SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

January 2019, Vol. 62 012202:1–012202:13
https://doi.org/10.1007/s11432-017-9306-x

FVO: floor vision aided odometry

Wenjun LV[1], Yu KANG[1,2*] & Jiahu QIN[1]

[1]*Department of Automation, University of Science and Technology of China, Hefei* 230027*, China;*
[2]*State Key Laboratory of Fire Science, University of Science and Technology of China, Hefei* 230027*, China*

Received 25 July 2017/Accepted 6 November 2017/Published online 15 August 2018

Abstract In many indoor scenarios, such as restaurants, laboratories, and supermarkets, the planar floors are covered with rectangular tiles. We realized that the abundant parallel lines and crossing points formed by tile joints can be used as natural features to assist indoor localization, and thus we propose a novel indoor localization method for mobile robots by fusing odometry and monocular vision. The method comprises three steps. First, the heading and location of the mobile robot are approximately estimated by odometry based on incremental encoders. Second, with the aid of a camera, the lens of which points vertically toward the floor, the odometric heading estimation can be corrected by detecting the relative angle between the robot's heading and the tile joints. Third, the odometric location estimation is corrected by detecting the perpendicular distance between the image center and the tile joints. As compared with the existing indoor localization methods, the proposed method, called floor vision aided odometry, is not only relatively low in economic cost and computational complexity, but also relatively high in accuracy and robustness. The effectiveness of this method is verified by a real-world experiment based on a differential-drive wheeled mobile robot.

Keywords mobile robot, indoor localization, monocular vision, odometry, tile joint

Citation Lv W J, Kang Y, Qin J H. FVO: floor vision aided odometry. Sci China Inf Sci, 2019, 62(1): 012202, https://doi.org/10.1007/s11432-017-9306-x

1 Introduction

Mobile robots, in particular wheeled mobile robots, are currently used in almost every domain, including universe exploration, factory automation, and restaurant service [1–3]. More than their maneuverability, it is their autonomy that makes mobile robots attractive, so that it is paramount that they maintain their sense of heading and location to allow them to navigate almost independently [4]. For outdoor scenarios, a satellite navigation system, which is able to achieve meter-level accuracy in open areas, is frequently used to solve this problem [5]. However, when blocked lines of sight to the satellites or multi-path propagation are present, it is not possible to achieve indoor localization by means of satellites. Hence, indoor localization has attracted great interest in both academia and industry in recent decades. Although various indoor localization methods have already been developed based on different sensing system architectures and data fusion algorithms, there is still room for developing more advanced indoor localization methods to improve their performance from different aspects.

Relative localization (also termed dead reckoning or odometry) is the process of estimating a robot's heading and location based on measured changes with respect to previous estimations. In [6–8], some

* Corresponding author (email: kangduyu@ustc.edu.cn)

advanced odometris which are adaptive to curved surfaces and rough terrains are proposed. Because odometric error diverges, it cannot in practice be used for long-term localization. However, proprioceptive sensors have the advantages of low cost, low energy consumption, high updating rate, and insensibility to environmental changes, which shows that odometry can be an irreplaceable localization method in scenarios where the performance of exteroceptive sensors is seriously affected by the working environment [9–11]. Different from relative localization, absolute localization means the ability to directly determine the robot's location with respect to a given frame of reference by using exteroceptive sensors, and therefore, it suffers no cumulative errors. The traditional absolute localization methods suffer from the following problems. The GPS is a preferred localization method for outdoor applications, but is not applicable in GPS-denied environments, such as rooms, tunnels, under water, or canyons [12]. The WiFi-based localization method can determine the robot's location by using trilateration or feature matching technology, but it still suffers from the same signal attenuation and multi-path effect as GPS [13]. The coverage of RFID-based localization method is relatively small, because a large number of RFID tags is required for large-area localization, which gives rise to problems, such as high cost and tag collisions [14]. Utilizing special LED lamps mounted on ceilings, a robot can determine its location via visual light communication [15]. However, the high cost of upgrading conventional lighting devices to LED lamps means that this localization method cannot be widely used. Magnetic fingerprint-based localization method requires low cost devices; its accuracy, however, seriously decreases with the presence of continuously varying electromagnetic interference [16].

As a promising absolute localization method, visual localization method has been an attractive research topic in recent years, because vision provides a large amount of environmental information. A common method is to use a global camera mounted in an overhead position, such as on the ceiling or a wall, to estimate a robot's heading and location by measuring its profile or the position of a marker attached to its top [17–19]. Because the surveillance camera is not mounted on the robot, this method is also referred to as off-board visual localization method. This is a relatively cheap solution, since surveillance cameras have usually already been mounted on the infrastructure of the environment, which means extra cost is incurred for devices. However, it still suffers from the following problems. Illumination conditions are one of the most challenging problems in practice, in particular in outdoor applications [20]. Extreme illumination and illumination variation may degrade the localization accuracy or even render it invalid. Additionally, the localizable area is constrained by the camera's field of view, and therefore, the expenditure on devices may be very large for large-area localization. In addition, the camera's lens may be occluded by dynamic objects (e.g., pedestrians) or static objects (e.g., furniture), which seriously limits the application of off-board visual localization method in indoor environments. On-board visual localization method with cameras directly mounted on the robot is an additional localization method that shows great promise [21–23]. The key objective is to recognize natural or artificial landmarks with known locations and then calculate the distances between the robot and the landmarks; consequently, the robot's location can be readily determined using trilateration. Similarly to that of off-board visual localization method, the performance of on-board visual localization method is influenced by extreme illumination and lens occlusion. Moreover, landmark mismatching may introduce extremely large localization errors.

In many indoor scenarios, the planar floors are covered with rectangular tiles. We realized that the abundant parallel lines and crossing points formed by tile joints can be used as natural features to assist indoor localization, and thus we propose a novel indoor localization method for mobile robots that is achieved by fusing odometry and monocular vision. The method, called floor vision aided odometry (FVO) comprises three steps. First, the heading and location of the mobile robot are roughly approximated by using odometry based on incremental encoders. Second, with the aid of a camera, the lens of which is pointed vertically to the floor, the odometric heading estimation can be corrected by detecting the relative angle between the robot's heading and the tile joints. For lack of knowledge about the tile-joint directions, the floor vision provides four possible results with respect to the robot's heading. Because odometry is accurate and reliable in the short term, we can use the odometric heading estimation to exclude the three pseudo results and then correct the odometric heading estimation, in turn using the preserved results. Third, the odometric location estimation is corrected by detecting the perpendicular

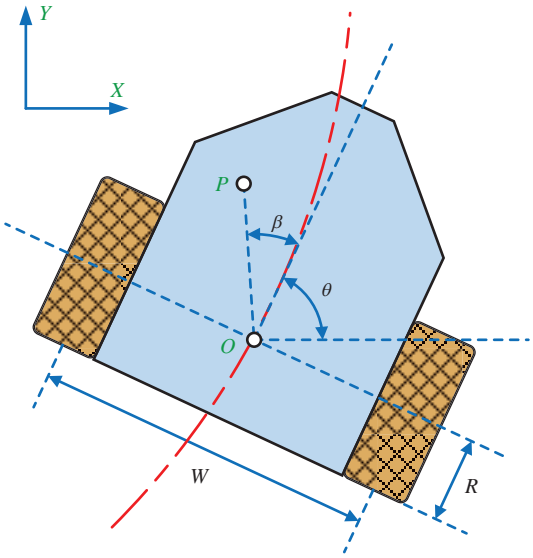

Figure 1 (Color online) Top-view schematic of mobile robot on a planar floor. The meshed yellow rectangles represent the robot's wheels and the blue polygon its body. The red non-consecutive arc represents the trajectory.

distance between the image center and the tile joints.

