as color, radar, and light detection and range (LiDAR) sensors for accurate, integrated analysis of driving situation to guarantee human safety and convenience. Especially, a complementary metal-oxide semiconductor (CMOS) imaging sensor is widely used in video recording systems and advanced driver assistance systems (ADAS) for around-view monitoring (AVM) because of low cost and the similar characteristics to the human vision [1, 2]. Recently, advanced technologies using a CMOS sensor are being studied to realize autonomous vehicles [3]. A number of deep learning-based object detection algorithms were developed for the next generation vehicles to provide visual intelligence. Three-dimensional (3D) imaging techniques ranging from stereo matching to dense 3D reconstruction are another important technical basis for autonomous driving systems [4]. Image sensor-based approaches can exploit various advantages that have been developed in image processing and computer vision fields, including a pre-processing algorithm to enhance the quality of input image [5], image-based depth-map estimation [6], and automatic calibration using a single camera [7], to name a few.

Camera calibration is the most important task in 3D imaging technology since it provides both intrinsic and extrinsic camera parameters associated with the geometric relationship between 3D world space and 2D imaging sensor. Conventional calibration methods used a special pattern such as a checkerboard [8] or orthogonal array of dots [9]. However, in-vehicle camera calibration is a challenging problem since a very large calibration pattern is needed and the camera is frequently dislocated due to the nature of dynamic driving.

To estimate and correct camera orientation during operation, a number of online camera calibration methods using vanishing points (VPs) were proposed [10–12]. In general, a VP can be

extracted by finding an intersection of the lines projected from parallel structures in a 3D world. More specifically, a VP extraction process is performed in either the *image space* or *Gaussian unit sphere*. Image-based approaches use line segments in the image [13–18]. Wu *et al.* proposed a voting method in the image space using a weight for robust estimation [14]. Elloumi *et al.* used a random sample consensus (RANSAC) algorithm to estimate three VPs for camera orientation estimation by separately considering both infinite and finite VPs [15]. J-linkage algorithm uses a modified random sampling method [16]. To reduce the computational complexity of the J-linkage algorithm, fast J-linkage algorithm was proposed by setting the initial hypothesis considering the length of the line [18]. Although the image-based approach can be used even when intrinsic parameters are unknown, its accuracy is low because of an inaccurate approximation of infinite parallel lines. On the other hand, the Gaussian sphere-based approach transforms 2D data into a spherical surface [19–23]. Since the Gaussian sphere is a finite space, both finite and infinite VPs are treated as the same in the sphere. 3-line RANSAC algorithm estimates an orthogonal VP triplet using the Gaussian sphere [21]. Although the RANSAC algorithm is fast and robust to noise, classification result depends on the random selection process, and therefore it does not guarantee the optimal solution. Another approach uses a branch-and-bound (BnB) algorithm to estimate the optimal camera orientation. It considers the rotation estimation problem as a convex problem that can be solved using interval analysis [22] and parametric space [23]. Lu *et al.* combined 2-line RANSAC with an exhaustive search scheme to find global solutions without significantly increasing the computational burden [24]. There was an approach to use the dual space that does not need camera calibration. More specifically, Lezama *et al.* used PCLines to transform lines to points in the dual space [25]. Furthermore, tracking-based methods that trace lines, motions or planes and estimates the relationship between two adjacent frames were proposed for stability of the estimated angles [15, 26–29].

In this paper, we present a novel in-vehicle camera orientation estimation method by finding three orthogonal VPs under assumption that the vehicle drives straight ahead in the Manhattan world [30, 31]. To ensure the orthogonality of the detected VP, lines in the image are converted to the corresponding plane normal vector on a spherical space and the voting algorithm is used. In order to efficiently estimate the vanishing point in the vehicle environment, the proposed method first estimates the VP along the driving direction which is the Z-axis of the vehicle coordinate system. The VP along the driving direction is extracted using linear Hough transform. In this step, unit plane normal vectors are scaled to make the problem into the 2D line fitting problem. Next, the rest VPs are selected from the circular histogram, and they are orthogonal to the VP along the driving direction. Finally, we estimate the camera orientation using three orthogonal VPs.

The proposed method is designed to speed up the camera orientation estimation process in the context of the a real vehicle driving environment. Specifically, the proposed step-by-step VP estimation process using Hough transform and circular histogram decreases the computational time to estimate the orientation angles. The circular histogram also provides a clear standard to vote other VPs since each bin of angle has the number of normal vectors of lines orthogonal to the driving direction. For that reason, the proposed method ensures the orthogonality of the VPs. The proposed method can be applied to automatic camera calibration for ADAS because it can accurately estimate the camera orientation while the vehicle drives straightforward.

This paper is organized as follows. After introducing theoretical background in section 2, the proposed camera orientation estimation method is presented in section 3. The performance of the proposed method is verified by experimental results in section 4, and section 5 concludes the paper.

2. Theoretical background

2.1. Properties of vehicle camera geometry

A digital image acquired by an imaging sensor is defined as a set of 2D points projected from a 3D world space. Given a point in the 3D homogeneous coordinate $\mathbf{X}_W = [X_w \ Y_w \ Z_w \ 1]^T \in \mathbb{P}^3$ and a projected point in the 2D planar space $\mathbf{x}_i = [u \ v \ 1]^T \in \mathbb{P}^2$, the camera projection model is defined as

$$\mathbf{x}_i = \mathbf{P}\mathbf{X}_W, \quad (1)$$

where \mathbf{P} represents the camera projection matrix, X_w , Y_w , and Z_w respectively x , y , and z value of 3D world coordinate, u and v respectively x and y value of 2D planar coordinate. The camera projection matrix is defined as

$$\mathbf{P} = \mathbf{K}[\mathbf{R} \mid \mathbf{T}] = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}, \quad (2)$$

where \mathbf{K} represents the camera matrix containing specifications of lens and sensor, \mathbf{R} the rotation matrix, $\mathbf{T} = [t_x \ t_y \ t_z]^T$ the translation vector, f_x and f_y respectively the focal lengths in x and y directions, (c_x, c_y) the principal points or the center of projection, and s the skew value of the camera. \mathbf{K} consists of intrinsic parameters while $[\mathbf{R} \mid \mathbf{T}]$ contains extrinsic parameters including orientation and position of the camera.