The rest of the paper is organized as follows. Section 2 describes the kinematics model of the mobile robot mathematically, as well as the incremental encoder-based odometry. Section 3 provides the core steps of the proposed FVO. Section 4 covers the results and analysis of a real-world experiment. The paper is concluded in Section 5.

2 Kinematics model and odometry

The kinematics of a mobile robot is schematically shown in Figure 1. Hereinafter, unless otherwise stated, we use the term mobile robot to refer to a differential-drive wheeled mobile robot, which is the type most widely used in industry.

The robot kinestates can be represented by a vector $[x_p, y_p, \theta]'$, where (x_p, y_p) denotes the coordinate of point P with respect to the world coordinate system (WCS) and θ denotes the heading angle (i.e., the angle between the heading and the X-axis). Furthermore, denoting the axle width by W, the wheel radius by R, the coordinate of axle midpoint O by (x, y), the length of the line PO by L, and the angle between the line PO and the central axis of the mobile robot by β, we have

$$x_p = x + L\cos(\theta + \beta), \tag{1a}$$

$$y_p = y + L\sin(\theta + \beta). \tag{1b}$$

The robot kinestates $[x, y, \theta]'$ and the rotational speeds of the left and right wheels (ω_r and ω_ℓ) can be related by

$$\dot{x} = \frac{R}{2}(\omega_r + \omega_\ell)\cos\theta, \tag{2a}$$

$$\dot{y} = \frac{R}{2}(\omega_r + \omega_\ell)\sin\theta, \tag{2b}$$

$$\dot{\theta} = \frac{R}{W}(\omega_r - \omega_\ell). \tag{2c}$$

Combining (1a), (1b), (2a), (2b), and (2c), we derive the kinematics model of the mobile robot, taking P as the reference point. Typically, by selecting O as the reference point, (2a), (2b), and (2c) can be read as the kinematics model, and its discrete-time form is

$$x_{t+1} = x_t + \frac{RT}{2}(\omega_{r,t} + \omega_{\ell,t})\cos\theta_t, \tag{3a}$$

$$y_{t+1} = y_t + \frac{RT}{2}(\omega_{r,t} + \omega_{\ell,t}) \sin\theta_t, \tag{3b}$$

$$\theta_{t+1} = \theta_t + \frac{RT}{W}(\omega_{r,t} - \omega_{\ell,t}), \tag{3c}$$

where T denotes the sampling period and t denotes the sequence number of the sampling points. An additional form of (3a), (3b), and (3c) based on the incremental motions is [4]

$$x_{t+1} = x_t + \delta_{d,t} \cos\left(\theta_t + \frac{\delta_{\theta,t}}{2}\right), \tag{4a}$$

$$y_{t+1} = y_t + \delta_{d,t} \sin\left(\theta_t + \frac{\delta_{\theta,t}}{2}\right), \tag{4b}$$

$$\theta_{t+1} = \theta_t + \delta_{\theta,t}, \tag{4c}$$

where $\delta_{d,t} = \frac{RT}{2}(\omega_{r,t} + \omega_{\ell,t})$ denotes the incremental displacement and $\delta_{\theta,t} = \frac{RT}{W}(\omega_{r,t} - \omega_{\ell,t})$ denotes the incremental heading, both during the sampling period from tT to $(t+1)T$. It is noted that (4a) and (4b) are not equal to (3a) and (3b) mathematically. Because θ may not be constant during the sampling period from tT to $(t+1)T$, using $\theta_t + \frac{\delta_{\theta,t}}{2}$ to replace θ_t is a simple yet useful technique used in engineering to reduce the discretization-induced error.

Therefore, using incremental encoders, the mathematical form of odometry can be expressed by

$$x_{t+1}^{(o)} = x_t^{(o)} + \delta_{d,t}^{(o)} \cos\left(\theta_t^{(o)} + \frac{\delta_{\theta,t}^{(o)}}{2}\right), \tag{5a}$$

$$y_{t+1}^{(o)} = y_t^{(o)} + \delta_{d,t}^{(o)} \sin\left(\theta_t^{(o)} + \frac{\delta_{\theta,t}^{(o)}}{2}\right), \tag{5b}$$

$$\theta_{t+1}^{(o)} = \theta_t^{(o)} + \delta_{\theta,t}^{(o)}, \tag{5c}$$

where $\delta_{d,t}^{(o)}$ and $\delta_{\theta,t}^{(o)}$ denote the measurements of $\delta_{d,t}$ and $\delta_{\theta,t}$, respectively. In this paper, the superscript (o) of the variables indicates that they are related to odometry. Based on data fusion technologies, a generalized odometry method is able to compensate the odometric errors by using exteroceptive sensors, thereby providing more accurate estimations of the robot's heading and location. Therefore, (5a), (5b), and (5c) change to

$$x_{t+1}^{(o)} = \widehat{x}_t + \delta_{d,t}^{(o)} \cos\left(\theta_t^{(o)} + \frac{\delta_{\theta,t}^{(o)}}{2}\right), \tag{6a}$$

$$y_{t+1}^{(o)} = \widehat{y}_t + \delta_{d,t}^{(o)} \sin\left(\theta_t^{(o)} + \frac{\delta_{\theta,t}^{(o)}}{2}\right), \tag{6b}$$

$$\theta_{t+1}^{(o)} = \widehat{\theta}_t + \delta_{\theta,t}^{(o)}, \tag{6c}$$

where $\widehat{x}_t$, $\widehat{y}_t$, and $\widehat{\theta}_t$ denote the outputs of generalized odometry at sampling point tT, which are optimal estimations in some sense, obtained by fusing the data from exteroceptive sensors and odometry.

In reality, $\delta_{d,t}^{(o)}$ and $\delta_{\theta,t}^{(o)}$ cannot reflect the robot's real incremental motion because of the presence of uncertainties, which are caused mainly by wheel deformation and wheel slip, as well as by irregular flooring [24]. Hence, odometry can provide an approximate estimation only in the short term, since its uncertainties grow unboundedly over time. In the existing literature, the majority of studies treated the uncertainties as Gaussian white noise, which is far divorced from reality. Let us take the wheel deformation as an example. The wheel radius is naturally time-varying at a slow rate because of barometric variation and tire abrasion, and therefore, the errors caused by wheel deformation cannot satisfy the assumption of white noise. Hence, in this paper, the uncertainties of $\delta_{d,t}^{(o)}$ and $\delta_{\theta,t}^{(o)}$ are assumed to be unknown-but-bounded (UBB) noise, which means

$$\delta_{d,t} \in \Delta_{d,t}^{(o)} = \left[\delta_{d,t}^{(o)} - \Delta\delta_{d,t}^{(o)},\ \delta_{d,t}^{(o)} + \Delta\delta_{d,t}^{(o)}\right], \tag{7a}$$

$$\delta_{\theta,t} \in \Delta_{\theta,t}^{(o)} = \left[\delta_{\theta,t}^{(o)} - \Delta\delta_{\theta,t}^{(o)},\ \delta_{\theta,t}^{(o)} + \Delta\delta_{\theta,t}^{(o)}\right], \tag{7b}$$

where $\Delta\delta_{d,t}^{(o)} \geqslant 0$ and $\Delta\delta_{\theta,t}^{(o)} \geqslant 0$ denote the odometric deviation radii, which meet

$$\Delta\delta_{d,t}^{(o)} = k_d \cdot \left|\delta_{d,t}^{(o)}\right|, \tag{8a}$$

$$\Delta\delta_{\theta,t}^{(o)} = k_\theta \cdot \left|\delta_{\theta,t}^{(o)}\right|, \tag{8b}$$

where k_d and k_θ are the deviation coefficients that are dependent on the mobile robot and its working environment. They should be determined experimentally by performing and analyzing representative movements.