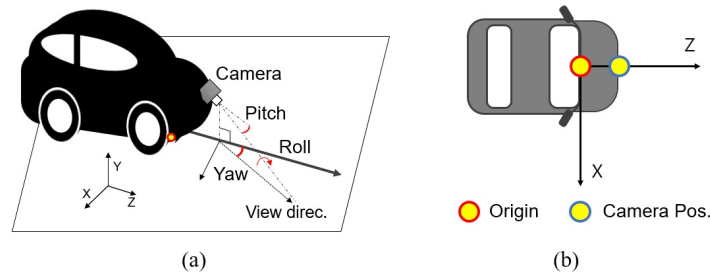


Fig. 1. Camera geometry in vehicle: (a) six extrinsic parameters including roll, pitch, yaw, and three-dimensional coordinates and (b) the origin and axis of the vehicle's 3D coordinate system. The Y -axis indicates the upward direction from the origin.

Figure 1 shows a camera geometry model in the vehicle coordinate system, where the camera calibration is performed on the vehicle's 3D coordinate system. For that reason, there are no changes in camera parameters in terms of the vehicle coordinate system when a vehicle moves straight ahead on a flat, well-paved road.

2.2. Orientation estimation using Gaussian sphere

A Gaussian sphere is defined as a unit sphere, and the principal point of a camera is mapped to the center of the Gaussian sphere. In transforming a 2D image plane to the 3D Gaussian sphere, the coordinate is shifted to the principal point and normalized by the focal length. Given a point \mathbf{x}_i in the image space, the point \mathbf{x}_g projected onto the Gaussian sphere can be obtained using the intrinsic matrix [27] as

$$\mathbf{x}_g = \mathbf{K}^{-1} \mathbf{x}_i. \tag{3}$$

Also, \mathbf{x}_g can be represented using the corresponding point \mathbf{X}_W in the world space as

$$\mathbf{x}_g = \mathbf{K}^{-1} \mathbf{P} \mathbf{X}_W = [\mathbf{R} \mid \mathbf{T}] \mathbf{X}_W. \tag{4}$$

Since the intrinsic parameters are removed from the transformation process, it is convenient to consider the orthogonality property of VPs. In addition, rotation can be directly estimated using the orthogonal VP triplet.

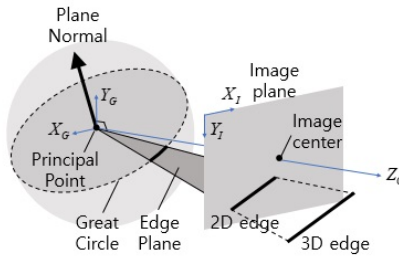


Fig. 2. Relationship between the image plane of a camera and the corresponding Gaussian sphere. \mathbf{X}_G , \mathbf{Y}_G , and \mathbf{Z}_G represent axes of the Gaussian space, and \mathbf{X}_I and \mathbf{Y}_I represent axes of the image plane.

Figure 2 shows the relationship between the image plane of a camera and the corresponding Gaussian sphere. The image plane contains a 2D edge or line and the principal point of the camera lies on the edge plane. The intersection of the edge plane and Gaussian sphere generates a great circle that represents the line in the Gaussian sphere. The normal vector of the edge plane defines the plane normal of the line [32].

When parallel lines in the world coordinate are projected onto the Gaussian sphere, they generate VPs at antipodal points in the Gaussian sphere as shown in Fig. 3(a). The line passing the principal point and VPs is called a vanishing direction (VD). Since plane normals of the parallel lines generate a distribution shaped like a great circle, the normal of the great circle coincides with the VD of the parallel lines as shown in Fig. 3(b).

Given a VD $\mathbf{V} = [v_x \ v_y \ v_z \ 0]^T \in \mathbb{P}^3$, the transformed coordinate by the extrinsic parameters is obtained as

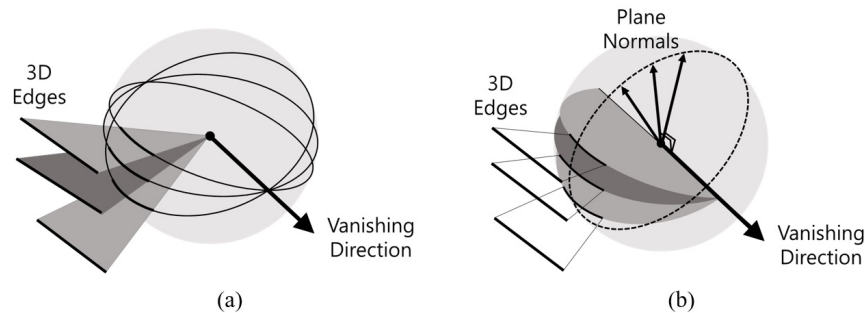


Fig. 3. Vanishing direction (VD) representations using the Gaussian sphere model: (a) VD representation using plane normals and (b) VD representation using the intersection of great circles.

$$\mathbf{V}' = [\mathbf{R} \mid \mathbf{T}]\mathbf{V} = \mathbf{R}[v_x \quad v_y \quad v_z]^T, \quad (5)$$

which states that the transformation of VD is influenced by only rotation, and it is invariant to translation. As a result, orientation can be estimated according to VD. Especially when the orthogonal VD triplet $\mathbf{V}_c = [\mathbf{V}_1 \ \mathbf{V}_2 \ \mathbf{V}_3]$ is known, rotation matrix \mathbf{R} about the world coordinate can be simply obtained from the following equation [10]

$$\mathbf{V}_c = \mathbf{R}\mathbf{I}, \quad (6)$$

where \mathbf{I} is the 3x3 identity matrix about the world coordinate.

3. Proposed method

In a vehicle camera system, the camera orientation can be estimated using a rotation vector. Therefore, it is necessary for online calibration to quickly analyze input images while driving and calculate the optimum angle of rotation. In this section, we present a novel camera orientation estimation method for online calibration in vehicle camera systems.