Therefore, it is better to calculate the intervals containing all the possible true states under the assumption of UBB noise. We have

$$\mathcal{X}_{t+1}^{(o)} = \widehat{\mathcal{X}}_t + \Delta_{d,t}^{(o)} \cos\left(\Theta_t^{(o)} + \frac{\Delta_{\theta,t}^{(o)}}{2}\right), \tag{9a}$$

$$\mathcal{Y}_{t+1}^{(o)} = \widehat{\mathcal{Y}}_t + \Delta_{d,t}^{(o)} \sin\left(\Theta_t^{(o)} + \frac{\Delta_{\theta,t}^{(o)}}{2}\right), \tag{9b}$$

$$\Theta_{t+1}^{(o)} = \widehat{\Theta}_t + \Delta_{\theta,t}^{(o)}, \tag{9c}$$

where $\mathcal{X}_t^{(o)}$, $\mathcal{Y}_t^{(o)}$, $\Theta_t^{(o)}$, $\widehat{\mathcal{X}}_t$, $\widehat{\mathcal{Y}}_t$, and $\widehat{\Theta}_t$ are intervals in which $x_t^{(o)}$, $y_t^{(o)}$, $\theta_t^{(o)}$, $\widehat{x}_t$, $\widehat{y}_t$, and $\widehat{\theta}_t$ lie, respectively. The operations and functions involved in interval arithmetic are provided in the appendix. As compared with the conventional odometry using (5a), (5b), and (5c), which give the single vector estimations of the robot kinestates, the equations (9a), (9b), and (9c) provide the interval vector estimation within which the true states are guaranteed to fall. If a single vector estimation is needed, the interval midpoints can be used, that is

$$x_t^{(o)} = \mathrm{mid}\left(\mathcal{X}_t^{(o)}\right), \tag{10a}$$

$$y_t^{(o)} = \mathrm{mid}\left(\mathcal{Y}_t^{(o)}\right), \tag{10b}$$

$$\theta_t^{(o)} = \mathrm{mid}\left(\Theta_t^{(o)}\right), \tag{10c}$$

where $\mathrm{mid}(\cdot)$ denotes the interval midpoint. It is noted that they are not, however, mathematically the optimal estimations.

3 Floor visual correction

The floor visual correction procedure constitutes essentially a three-step strategy: (1) floor visual features extraction, (2) robot heading correction, and (3) robot location correction.

3.1 Floor visual features extraction

The first step is elementary but crucial to the success of floor visual correction. Although only a few interferences appear in the imaging formation, the motion blur, tile patterns, and line non-smoothness should be taken into consideration in the image preprocessing step. After deblurring, binarization, skeletonization and pruning, the original image is transformed into a binary image with the tile-joint lines.

A popular feature extraction method in the domain of image processing, the standard Hough transformation, is able to detect objects having a line form via a voting algorithm. First, the lines are parameterized as follows:

$$u\cos\vartheta + v\sin\vartheta = r, \tag{11}$$

where $r \in [r_{\min}, r_{\max}]$ denotes the perpendicular distance from the origin of the PCS to the line and $\vartheta \in [-\frac{\pi}{2}, \frac{\pi}{2})$ denotes the angle from the U-axis to the perpendicular. Hence, each line can be mapped to the (r, ϑ)-space (also known as the Hough space) as a point. Second, the Hough space is quantized such that it yields an $N_\vartheta \times N_r$ matrix A, which is referred to as an accumulator matrix. For a $\mu \times \nu$ image, we have

$$r_{\max} = -r_{\min} = \sqrt{\mu^2 + \nu^2}, \tag{12}$$

and the accumulator matrix A has N_ϑ elements equally dividing the interval $\vartheta \in [-\frac{\pi}{2}, \frac{\pi}{2})$ and N_r elements equally dividing the interval $r \in [r_{\min}, r_{\max}]$. The matrix indices are integers $(i, j) \in \mathbb{Z}^2$ that

$$i \in [1, N_\vartheta] \mapsto \vartheta \in \Big[-\frac{\pi}{2}, \frac{\pi}{2}\Big), \tag{13a}$$

$$j \in [1, N_r] \mapsto r \in [r_{\min}, r_{\max}]. \tag{13b}$$

Finally, each pixel point (u, v) votes for all the lines existing within the quantized Hough space, and then, the potential lines represented by (r, ϑ)-pairs can be determined by calculating the peak values of elements in the accumulator matrix. For every line, two or more (r, ϑ)-pairs with small differences in value may be obtained because of the imperfection of the pruning algorithm. To solve this problem, the (r, ϑ)-pairs should be clustered. For any two pairs, (r_i, ϑ_i)-pair and (r_j, ϑ_j)-pair, where $i \neq j$, if

$$|r_i - r_j| \leqslant \sigma_r, \tag{14a}$$

$$|\vartheta_i - \vartheta_j| \leqslant \sigma_\vartheta, \tag{14b}$$

where σ_r and σ_ϑ are similarity thresholds, which are empirically determined, the average of (r, ϑ)-pairs clustered in one group can serve as the final output of line detection.

3.2 Robot heading correction

In the previous step, the lines in the floor image are detected, thereby yielding a certain number of (r, ϑ)-pairs. The tile-joint lines are mutually parallel or orthogonal. If two tile-joint lines with ϑ_1 and ϑ_2 are parallel, we have $\vartheta_1 = \vartheta_2$. If they are orthogonal, ϑ_1 and ϑ_2 are either non-negative or negative, since $|\vartheta_1 - \vartheta_2| = \frac{\pi}{2}$ and $\vartheta_1, \vartheta_2 \in [-\frac{\pi}{2}, \frac{\pi}{2})$. Now, we examine the transformation from ϑ to θ. As shown in Figure 2(a), the four cases where the robot's heading θ lies in $(0, \frac{\pi}{2}]$, $(\frac{\pi}{2}, \pi]$, $(\pi, \frac{3\pi}{2}]$, and $(\frac{3\pi}{2}, 2\pi]$ should be analyzed respectively, because their floor vision is the same. Consider Case 1, where the robot heading lies in the first quadrant. As shown in Figure 2(b), two orthogonal lines parameterized by (r_1, ϑ_1) and (r_2, ϑ_2) are detected. Clearly, we have $\theta_{(\mathrm{c1})} = \frac{\pi}{2} - \vartheta_1 = -\vartheta_2$, where $\theta_{(\mathrm{c1})} \in (0, \frac{\pi}{2}]$ denotes the robot's heading in Case 1. Without generality, we have

$$\theta_{(\mathrm{c1})} = \begin{cases} \frac{\pi}{2} - \vartheta, & \text{if } \vartheta \geqslant 0, \\ -\vartheta, & \text{if } \vartheta < 0. \end{cases} \tag{15}$$

For Cases 2–4, similarly we have

$$\theta_{(\mathrm{c2})} = \theta_{(\mathrm{c1})} + \frac{\pi}{2}, \tag{16a}$$

$$\theta_{(\mathrm{c3})} = \theta_{(\mathrm{c2})} + \frac{\pi}{2}, \tag{16b}$$

$$\theta_{(\mathrm{c4})} = \theta_{(\mathrm{c3})} + \frac{\pi}{2}, \tag{16c}$$

where $\theta_{(\mathrm{c2})} \in (\frac{\pi}{2}, \pi]$, $\theta_{(\mathrm{c3})} \in (\pi, \frac{3\pi}{2}]$, and $\theta_{(\mathrm{c4})} \in (\frac{3\pi}{2}, 2\pi]$ denote the robot heading in Cases 2–4, respectively. Thus, we have

$$\theta \in \big\{\theta_{(\mathrm{c1})}, \theta_{(\mathrm{c2})}, \theta_{(\mathrm{c3})}, \theta_{(\mathrm{c4})}\big\} = \Big\{-\vartheta, -\vartheta + \frac{\pi}{2}, -\vartheta + \pi, -\vartheta + \frac{3\pi}{2}\Big\}, \tag{17}$$

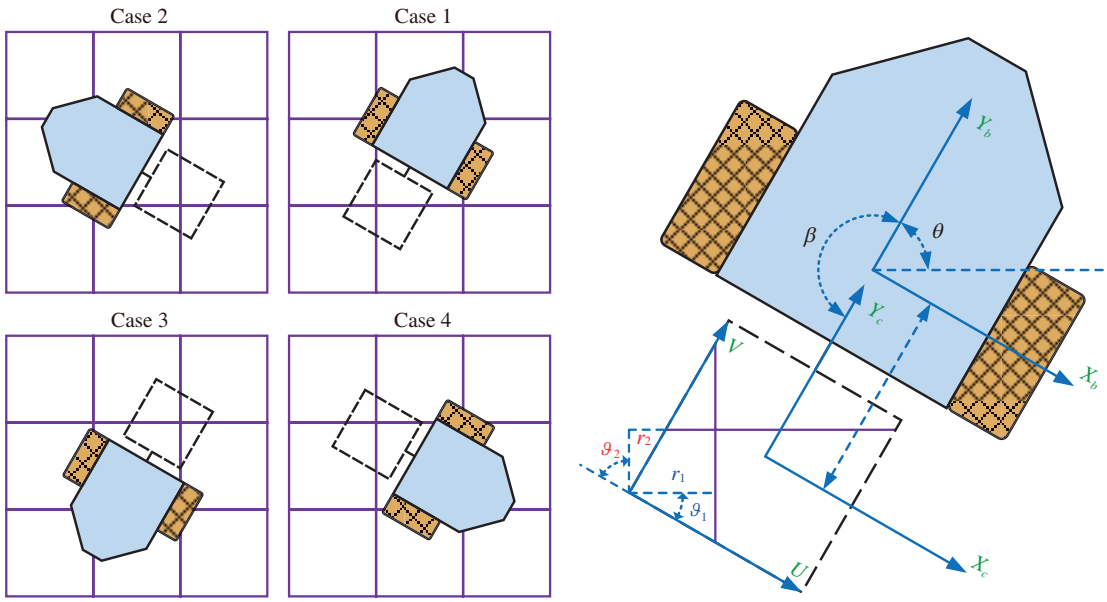

Figure 2 (Color online) (a) Relation between floor vision and robot's heading and location. The purple grid represents tile joints. The rectangle with black non-consecutive edges represents the camera's field of view. The robot's headings fall into four different quadrants, respectively, but they have the same floor vision. (b) The schematic of Case 1.

which reveals the relation between θ and ϑ.

Now, we are in a position to calculate the heading measurements obtained by the camera. Denoting by $\vartheta_{t,i}$ the parameter ϑ of the i-th line at sampling point tT, we have

$$\vartheta_{t,i} \in \Phi_{t,i}^{(c)} = \left[\vartheta_{t,i}^{(c)} - \Delta\vartheta_{t,i}^{(c)}, \vartheta_{t,i}^{(c)} + \Delta\vartheta_{t,i}^{(c)}\right], \tag{18}$$

where $\vartheta_{t,i}^{(c)}$ denotes the measurement of $\vartheta_{t,i}$ obtained by the camera and $\Delta\vartheta_{t,i}^{(c)} \geqslant 0$ denotes the deviation radius of line detection, which may be caused by the roundoff of the standard Hough transformation and spurs connected to the line skeleton. The superscript (c) indicates that the variables are derived from the camera. The subscript $i \in \mathcal{I}_t = \{1, 2, \ldots, I_t\}$, where I_t is the maximum sequence number of detected lines at sampling point tT. Combining (17) and (18) yields

$$\Theta_{t,i}^{(c)} = \left\{-\Phi_{t,i}^{(c)}, -\Phi_{t,i}^{(c)} + \frac{\pi}{2}, -\Phi_{t,i}^{(c)} + \pi, -\Phi_{t,i}^{(c)} + \frac{3\pi}{2}\right\}, \tag{19}$$

where $\Theta_{t,i}^{(c)}$ denotes the intervals in which θ_t lies deduced from the i-th line.

It is noted that other objects, such as electric wires, cannot be easily distinguished, resulting in a situation where some non-tile-joint lines may be mistakenly recognized as tile-joint lines. A non-tile-joint line has a stochastic angle, which may lead to $\Theta_t^{(o)} \cap \Theta_{t,i}^{(c)} = \emptyset$, and therefore, we have

$$\mathcal{I}_{\theta,t} = \left\{i : \Theta_t^{(o)} \cap \Theta_{t,i}^{(c)} \neq \emptyset, i \in \mathcal{I}_t\right\}, \tag{20}$$

where $\mathcal{I}_{\theta,t}$ denotes the sequence number set of the detected lines, which are tile joints with a high possibility at sampling point tT. It is clear that $\mathcal{I}_{\theta,t} \neq \emptyset$; otherwise, the robot heading correction cannot be conducted because no tile joint has been detected. Furthermore, we have

$$\theta_t \in \Theta_t^{(c)} = \bigcap_{i \in \mathcal{I}_{\theta,t}} \Theta_{t,i}^{(c)}, \tag{21}$$

where $\Theta_t^{(c)}$ denotes the interval of floor visual heading measurement at sampling point tT.

Finally, the robot heading correction can be achieved by calculating

$$\widehat{\Theta}_t = \Theta_t^{(o)} \cap \Theta_t^{(c)}, \tag{22}$$

where $\widehat{\Theta}_t$ denotes the interval of the heading estimation of FVO.