3.1. Overview

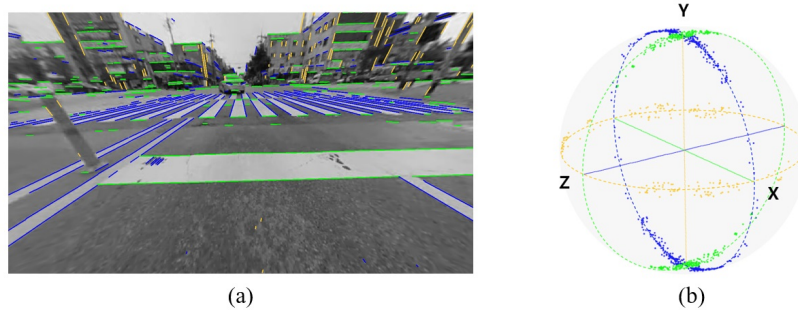


Fig. 4. Relationship between Manhattan world and Gaussian sphere: (a) a street image that satisfies Manhattan world assumption and detected line segments and (b) the corresponding plane normal vectors distributed in the Gaussian sphere.

The proposed method estimates the orientation of the camera in a vehicle that moves straight ahead in the Manhattan world [30]. The world coordinate system is aligned with the direction of the Manhattan world. Specifically, the moving direction of the vehicle becomes the Z-axis, the vertical direction becomes the Y-axis, and the horizontal direction becomes the X-axis. As shown in Fig. 4(a), we could detect a sufficient number of the horizontal and vertical groups of lines. In particular, the straightforward movement of the vehicle makes many blue lines in the z direction, and the rectangular structure of the Manhattan world makes green and yellow lines in the x and y directions, respectively. Figure 4(b) shows that normal vectors corresponding to the three groups of lines are distributed on three great circles in the Gaussian sphere. The three great circles are mutually orthogonal in the spherical space under the Manhattan world assumption. We assume that: i) the vehicle moves straight ahead to avoid the angle variation due to non-straight motions, and ii) the intrinsic matrix used in the projection onto the Gaussian sphere is a known constant because an in-vehicle camera commonly uses a fixed-focus camera and because intrinsic parameters are *a priori* determined.

Figure 5 shows the block diagram of the proposed camera orientation estimation algorithm. We first detect line segments L from the input image $f(x, y)$ using the line segment detection

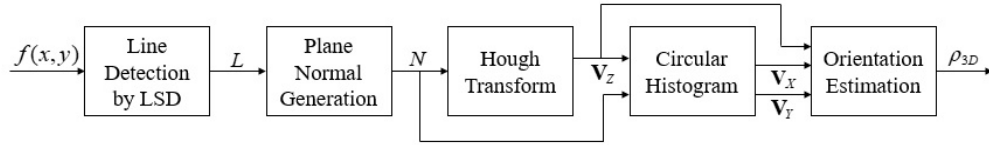


Fig. 5. Block diagram of the proposed camera orientation estimation algorithm.

(LSD) algorithm and the corresponding plane normal vectors N are computed in the Gaussian sphere. After projecting normal vectors into a plane of a cube, the proposed method performs the linear Hough transform to obtain the Z -axis representing vehicle's moving direction. To estimate two other axes, we compute a circular histogram using the orthogonal property to the Z -axis. Since each estimated axis are represented as the vanishing point, the proposed method finally obtains three camera orientation angles such as pitch, yaw, and roll.

3.2. Line segment detection and normal vector generation

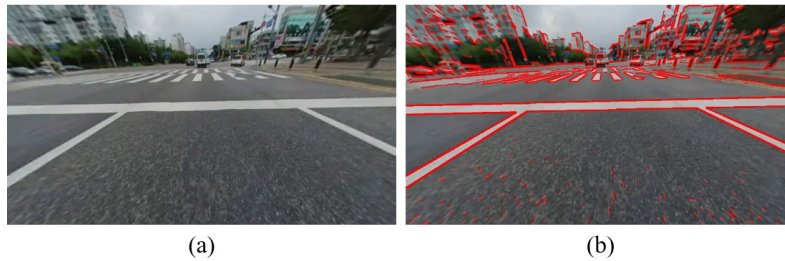


Fig. 6. Line segment detection result: (a) input image and (a) the result of LSD.

Given an input image $f(x, y)$, the proposed method starts from line detection for the computation of three main axes of the camera coordinate system. The input image has structures satisfying the geometric orthogonality by projection from the Manhattan world. For camera orientation estimation, it is necessary to detect solid lines from structures, rather than gradient information. For that reason, the proposed method detects lines using the line segment detection (LSD) algorithm [33]. After preprocessing for noise reduction using a simple Gaussian filter, line candidates of local regions are detected by calculating angles θ as

$$\theta = \arctan\left(\frac{\nabla f_x(x, y)}{-\nabla f_y(x, y)}\right), \quad (7)$$

where

$$\begin{aligned} \nabla f_x(x, y) &= \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} * f(x, y), \\ \nabla f_y(x, y) &= \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} * f(x, y), \end{aligned} \quad (8)$$

and $\nabla f_x(x, y)$ and $\nabla f_y(x, y)$ respectively represent the horizontal and vertical gradients. Finally, a line is extracted by searching line candidates with similar θ . As a result, we obtain major structures by lines as shown in Fig. 6.

Next, detected lines are projected into the Gaussian sphere to estimate the vanishing direction. From lines in 2D image $L = \{\mathbf{l}^1, \mathbf{l}^2, \dots, \mathbf{l}^n\}$, the correspondingly projected 3D lines are given as

$$\mathbf{l}_G^i = \mathbf{K}^{-1} \mathbf{l}^i, \quad (9)$$

where \mathbf{l}_G^i denotes the line projected version of \mathbf{l}^i into the surface of the Gaussian sphere. A line in the 3D sphere represents a great circle whose center is the origin of the sphere. Likewise, two and more lines satisfying the parallelism pass through two antipodal intersections. However, incorrect intersections are actually created by various potential problems such as camera jittering and image noise. In addition, many false candidates for the vanishing direction (VD) are generated since every two lines have an intersection even if they are not parallel.

For efficient computation of VDs, the proposed method uses the unit plane normal of the great circle. Given projected lines $L_G = \{\mathbf{l}_G^1, \mathbf{l}_G^2, \dots, \mathbf{l}_G^n\}$, a unit vector of plane normal $N = \{\mathbf{n}^1, \mathbf{n}^2, \dots, \mathbf{n}^n\}$ is computed as

$$\mathbf{n}^i = \frac{\mathbf{n}_y^{i'}}{|\mathbf{n}_y^{i'}|} \mathbf{n}^{i'}, \quad (10)$$

where $\mathbf{n}^{i'}$ represents normal vector of the \mathbf{l}_G^i with the coordinate $(\mathbf{n}_x^{i'}, \mathbf{n}_y^{i'}, \mathbf{n}_z^{i'})$ and computed as

$$\mathbf{n}^{i'} = \frac{\mathbf{p}_s^i \times \mathbf{p}_e^i}{|\mathbf{p}_s^i \times \mathbf{p}_e^i|}, \quad (11)$$

and \mathbf{p}_s^i and \mathbf{p}_e^i respectively represent the start and end point of \mathbf{l}_G^i . In Eq. (10), all plane normal vectors are projected into a hemisphere satisfying $y > 0$ to prevent the direction ambiguity. Because a plane normal vector is projected onto antipodal points in the Gaussian sphere.

3.3. Vanishing direction estimation using linear Hough transform

Detected line segments in the 2D image are mapped to unit plane normals in the Gaussian sphere as shown in Fig. 7. The set of plane normals form a great circle whose normal vector represents the vanishing direction. Unfortunately, it is difficult to determine the great circle since the set of unit plane normal, $N = \{\mathbf{n}^1, \mathbf{n}^2, \dots, \mathbf{n}^n\}$, contains outliers that do not satisfy the orthogonality property in the Manhattan world. To solve this problem, the proposed method uses the unit cube whose centroid is the same to the center of the Gaussian sphere [34]. A plane normal distribution generated by a certain VD is projected onto the adjacent plane to form a line as shown in Fig. 7.

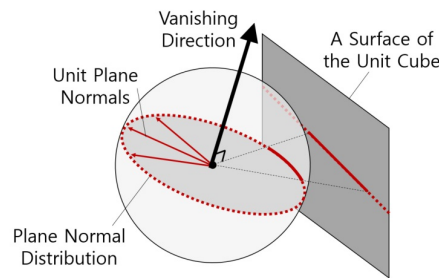


Fig. 7. Relationship between the Gaussian sphere and unit cube.

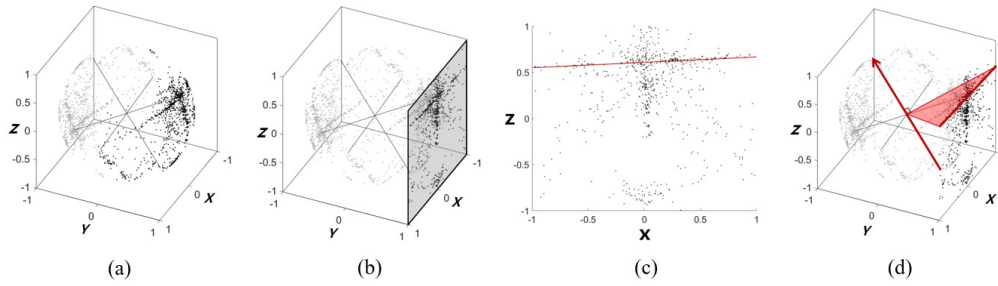


Fig. 8. Vanishing direction estimation using the linear Hough transform: (a) distribution of unit plane normals in the Gaussian sphere, (b) projection of plane normals into a 2D plane of the unit cube, (c) the strongest line estimation by linear Hough Transform, and (d) result of VD estimation.

In addition, lines in the Z-direction are mostly detected by the LSD algorithm while the vehicle is driving straightforward. A step-by-step vanishing direction process is illustrated in Fig. 8.

Given unit plane normals that are transformed from the detected line segments are shown in Fig. 8(a), the proposed method estimates the VD by extracting a strongest line using the linear Hough transform. We then project plane normals on the 3D spherical surface into the 2D plane satisfying $y = 1$ to define the Z-axis as vehicle's driving direction. A projected point of \mathbf{n}^i is defined as

$$\mathbf{u}^i = \{\mathbf{n}^i / \mathbf{n}_y^i \mid |\mathbf{n}_x^i| \leq 1, |\mathbf{n}_z^i| \leq 1\}. \quad (12)$$

Next, the line for \mathbf{n}^i is estimated using the linear Hough transform. To generate the accumulated space, we use some parameters such as angle θ in the range of $(-\pi/2, \pi/2)$, offset μ in the range of $(-1, 1)$, and interval between adjacent bins of 0.01. Therefore, the proposed method obtains the Z-axis VD \mathbf{V}_Z by computing maximum values of two parameters, θ_{max} and μ_{max} , as

$$\mathbf{V}_Z = \mathbf{v}_1 \times \mathbf{v}_2. \quad (13)$$

where \mathbf{v}_1 and \mathbf{v}_2 represent end points of the extracted line by the linear Hough transform as

$$\begin{aligned} \mathbf{v}_1 &= \begin{bmatrix} 1 & 1 & \theta_{max} + \mu_{max} \end{bmatrix}, \\ \mathbf{v}_2 &= \begin{bmatrix} -1 & 1 & -\theta_{max} + \mu_{max} \end{bmatrix}. \end{aligned} \quad (14)$$

Figures 8(c) and 8(d) show the Z-axis VD estimation result by extracting the strongest line using the linear Hough transform. As a result, the estimated line as shown in Fig. 8(c) is re-projected onto a great circle with the Z-axis VD into 3D sphere as shown in Fig. 8(d).

3.4. Voting for the vanishing direction using circular histogram

Although the linear Hough transform can determine a main axis by searching the strongest line distribution, it is not easy to determine two other axes if plane normals are not sufficiently detected as shown in Fig. 9(a). For that reason, the proposed method casts a vote for two VDs using the geometric orthogonality with the Z-axis. If we have the reliable Z-axis VD, corresponding two great circles are orthogonal to \mathbf{V}_Z and meet the great circle of \mathbf{V}_Z on a pair of antipodal points as shown in Fig. 9(b). Given a set of two VD candidates C with the great circle for the Z-axis, an element $c(\omega)$ in the continuous range ω of $[0, 2\pi)$ lies on C . Therefore, $c(\omega)$ is located on the

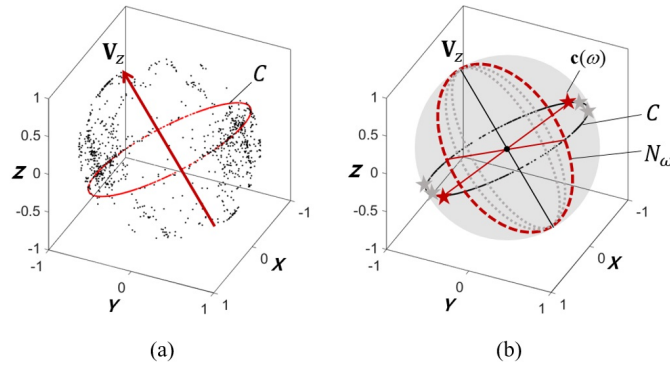


Fig. 9. Estimation of X and Y -axes: (a) distribution of plane normal on the spherical surface and the estimated Z -axis and (b) VD candidate $c(\omega)$.