Table 1 Transformation rules from θ, ϑ, and r to x and y

Case	Rule
Case 1: $\theta \in (0, \frac{\pi}{2}]$	$x \in \{E_x i + d\|_{\vartheta \geqslant 0} + L\cos\theta : i = 0, 1, 2, \ldots\}$ $y \in \{E_y i - d\|_{\vartheta < 0} + L\sin\theta : i = 0, 1, 2, \ldots\}$
Case 2: $\theta \in (\frac{\pi}{2}, \pi]$	$x \in \{E_x i + d\|_{\vartheta < 0} + L\cos\theta : i = 0, 1, 2, \ldots\}$ $y \in \{E_y i + d\|_{\vartheta \geqslant 0} + L\sin\theta : i = 0, 1, 2, \ldots\}$
Case 3: $\theta \in (\pi, \frac{3\pi}{2}]$	$x \in \{E_x i - d\|_{\vartheta \geqslant 0} + L\cos\theta : i = 0, 1, 2, \ldots\}$ $y \in \{E_y i + d\|_{\vartheta < 0} + L\sin\theta : i = 0, 1, 2, \ldots\}$
Case 4: $\theta \in (\frac{3\pi}{2}, 2\pi]$	$x \in \{E_x i - d\|_{\vartheta < 0} + L\cos\theta : i = 0, 1, 2, \ldots\}$ $y \in \{E_y i - d\|_{\vartheta \geqslant 0} + L\sin\theta : i = 0, 1, 2, \ldots\}$

3.3 Robot location correction

First, the transformation from θ, ϑ, and r to x_c and y_c should be established, where (x_c, y_c) is the coordinate of the camera lens center with respect to the WCS. Taking Case 1, where $\theta \in (0, \frac{\pi}{2}]$, as an example, it is easy to determine that the tile-joint line is parallel to the X-axis if $\vartheta \in [-\frac{\pi}{2}, 0)$ or to the Y-axis if $\vartheta \in [0, \frac{\pi}{2})$, as illustrated in Figure 2. Furthermore, we denote

$$d = \frac{\frac{\mu}{2}\cos\vartheta + \frac{\nu}{2}\sin\vartheta - r}{\kappa\sqrt{\cos^2\vartheta + \sin^2\vartheta}} = \frac{\mu\cos\vartheta + \nu\sin\vartheta - 2r}{2\kappa}, \tag{23}$$

where $|d|$ is the perpendicular distance from the camera lens center to the tile-joint line with respect to the WCS and κ denotes the scaling factor from the PCS to the WCS. Define O_c as the camera lens center with respect to the PCS. In the case of a line where $\vartheta \in [-\frac{\pi}{2}, 0)$, it is clear that O_c lies in the upper area of the PCS divided by the line parallel to the X-axis if $d < 0$, or in the lower area if $d > 0$. In the case of a line where $\vartheta \in [0, \frac{\pi}{2})$, it is clear that O_c lies in the left area of PCS divided by the line parallel to the Y-axis if $d < 0$, or in the right area if $d > 0$. The exact coordinates of the tile-joint lines are unknown in advance, but they must be multiples of the tile-edge length. Hence, the coordinate of the camera lens center with respect to the WCS under Case 1 is

$$x_c \in \{E_x i + d|_{\vartheta \geqslant 0} : i = 0, 1, 2, \ldots\}, \tag{24a}$$

$$y_c \in \{E_y i - d|_{\vartheta < 0} : i = 0, 1, 2, \ldots\}, \tag{24b}$$

where E_x and E_y denote the lengths of the tile-edges parallel to the X- and Y-axis, respectively.

Substituting $\beta = \pi$ into (1a) and (1b) yields

$$x_c = x - L\cos\theta, \tag{25a}$$

$$y_c = y - L\sin\theta, \tag{25b}$$

where L denotes the distance from the robot's axle midpoint to the camera's lens center. Therefore, the coordinate of the robot location with respect to the WCS is derived, that is,

$$x \in \{E_x i + d|_{\vartheta \geqslant 0} + L\cos\theta : i = 0, 1, 2, \ldots\}, \tag{26a}$$

$$y \in \{E_y i - d|_{\vartheta < 0} + L\sin\theta : i = 0, 1, 2, \ldots\}. \tag{26b}$$

The two equations above show the transformation from θ, ϑ, and r to x and y in Case 1. Similarly, the transformation rules under Cases 2–4 can be obtained, as summarized in Table 1.

Now, we are in a position to calculate the location measurements obtained by the camera. For Case 1 where $\widehat{\Theta}_t \subseteq (0, \frac{\pi}{2}]$, we have

$$\mathcal{I}_{x,t} = \left\{i : \max\left\{\Phi_{t,i}^{(c)}\right\} < 0, i \in \mathcal{I}_{\theta,t}\right\}, \tag{27a}$$

$$\mathcal{I}_{y,t} = \left\{i : \min\left\{\Phi_{t,i}^{(c)}\right\} \geqslant 0, i \in \mathcal{I}_{\theta,t}\right\}, \tag{27b}$$

Table 2 Calculation of the interval of floor visual location measurement

Case	Intervals
Case 1: $\widehat{\Theta}_t \subseteq (0, \frac{\pi}{2}]$	$\mathcal{X}_t^{(c)} = \left\{E_x i + \bigcap_{j\in\mathcal{I}_{y,t}} \mathcal{D}_{t,j}^{(c)} + L\cos\widehat{\Theta}_t : i = 0,1,2,\dots\right\}$, where $\mathcal{I}_{y,t} = \left\{i : \min\left\{\Phi_{t,i}^{(c)}\right\} \geqslant 0, i \in \mathcal{I}_{\theta,t}\right\}$ $\mathcal{Y}_t^{(c)} = \left\{E_y i - \bigcap_{j\in\mathcal{I}_{x,t}} \mathcal{D}_{t,j}^{(c)} + L\sin\widehat{\Theta}_t : i = 0,1,2,\dots\right\}$, where $\mathcal{I}_{x,t} = \left\{i : \max\left\{\Phi_{t,i}^{(c)}\right\} < 0, i \in \mathcal{I}_{\theta,t}\right\}$
Case 2: $\widehat{\Theta}_t \subseteq (\frac{\pi}{2}, \pi]$	$\mathcal{X}_t^{(c)} = \left\{E_x i + \bigcap_{j\in\mathcal{I}_{y,t}} \mathcal{D}_{t,j}^{(c)} + L\cos\widehat{\Theta}_t : i = 0,1,2,\dots\right\}$, where $\mathcal{I}_{y,t} = \left\{i : \max\left\{\Phi_{t,i}^{(c)}\right\} < 0, i \in \mathcal{I}_{\theta,t}\right\}$ $\mathcal{Y}_t^{(c)} = \left\{E_y i + \bigcap_{j\in\mathcal{I}_{x,t}} \mathcal{D}_{t,j}^{(c)} + L\sin\widehat{\Theta}_t : i = 0,1,2,\dots\right\}$, where $\mathcal{I}_{x,t} = \left\{i : \min\left\{\Phi_{t,i}^{(c)}\right\} \geqslant 0, i \in \mathcal{I}_{\theta,t}\right\}$
Case 3: $\widehat{\Theta}_t \subseteq (\pi, \frac{3\pi}{2}]$	$\mathcal{X}_t^{(c)} = \left\{E_x i - \bigcap_{j\in\mathcal{I}_{y,t}} \mathcal{D}_{t,j}^{(c)} + L\cos\widehat{\Theta}_t : i = 0,1,2,\dots\right\}$, where $\mathcal{I}_{y,t} = \left\{i : \min\left\{\Phi_{t,i}^{(c)}\right\} \geqslant 0, i \in \mathcal{I}_{\theta,t}\right\}$ $\mathcal{Y}_t^{(c)} = \left\{E_y i + \bigcap_{j\in\mathcal{I}_{x,t}} \mathcal{D}_{t,j}^{(c)} + L\sin\widehat{\Theta}_t : i = 0,1,2,\dots\right\}$, where $\mathcal{I}_{x,t} = \left\{i : \max\left\{\Phi_{t,i}^{(c)}\right\} < 0, i \in \mathcal{I}_{\theta,t}\right\}$
Case 4: $\widehat{\Theta}_t \subseteq (\frac{3\pi}{2}, 2\pi]$	$\mathcal{X}_t^{(c)} = \left\{E_x i - \bigcap_{j\in\mathcal{I}_{y,t}} \mathcal{D}_{t,j}^{(c)} + L\cos\widehat{\Theta}_t : i = 0,1,2,\dots\right\}$, where $\mathcal{I}_{y,t} = \left\{i : \max\left\{\Phi_{t,i}^{(c)}\right\} < 0, i \in \mathcal{I}_{\theta,t}\right\}$ $\mathcal{Y}_t^{(c)} = \left\{E_y i - \bigcap_{j\in\mathcal{I}_{x,t}} \mathcal{D}_{t,j}^{(c)} + L\sin\widehat{\Theta}_t : i = 0,1,2,\dots\right\}$, where $\mathcal{I}_{x,t} = \left\{i : \min\left\{\Phi_{t,i}^{(c)}\right\} \geqslant 0, i \in \mathcal{I}_{\theta,t}\right\}$