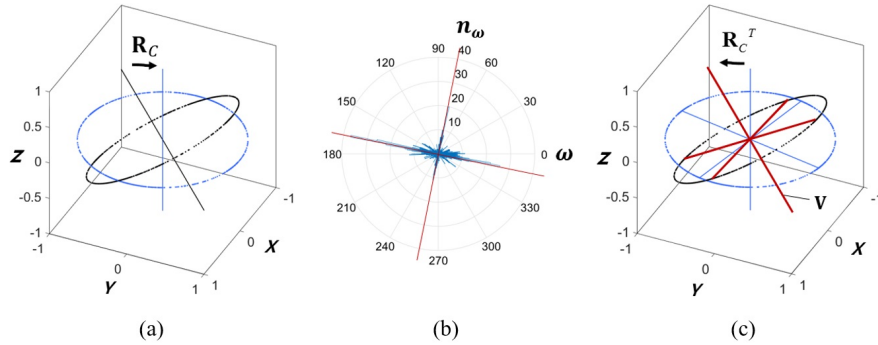


Fig. 10. Estimation of X - and Y -axes: (a) rotated vectors onto the XY -plane, (b) the corresponding circular histogram and (c) the finally estimated VD.

intersection between C and a great circle satisfying the orthogonality with C in ω , denoted as N_ω , as shown in Fig. 9(b).

To vote for two main axes, we generate a circular histogram for \mathbf{V}_Z . Given a plane normal \mathbf{n}^i , the proposed method computes a normal vector \mathbf{m}^i by computing the cross-product with \mathbf{V}_Z as

$$\mathbf{m}^i = \frac{\mathbf{V}_Z \times \mathbf{n}^i}{|\mathbf{V}_Z \times \mathbf{n}^i|}. \quad (15)$$

As a result of Eq. (15), all unit plane normals lie on C .

Since C represents a plane with the normal vector \mathbf{V}_Z , it can be considered as the rotated circle as shown in Fig. 10(a). To simplify the accumulation of $c(\omega)$, all \mathbf{m}^i in C are rotated in the plane with $z = 0$ as

$$\mathbf{m}^i = \mathbf{R}_C \mathbf{m}^i, \quad (16)$$

where \mathbf{m}^i represents the normal vector in the plane with $z = 0$, and \mathbf{R}_C the rotation matrix to transform \mathbf{m}^i in C as

$$\mathbf{R}_C = \mathbf{R}_{C_x} \mathbf{R}_{C_z} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \begin{bmatrix} \cos \beta & -\sin \beta & 0 \\ \sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (17)$$

where \mathbf{R}_{C_z} is the Z-axis rotation to transform the vector to the plane for $x = 0$, \mathbf{R}_{C_x} is the X-axis rotation to transform the vector to the plane for $y = 0$, and α and β respectively denote the angles of \mathbf{R}_{C_x} and \mathbf{R}_{C_z} that can be computed using $\mathbf{V}_z = (v_{z_x}, v_{z_y}, v_{z_z})^T$ as

$$\alpha = \arccos(v_{z_z}), \beta = \arctan(v_{z_x}/v_{z_y}). \quad (18)$$

Next, $c(\omega)$ is accumulated in the range of $\omega \in [0, 2\pi)$ from 3600 bins with the interval of $\pi/1800$. Figure 10(b) shows the circular histogram represented by the rose diagram as shown in Fig. 10(b). From the histogram, we vote the maximum index ω_{max} as

$$\omega_{max} = \max_{\omega} h(\omega), \quad (19)$$

where $h(\omega')$ represents the histogram for ω' with the range of $[0, \pi/2)$ which is accumulated from four bins of $c(\omega)$ with the interval of 90° as

$$h(\omega') = \sum_{i=0}^3 c(\omega + 90i). \quad (20)$$

Since ω_{max} contains two orthogonal points by voting from the circular histogram, it has the information of two VDs satisfying the geometrical orthogonality with \mathbf{V} . For that reason, the proposed method finally estimates two VDs by transformation of the inverse of \mathbf{R}_C as

$$\mathbf{V}_x = \mathbf{R}_C \mathbf{V}_{rot}, \quad (21)$$

where \mathbf{V}_{rot} represents the vector with ω_{max} in a plane for $z = 0$. \mathbf{V}_y is also computed by the cross-product between \mathbf{V}_x and \mathbf{V}_z as

$$\mathbf{V}_y = \mathbf{V}_x \times \mathbf{V}_z. \quad (22)$$

Consequently, the camera orientation ρ_{3D} is finally obtained by computing the rotation matrix and Euler's formula from three VDs representing as the XYZ axis of the camera [10].

4. Experiment results

In this section, we demonstrate the feasibility of the proposed method by comparing the performance with existing methods. The experiments were performed using a personal computer with a i7-7700 4.20 GHz processor and 16 GB RAM. For performance comparison, we used 3-line RANSAC [21], J-linkage [16], and the dual space-based [25] methods. 3-line RANSAC algorithm obtains three orthogonal VDs in a Gaussian sphere and to determine the direction of the axis by making a minimal set using three randomly selected lines with repetitive random sampling. J-linkage algorithm estimates multiple VPs in the image space by creating a relationship with the edges by randomly creating a hypotheses of VPs. Dual space-based method also computes VPs using PCLines transformation to classify lines. For quantitative evaluation of three methods, we measured the mean and standard deviation of the angle in each frame.

To test the performance under the actual driving environment, we acquired three videos from a CMOS camera employed in the front side of a vehicle. More specifically, a fish-eye lens camera of 1280×720 resolution and 60 frames per second is equipped in the vehicle system to

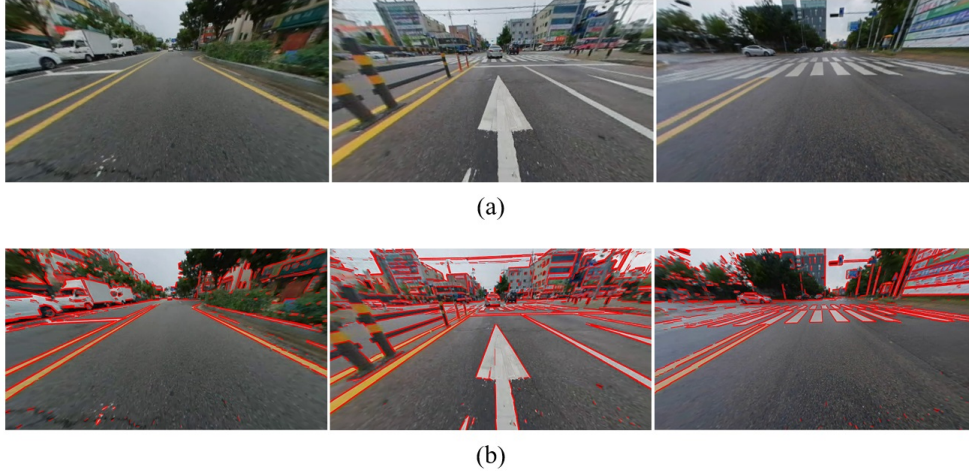


Fig. 11. A test video acquired from the real world: (a) input frame of three videos under the straightforward driving environment (see [Visualization 1](#), [Visualization 2](#), and [Visualization 3](#)) and (b) line extraction results from (a).

generate a top-view image for AVM. We assume that all intrinsic parameters including some distortion factors are known. We used a fixed focal length within a moderate range which is commonly used in a vehicle camera. In addition, the controller area network (CAN) data about steering information was used to perform the orientation estimation while the vehicle is driving straightforward. Figure 11 shows three sets of frames used in the first experiment and the line detection results, respectively. We acquired the first set of 600 frames, Video 1, on a two-lane road environment (see [Visualization 1](#)). In the first set, all the extracted lines do not correspond to the Manhattan world, especially in regions including trees and bushes. We acquired another 600 frames, Video 2, on a six-lane road (see [Visualization 2](#)). Many lines satisfying the Manhattan world show from not only road but also background with a lot of buildings. The third set of 1200 frames, Video 3, has a large number of markers on the street, and many lines of street lamps, banners and buildings are found altogether (see [Visualization 3](#)).

Table 1. Evaluated standard deviation of camera orientation estimation using four different methods.

No.	Result	3-line RANSAC [21]			J-linkage [16]			PCLines [25]			Proposed Method		
		Pitch	Yaw	Roll	Pitch	Yaw	Roll	Pitch	Yaw	Roll	Pitch	Yaw	Roll
1	Angle	-28.5	-0.69	0.96	-25.3	1.49	-7.64	-25.9	-2.68	2.28	-28.2	-0.47	1.52
	Error	0.29	0.89	0.38	3.44	3.07	8.22	2.90	1.10	1.70	0.57	1.11	0.94
	Max	22.5	49.1	41.9	33.4	25.8	44.4	90.8	63.2	44.4	7.97	26.2	11.2
	Std.	1.04	4.09	11.3	3.65	6.02	17.1	12.9	22.1	11.6	0.84	2.46	4.62
2	Angle	-28.6	0.73	1.70	-26.9	0.60	1.27	-27.5	-2.93	2.20	-28.4	-0.68	0.69
	Error	0.43	2.31	1.12	1.88	2.18	0.69	1.32	1.35	1.62	0.36	0.90	0.11
	Max	21.4	57.0	15.2	10.0	18.3	40.5	90.7	56.1	44.4	2.68	5.14	2.11
	Std.	1.09	5.27	2.62	1.21	3.05	4.44	8.20	11.3	4.85	0.49	0.88	0.76
3	Angle	-28.4	-0.86	1.06	-26.2	0.01	-3.63	-28.8	-2.75	1.72	-28.9	-1.55	0.09
	Error	0.41	0.72	0.48	2.61	1.59	4.21	0.02	1.17	1.14	0.14	0.03	0.49
	Max	24.8	46.8	36.1	18.3	23.4	18.8	88.0	51.8	31.4	1.31	3.57	10.8
	Std.	0.48	2.15	5.12	3.12	5.46	9.36	2.81	6.84	1.69	0.24	0.52	2.99

Table 1 shows the results of the first experiment in the actual driving environment. Even in an urban environment, the accuracy of estimating VD and camera orientation becomes lower as the number of non-orthogonal lines increases. Video 1 has many lines that do not match the Manhattan world assumption because of bushes and trees. In this case, the performances of the algorithm tend to be lower than in other test cases. Although the 3-line RANSAC algorithm provides the most accurate results among four algorithms, the proposed method provides a similar accuracy with the lowest standard deviation. In case of Video 2 and Video 3, the proposed method showed stable and accurate results with the lowest error and standard deviation among comparison methods. Overall, the proposed method has more stable results than other methods in real vehicle environment because it ensures the orthogonality of VDs and at the same time considers all the angles.

Table 2. Orthogonality between vanishing directions estimated by three different methods.

	3-line RANSAC [21]	J-linkage [16]	PCLines [25]	Proposed
Average	0	0.03	0.08	0
Max	0	1	0.59	0

Table 3. Running time of each part of the proposed algorithm (sec/frame).

	Line detection	Plane normal generation	Hough transform	Circular histogram	Orientation estimation	Total
Time	0.1051	0.0366	0.0209	0.0034	0.0002	0.1663

Table 4. Comparison of running time (sec/frame).