where $\mathcal{I}_{x,t}$ and $\mathcal{I}_{y,t}$ denote the sequence number set of detected lines that are parallel to the X-axis and Y-axis, respectively. It is clear that $\mathcal{I}_{x,t} \neq \emptyset$ and $\mathcal{I}_{y,t} \neq \emptyset$; otherwise, the robot location correction cannot be conducted. Let $d_{t,i}$ denote d of the i-th line at sampling point tT that lies in the interval $\mathcal{D}_{t,i}^{(c)}$ measured by the camera. It can be obtained that

$$d_{t,i} \in \mathcal{D}_{t,i}^{(c)} = \frac{\mu\cos\Phi_{t,i}^{(c)} + \nu\sin\Phi_{t,i}^{(c)} - 2\mathcal{R}_{t,i}^{(c)}}{2\kappa}, \tag{28}$$

where $\mathcal{R}_{t,i}^{(c)}$ denotes the interval in which $r_{t,i}$ lies. Consequently, we have

$$x_t \in \mathcal{X}_t^{(c)} = \left\{E_x i + \bigcap_{j\in\mathcal{I}_{y,t}} \mathcal{D}_{t,j}^{(c)} + L\cos\widehat{\Theta}_t : i = 0,1,2,\dots\right\}, \tag{29a}$$

$$y_t \in \mathcal{Y}_t^{(c)} = \left\{E_y i - \bigcap_{j\in\mathcal{I}_{x,t}} \mathcal{D}_{t,j}^{(c)} + L\sin\widehat{\Theta}_t : i = 0,1,2,\dots\right\}, \tag{29b}$$

where $\mathcal{X}_t^{(c)}$ and $\mathcal{Y}_t^{(c)}$ denote the intervals of the floor visual location measurement at sampling point tT.

Finally, the robot location correction can be achieved by calculating

$$\widehat{\mathcal{X}}_t = \mathcal{X}_t^{(o)} \cap \mathcal{X}_t^{(c)}, \tag{30a}$$

$$\widehat{\mathcal{Y}}_t = \mathcal{Y}_t^{(o)} \cap \mathcal{Y}_t^{(c)}, \tag{30b}$$

where $\widehat{\mathcal{X}}_t$ and $\widehat{\mathcal{Y}}_t$ denote the intervals of the location estimation of FVO.

In the other cases, where $\widehat{\Theta}_t \subseteq (\frac{\pi}{2}, \pi]$, $\widehat{\Theta}_t \subseteq (\pi, \frac{3\pi}{2}]$, and $\widehat{\Theta}_t \subseteq (\frac{3\pi}{2}, 2\pi]$, the corresponding $\mathcal{I}_{x,t}$, $\mathcal{I}_{y,t}$, $\mathcal{X}_t^{(c)}$, and $\mathcal{Y}_t^{(c)}$ can be obtained, as summarized in Table 2, using the analogous analytical method.

4 Experiment results and analysis

In order to observe the performance of the FVO method proposed in this paper, a real-world experiment was conducted based on the P3-DX wheeled mobile robot, which was equipped with a camera pointing vertically to the floor and a laptop on the robot top. The robot working area is limited to a rectangular space measuring 3.6 m × 2.4 m, which is covered with 30 cm × 30 cm rectangular non-patterned tiles. A camera having a field of view that is able to cover the robot working area is mounted on the wall. It is able to provide relatively accurate measurements of the robot's heading and location, which are treated as ground truths. The sampling period was 0.2 s and the actual moving speed was between 0 and 0.51 m/s. After the readings from the incremental encoders and camera were collected and sent to the laptop (3.2 GHz with 8 GB RAM), the procedure of FVO was performed using MATLAB.

To verify the effectiveness of the floor visual features extraction, one frame was selected to show the feature extracting process, as shown in Figure 3. We can see that the original image is blurred because

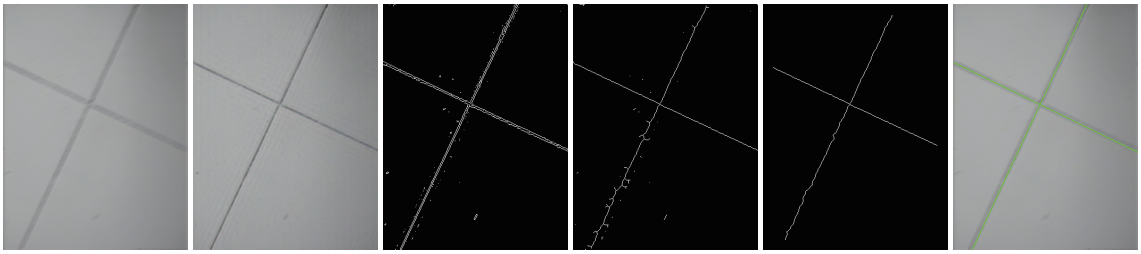

Figure 3 (Color online) Feature extracting process of floor vision. The green orthogonal lines in the sixth picture are the detected lines that almost coincide with the tile joints.

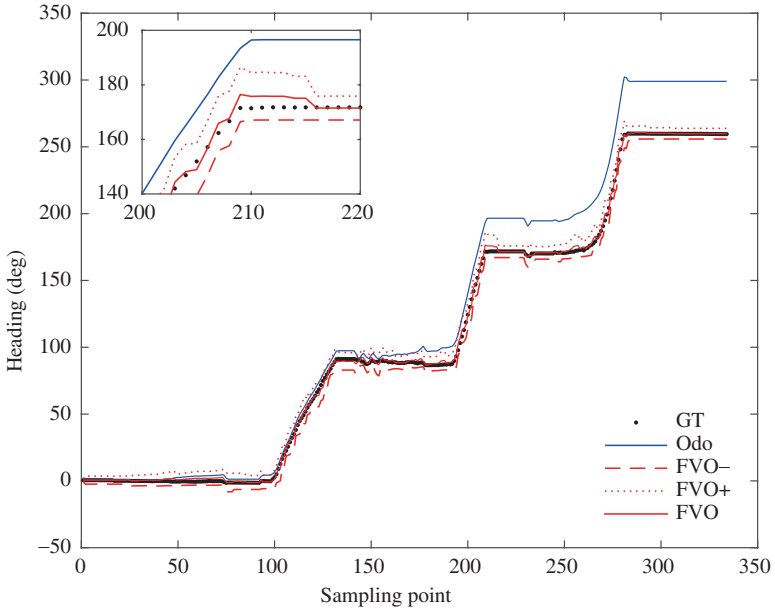

Figure 4 (Color online) Test results of heading estimation of FVO. The curve "GT" represents the ground truths of the robot heading obtained from the off-board camera. The curve "Odo" represents the odometric heading estimation. The curves "FVO", "FVO−", and "FVO+" represent the midpoint, lower bound, and upper bound of the heading estimation of FVO, respectively.

of the robot's motion, but after the deconvolution of the original images is calculated with a point spread function, the motion blur effects are almost eliminated and the tile joints become clear. By zooming in on the deblurred image, ringing effects can be found near the tile joints, because the high-frequency information was lost as a result of image degradation. They can be restrained by adjusting the point spread function. By applying Canny edge detector to the deblurred image, a binary image containing the edges of tile-joint lines is obtained. In the skeletonized image, there exist many branches connected or not connected to the skeleton of the tile joints, which affects the precision of the line detection to some extent. It can be seen that the skeletons of the tile joints usually possess a larger pixel amount than the other connected components. Hence, only the tile-joint lines are preserved after pruning, and their parameterized form is consequently precisely obtained.

Because FVO is not a localization method created by improving existing methods, but rather a localization method with a completely different sensory system architecture and data fusion algorithm, we could not perform a comparison experiment with the existing methods. Therefore, we present only the performance of FVO and its differences from that of odometry. The test results of FVO are shown in Figures 4 and 5. Because the off-board camera can provide millimeter-level measurements of the robot's heading and location, its outputs are treated as ground truths. It can be seen that the odometric heading estimation and odometric location estimation based on incremental encoders both gradually diverge from the ground truths as the moving distance of the mobile robot increases. This is the exact reason why rel-

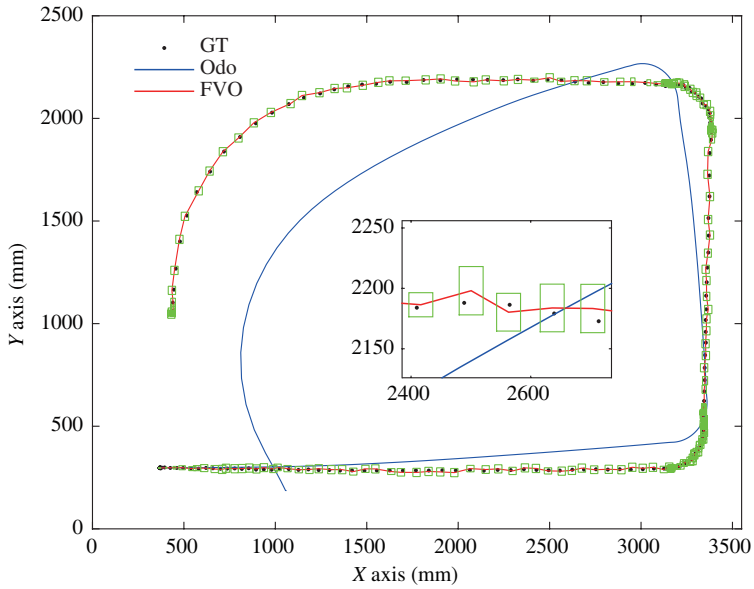

Figure 5 (Color online) Test results of location estimation of FVO. The curve "GT" represents the ground truths of the robot's location obtained from the off-board camera. The curve "Odo" represents the odometric location estimation. The green rectangular boxes represent the location estimation of FVO, while the curve "FVO" represents the midpoint of the location estimation of FVO.

Table 3 Error statistics of FVO

Variable	Method	1st period	2nd period	3rd period
θ (deg)	Odometry	1.93	7.16	19.65
	FVO	1.88	2.24	2.83
x (mm)	Odometry	34.25	74.28	259.56
	FVO	6.06	5.43	5.22
y (mm)	Odometry	47.80	66.22	373.17
	FVO	5.34	6.34	6.61

ative location methods should be combined with exteroceptive sensors. With the help of the floor visual correction, odometric errors can be significantly reduced and bounded. It is clear that all the ground truths are contained in the intervals or boxes of the FVO outputs. Taking the midpoints of the FVO outputs as the point estimations, the root mean squared error (RMSE) statistics can be obtained, as shown in Table 3. We averagely divide the data into three parts in chronological order, to yield three periods. Clearly, the accuracy of odometry decreases over time, while FVO retains a very stable millimeter-level accuracy. In fact, the accuracy of FVO can be further improved from two aspects. First, it is known that inevitable slippages occur while a mobile robot is turning, which leads to an imprecise kinematics model; however, gyroscopes can be used to increase the accuracy of odometry. Second, because time delays cannot be avoided in floor vision, we could use high-speed cameras or a compensation algorithm to increase the accuracy of the floor visual correction.

5 Conclusion

In this paper, we proposed a novel indoor localization method for mobile robots in which odometry and monocular vision are fused. It can be applied in many indoor scenarios, such as restaurants, laboratories, and supermarkets having floors covered with rectangular tiles. The method comprises three steps. First, the heading and location of the mobile robot are roughly estimated by odometry based on incremental encoders. Second, with the aid of a camera, the lens of which is pointed vertically toward the floor, the odometric heading estimation can be corrected by using the relative angle detected between the robot's

heading and the tile joints. Third, the odometric location estimation is corrected by using the detected perpendicular distance between the image center and the tile joints. As compared with the existing indoor localization methods, the proposed method, called FVO, has a lower economic cost and computational complexity, but is also characterized by relatively high accuracy and robustness.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. 61725304, 61673361). The authors also gratefully acknowledge the support from Youth Top-notch Talent Support Program, 1000-talent Youth Program and Youth Yangtze River Scholar.