	3-line RANSAC [21]	J-linkage [16]	PCLines [25]	Proposed
Time	0.25	3.6	3.98	0.17

The second experiment tests the orthogonality of the estimated VDs. The orthogonality error is defined as inner products of each pair of VDs as

$$e = \mathbf{V}_x \cdot \mathbf{V}_y + \mathbf{V}_y \cdot \mathbf{V}_z + \mathbf{V}_x \cdot \mathbf{V}_z. \quad (23)$$

Table 2 shows the results of orthogonality error of estimated VDs using images of Fig. 11. The J-linkage algorithm has nonzero orthogonality errors since the VPs are estimated in the image space without considering the orthogonality. The PCLines algorithm extracts the VD triplet among estimated multiple VDs considering the orthogonality. However, it does not fully guarantee the orthogonality. On the other hand, the 3-line RANSAC algorithm produces orthogonal VDs since it estimates the VDs in the Gaussian sphere using minimal solution sets ensuring orthogonality. The proposed method also satisfies the orthogonality since it sequentially estimates the orthogonal VDs using the linear Hough transform and circular histogram.

Table 3 shows the processing time of each part of the proposed algorithm. Also, Table 4 shows the processing time of the three estimation methods. Although the proposed method is based on

the voting algorithm using all of the detected edges to consider all directions, it is faster than any other methods by converting accumulation space into the 2D plane using only cross products and projections.

For qualitative evaluation of the camera orientation estimation, we classified some detected lines using thresholding of the geodesic distance $d(\mathbf{V}, \mathbf{l}) = |\arcsin(\mathbf{V} \cdot \mathbf{n})|$ less than 0.07. Figure 12 shows the result of classifying lines from a real video acquired by a driving vehicle. The odd rows are the input images and the even rows are the results of line detection. The blue lines represent the classified lines of the Z -axis for the direction of straightforward driving, the yellow lines the Y -axis for the horizontal direction, and the green lines the X -axis for the vertical direction. Experimental results show that most suitable lines in the Manhattan World are shown in the right direction, which demonstrates that the proposed algorithm can accurately estimates the actual camera orientation.

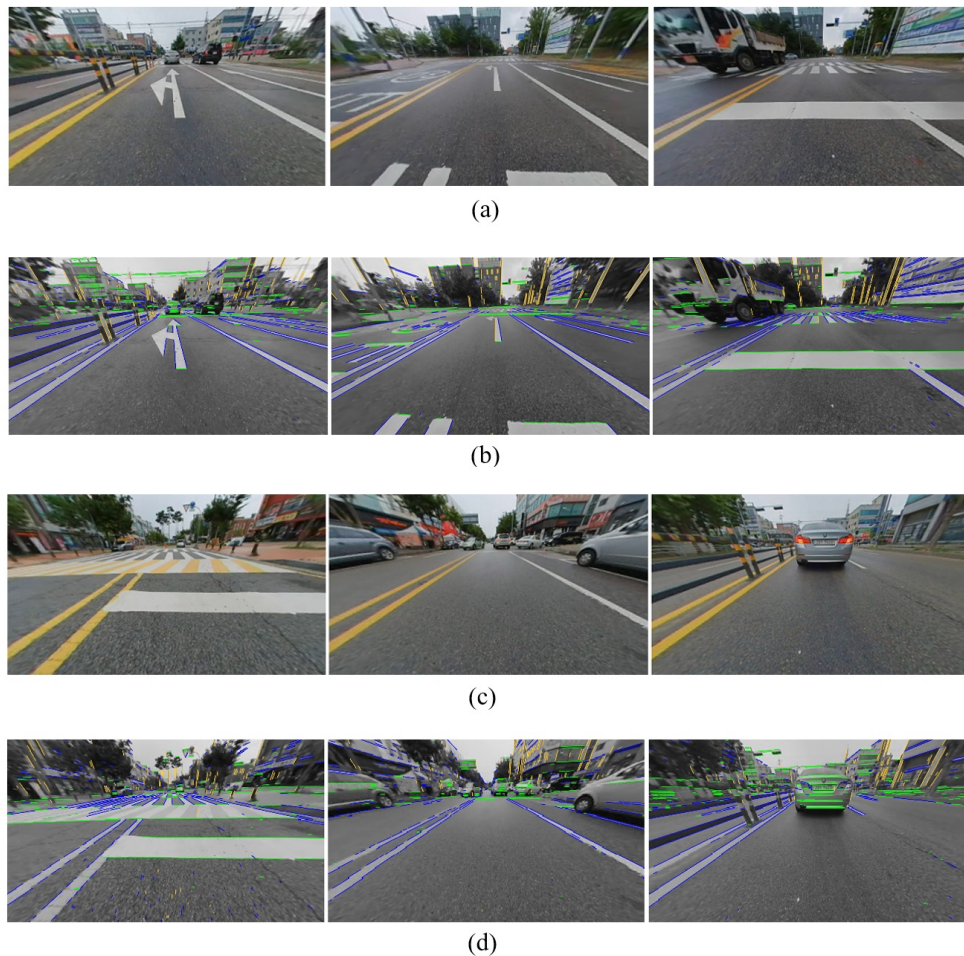


Fig. 12. Classification results using the camera orientation angles estimated from a real video: (a) the input 160th, 230th frames of first video, and 400th frame of second video, (b) the corresponding classification results, (c) the input of 530th frame of second video, 120th, and 500th frames of third video, and (d) the corresponding classification results.

5. Conclusions

In this paper, the camera orientation based on voting method is proposed for online camera calibration when vehicle drives straightforward. From some lines detected by LSD algorithm, the proposed method has the fast performance by estimating the Z-axis VD using linear Hough transform and unit cube projection. In addition, the voting method based on circular histogram provides the accurate camera angles since it sufficiently considers all detected lines into the accumulation space. Especially, the proposed method ensures the geometrical orthogonality of estimated camera angles by performing step-by-step process. Experimental results verify that the proposed method provides the stable performance in the actual driving situation as well as the ideal Manhattan world within a short period of time. Therefore, the proposed method can play a role in online calibration of the vehicle cameras in ADAS. It can be also applied in 3D object detection by measuring distance using the estimated angles for each frame. Furthermore, the proposed method can be used for view transformation of the camera to monitor surrounding circumstances in a smart parking assistance system if the camera system stores the orientation angles and computes optimal parameters.

Funding

Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2014-0-00077); Chung-Ang University Research Scholarship (2016).