References

1 Ye C L, Ma S G, Hui L. An omnidirectional mobile robot. Sci China Inf Sci, 2011, 54: 2631–2638
2 Lv W J, Kang Y, Qin J H. Indoor localization for skid-steering mobile robot by fusing encoder, gyroscope, and magnetometer. IEEE Trans Syst Man Cybern Syst, 2017. doi: 10.1109/TSMC.2017.2701353
3 Breuer T, Macedo G R G, Hartanto R, et al. Johnny: an autonomous service robot for domestic environments. J Intel Robot Syst, 2012, 66: 245–272
4 Siegwart R, Nourbakhsh I R, Scaramuzza D. Introduction to Autonomous Mobile Robots. 2nd ed. Cambridge: MIT Press, 2011
5 Li K, Ji H B. Inverse optimal adaptive backstepping control for spacecraft rendezvous on elliptical orbits. Int J Control, 2017, 7: 1–11
6 Chung H, Ojeda L, Borenstein J. Accurate mobile robot dead-reckoning with a precision-calibrated fiber-optic gyroscope. IEEE Trans Robot Autom, 2001, 17: 80–84
7 Kim J H, Lee J C. Dead-reckoning scheme for wheeled mobile robots moving on curved surfaces. J Intel Robot Syst, 2015, 79: 211–220
8 Reinstein M, Kubelka V, Zimmermann K. Terrain adaptive odometry for mobile skid-steer robots. In: Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, 2013. 4706–4711
9 Lee H, Jung J, Choi K, et al. Fuzzy-logic-assisted interacting multiple model (FLAIMM) for mobile robot localization. Robot Auton Syst, 2012, 60: 1592–1606
10 Borenstein J, Feng L. Gyrodometry: a new method for combining data from gyros and odometry in mobile robots. In: Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, 1996. 423–428
11 Myung H, Lee H K, Choi K, et al. Mobile robot localization with gyroscope and constrained Kalman filter. Int J Control Autom Syst, 2010, 8: 667–676
12 Georgy J, Noureldin A, Korenberg M J, et al. Modeling the stochastic drift of a MEMS-based gyroscope in Gyro/Odometer/GPS integrated navigation. IEEE Trans Intel Transp Syst, 2010, 11: 856–872
13 Garcia-Valverde T, Garcia-Sola A, Hagras H, et al. A fuzzy logic-based system for indoor localization using WiFi in ambient intelligent environments. IEEE Trans Fuzzy Syst, 2013, 21: 702–718
14 Yang P, Wu W Y. Efficient particle filter localization algorithm in dense passive RFID tag environment. IEEE Trans Ind Electron, 2014, 61: 5641–5651
15 Yasir M, Ho S W, Vellambi B N. Indoor positioning system using visible light and accelerometer. J Lightwave Technol, 2014, 32: 3306–3316
16 Jung J, Lee S M, Myung H. Indoor mobile robot localization and mapping based on ambient magnetic fields and aiding radio sources. IEEE Trans Instrum Meas, 2015, 64: 1922–1934
17 Hoermann S, Borges P V K. Vehicle localization and classification using off-board vision and 3-D models. IEEE Trans Robot, 2014, 30: 432–447
18 How J P, Behihke B, Frank A, et al. Real-time indoor autonomous vehicle test environment. IEEE Control Syst Mag, 2008, 28: 51–64
19 Panich S, Afzulpurkar N. Mobile robot integrated with gyroscope by using IKF. Int J Adv Robot Syst, 2011, 8: 122–136
20 Wang W S, Cao Q X, Zhu X X, et al. An automatic switching approach of robotic components for improving robot localization reliability in complicated environment. Ind Robot, 2014, 41: 135–144
21 Winterhalter W, Fleckenstein F, Steder B, et al. Accurate indoor localization for RGB-D smartphones and tablets given 2D floor plans. In: Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, Hamburg, 2015. 3138–3143
22 Marinho L B, Almeida J S, Souza J W M, et al. A novel mobile robot localization approach based on topological maps using classification with reject option in omnidirectional images. Expert Syst Appl, 2017, 72: 1–17
23 Xu D, Han L W, Tan M, et al. Ceiling-based visual positioning for an indoor mobile robot with monocular vision. IEEE Trans Ind Electron, 2009, 56: 1617–1628
24 Farrell J. Aided Navigation: GPS With High Rate Sensors. New York: McGraw-Hill, 2008

Appendix A Interval arithmetic

This section exhibits some fundamentals in interval arithmetic that may be involved in this paper.

An interval $\mathcal{A}=[a_1,a_2]$ is a set of real numbers denoted by

$$[a_1,a_2]=\{x\in\mathbb{R}:a_1\leqslant x\leqslant a_2\},\tag{A1}$$

where $a_1=-\infty$ and $a_2=+\infty$ are allowed, with $\mathrm{mid}(\mathcal{A})=\frac{a_1+a_2}{2}$ and $\mathrm{rad}(\mathcal{A})=\frac{a_2-a_1}{2}$ denoting its midpoint and radius, respectively.

The four basic arithmetic operations are as follows.

- Addition. $\mathcal{A}+\mathcal{B}=[a_1+b_1,a_2+b_2]$;
- Substraction. $\mathcal{A}-\mathcal{B}=[a_1-b_2,a_2-b_1]$;
- Multiplication. $\mathcal{A}\cdot\mathcal{B}=[\min\{a_1\cdot b_1,a_1\cdot b_2,a_2\cdot b_1,a_2\cdot b_2\},\ \max\{a_1\cdot b_1,a_1\cdot b_2,a_2\cdot b_1,a_2\cdot b_2\}]$;
- Division. $\mathcal{A}/\mathcal{B}=[a_1,a_2]\cdot[1/b_2,1/b_1]$ if $0\notin[b_1,b_2]$.

If $b_2<a_1$ or $a_2<b_1$, the intersection of two intervals $\mathcal{A}=[a_1,a_2]$ and $\mathcal{B}=[b_1,b_2]$ is empty, that is,

$$\mathcal{A}\cap\mathcal{B}=\emptyset.\tag{A2}$$

Otherwise, we have

$$\mathcal{A}\cap\mathcal{B}=\{x:x\in\mathcal{A}\text{ and }\mathcal{B}\}=[\max\{a_1,b_1\},\min\{a_2,b_2\}].\tag{A3}$$

The function of an interval $\mathcal{A}=[a_1,a_2]$ is defined by

$$f(\mathcal{A})=\{f(x):x\in\mathcal{A}\},\tag{A4}$$

which is still an interval. For the monotonic functions, such as an exponential or logarithmic function, we have

$$f(\mathcal{A})=[\min\{f(a_1),f(a_2)\},\max\{f(a_1),f(a_2)\}].\tag{A5}$$

For a sine function, a piecewise monotonic function with critical points at $n\pi+\frac{\pi}{2}$, where $n\in\mathbb{Z}$, we have

$$\sin(\mathcal{A})=\begin{cases}[\min\{\sin(a_1),\sin(a_2)\},\max\{\sin(a_1),\sin(a_2)\}], & \text{if } n_2-n_1=0,\\ [\min\{\sin(a_1),\sin(a_2)\},+1], & \text{if } n_2-n_1=1 \text{ and } n_1 \text{ is even},\\ [-1,\max\{\sin(a_1),\sin(a_2)\}], & \text{if } n_2-n_1=1 \text{ and } n_1 \text{ is odd},\\ [-1,+1], & \text{if } n_2-n_1\geqslant 2,\end{cases}\tag{A6}$$

where $n_1=\lfloor\frac{a_1+\frac{\pi}{2}}{\pi}\rfloor$ and $n_2=\lfloor\frac{a_2+\frac{\pi}{2}}{\pi}\rfloor$ are two integers. Similarly, we have

$$\cos(\mathcal{A})=\begin{cases}[\min\{\cos(a_1),\cos(a_2)\},\max\{\cos(a_1),\cos(a_2)\}], & \text{if } n_2-n_1=0,\\ [\min\{\cos(a_1),\cos(a_2)\},+1], & \text{if } n_2-n_1=1 \text{ and } n_1 \text{ is odd},\\ [-1,\max\{\cos(a_1),\cos(a_2)\}], & \text{if } n_2-n_1=1 \text{ and } n_1 \text{ is even},\\ [-1,+1], & \text{if } n_2-n_1\geqslant 2,\end{cases}\tag{A7}$$

where $n_1=\lfloor\frac{a_1}{\pi}\rfloor$ and $n_2=\lfloor\frac{a_2}{\pi}\rfloor$ are also two integers.