References

1. P. H. Yuan, K. F. Yang, and W. H. Tsai, "Real-time security monitoring around a video surveillance vehicle with a pair of two-camera omni-imaging devices," *IEEE Trans. Veh. Technol.* **60**(8), 3603–3614 (2011).
2. Y. L. Chang, L. Y. Hsu, and O. T. C. Chen, "Auto-calibration around-view monitoring system," in *Proceedings of 2013 IEEE 78th Vehicular Technology Conference (VTC Fall)* (IEEE, 2013), pp. 1–5.
3. W. Kaddah, Y. Ouerhani, A. Alfalou, M. Desthieux, C. Brosseau, and C. Gutierrez, "Road marking features extraction using the VIAPIX[®] system," *Opt. Commun.* **371**, 117–127 (2016).
4. H. Zhou, D. Zou, L. Pei, R. Ying, P. Liu, and W. Yu, "StructSLAM: Visual SLAM with building structure lines," *IEEE Trans. Veh. Technol.* **64**(4), 1364–1375 (2015).
5. S. Park, K. Kim, S. Yu, and J. Paik, "Contrast enhancement for low-light image enhancement: A survey," *IEIE Trans. Smart Process. Comput.* **7**(1), 36–48 (2018).
6. O. Stankiewicz and M. Domański, "Depth map estimation based on maximum a posteriori probability," *IEIE Trans. Smart Process. Comput.* **7**(1), 49–61 (2018).
7. M. Shin, J. Jang, and J. Paik, "Calibration of a surveillance camera using a pedestrian homology-based rectangular model," *IEIE Trans. Smart Process. Comput.* **7**(4), 305–312 (2018).
8. Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Analysis Mach. Intell.* **22**(11), 1330–1334 (2000).
9. J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (IEEE, 1997), pp. 1106–1112.
10. B. Caprile and V. Torre, "Using vanishing points for camera calibration," *Int. J. Comput. Vis.* **4**(2), 127–139 (1990).
11. S. T. Barnard, "Interpreting perspective images," *Artif. Intell.* **21**(4), 435–462 (1983).
12. H. Wildenauer and A. Hanbury, "Robust camera self-calibration from monocular images of Manhattan worlds," in *Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2012), pp. 2831–2838.
13. M. Hornáček and S. Maierhofer, "Extracting vanishing points across multiple views," in *Proceedings of 2011 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2011), pp. 953–960.
14. Z. Wu, W. Fu, R. Xue, and W. Wang, "A novel line space voting method for vanishing-point detection of general road images," *Sensors* **16**(7), 948–960 (2016).
15. W. Elloumi, S. Treuillet, and R. Leconge, "Real-time camera orientation estimation based on vanishing point tracking under Manhattan world assumption," *J. Real-Time Image Process.* **13**(4), 1–16 (2014).
16. J. P. Tardif, "Non-iterative approach for fast and accurate vanishing point detection," in *Proceedings of 2009 IEEE 12th International Conference on Computer Vision* (IEEE, 2009), pp. 1250–1257.
17. Y. Xu, S. Oh, and A. Hoogs, "A minimum error vanishing point detection approach for uncalibrated monocular images of man-made environments," in *Proceedings of 2013 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2013), pp. 1376–1383.
18. C. H. Chang and N. Kehtarnavaz, "Fast J-linkage algorithm for camera orientation applications," *J. Real-Time Image Process.* **14**(4), 823–832 (2018).

19. M. J. Magee and J. K. Aggarwal, "Determining vanishing points from perspective images," *Comput. Vision, Graph. Image Process.* **26**(2), 256–267 (1984).
20. M. Antunes and J. P. Barreto, "A global approach for the detection of vanishing points and mutually orthogonal vanishing directions," in *Proceedings of 2013 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2013), pp. 1336–1343.
21. J. C. Bazin and M. Pollefeys, "3-line RANSAC for orthogonal vanishing point detection," in *Proceedings of 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2012), pp. 4282–4287.
22. J. C. Bazin, Y. Seo, C. Demonceaux, P. Vasseur, K. Ikeuchi, I. Kweon, and M. Pollefeys, "Globally optimal line clustering and vanishing point estimation in Manhattan world," in *Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2012), pp. 638–645.
23. J. C. Bazin, Y. Seo, and M. Pollefeys, "Globally optimal consensus set maximization through rotation search," in *Proceedings of Asian Conference on Computer Vision* (Springer, 2012), pp. 539–551.
24. X. Lu, J. Yaoy, H. Li, and Y. Liu, "2-line exhaustive searching for real-time vanishing point estimation in Manhattan world," in *Proceedings of 2017 IEEE Winter Conference on Applications of Computer Vision* (IEEE, 2017), pp. 345–353.
25. J. Lezama, G. Randall, and R. G. von Gioi, "Vanishing point detection in urban scenes using point alignments," *Image Process. On Line* **7**, 131–164 (2017).
26. T. Kroeger, D. Dai, and L. Van Gool, "Joint vanishing point extraction and tracking," in *Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2015), pp. 2449–2457.
27. J. Lee and K. Yoon, "Real-time joint estimation of camera orientation and vanishing points," in *Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2015), pp. 1866–1874.
28. W. Elloumi, S. Treuillet, and R. Leconge, "Tracking orthogonal vanishing points in video sequences for a reliable camera orientation in Manhattan world," in *Proceedings of 2012 5th International Congress on Image and Signal Processing* (IEEE, 2012), pp. 128–132.
29. R. Guo, K. Peng, D. Zhou, and Y. Liu, "Robust visual compass using hybrid features for indoor environments," *Electronics* **8**(2), 220–236 (2019).
30. J. M. Coughlan and A. L. Yuille, "Manhattan world: Compass direction from a single image by bayesian inference," in *Proceedings of the 7th IEEE International Conference on Computer Vision* (IEEE, 1999), pp. 941–947.
31. W. J. Kim and S. W. Lee, "Depth estimation with Manhattan world cues on a monocular image," *IEIE Trans. Smart Process. Comput.* **7**(3), 201–209 (2018).
32. M. E. Antone and S. Teller, "Automatic recovery of relative camera rotations for urban scenes," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2000), pp. 282–289.
33. R. G. von Gioi, J. Jakubowicz, J. M. Morel, and G. Randall, "LSD: a line segment detector," *Image Process. On Line* **2**, 35–55 (2012).
34. T. Tuytelaars, M. Proesmans, and L. Van Gool, "The cascaded hough transform as support for grouping and finding vanishing points and lines," in *Proceedings of International Workshop on Algebraic Frames for the Perception-Action Cycle* (Springer, 1997), pp. 278–289.