SLAM and Navigation of a Mobile Robot for Indoor Environments

Shuangshuang Lei and Zhijun Li

Abstract In this paper, a real-time implementation of simultaneous localization and mapping (SLAM) and navigation is described based on a mobile service robot platform, which consists of two driving wheels, a laser, and a Kinect. The main algorithm is extended Kalman filter (EKF) which is combined with feature extraction from laser scan data and extended beam curvature method for obstacle avoidance.

Keywords SLAM · Kalman filter · Obstacle avoidance

1 Introduction

Mobile service robots are very promising application of robotics. They should help people to deal with various tasks. The autonomous movement is the primary capability, so to let robot know where it is and how to achieve the goal would be fundamental problems of the mobile robot. To solve this problem, there are several simultaneous localization and mapping (SLAM) approaches, such as Kalman filter–based algorithm [1], Monte Carlo localization algorithm [2].

Monte Carlo localization algorithm is a probabilistic method which utilizes a samples set (particles) to approximate the probability density functions from a Bayesian perspective. This method had successfully introduced several indoor navigation applications. But the disadvantage is the high computational requirements.

Kalman filters are a efficient algorithm and widely introduced to solve the SLAM [3, 4]. However, the regular Kalman filter only can cope with linear

S. Lei · Z. Li (✉)
College of Automation Science and Engineering, South China University of Technology,
Guangzhou 510640, Guangdong, People's Republic of China
e-mail: zjli@ieee.org

F. Sun et al. (eds.), *Foundations and Applications of Intelligent Systems*,
Advances in Intelligent Systems and Computing 213,
DOI: 10.1007/978-3-642-37829-4_13, © Springer-Verlag Berlin Heidelberg 2014

problem. In Smith, Self, Cheeseman paper [5], they proposed the extended Kalman filter(EKF) to incrementally estimate the posterior distribution over robot pose along with the positions of the landmarks, through a Taylor expansion to linearize nonlinear system.

Obstacle avoidance is an another importance issue studied by researchers. Artificial potential field method [6], the neural network method [7], vector field histogram method (VFH) [8], the curvature velocity method (CVM) [9], the lane curvature method (LCM) [10], and the beam curvature method (BCM) [11] are typical techniques for obstacle avoidance. Among these techniques, beam curvature method is a simple and efficient method to find the promising forward path, which combined the LCM with CVM.

In this paper, we combine several methods to implement our algorithm. First, we utilize split–merge method to extract the essential features as the landmarks. Then, a EKF is used to solve the SLAM problem. Moreover, a modified BCM is adopted to avoid obstacles. At last, a kinematic control law is used to command the motion of the robot.

2 The Proposed Robot System

2.1 Mobile Service Robot

The main task of proposed system is to help operators to collect external environment information and build environment map in real time without collisions in the whole process. Our developed mobile robot platform is illustrated in Fig. 1. It equips two sensors, SICK LMS111 Laser Finder and Kinect, to grasp external information. The two wheels are driven by servo motors, and each servo motor is controlled by an ELMO driver which communicates with host through Kvaser CAN device.

3 Simultaneous Localization and Mapping

SLAM is a process which estimates the robot's pose and landmark's position at the same time, namely location and mapping. The fundamental step is to obtain external information to search proper features (landmarks). After landmarks' extraction, it is essential to match the observed landmarks with the existed landmarks in map. The heart of our SLAM process is correlating the mean and covariance matrix of system state through EKF technique.

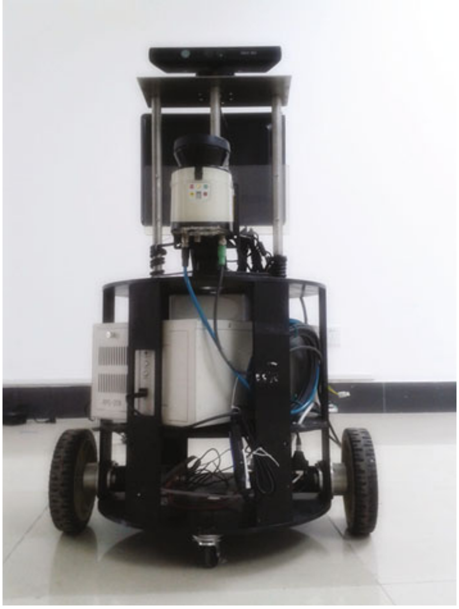


Fig. 1 The developed mobile service robot

3.1 Feature Extraction

For the indoor environment, there exist many line features; accordingly, we choose the corners as landmarks. There are several common methods to extract line features [12]. In this paper, we use split–merge method [13] to extract lines and compute the corners' position.

The basic idea of split–merge is to repeatedly search a point with the maximum distance d_p to a line, which is decided by the start and end points of a dot set, and compare d_p with the threshold $d_{\text{threshold}}$. If d_p is lesser than $d_{\text{threshold}}$, the point P is considered as the point in the line, otherwise divide the dot set into two new sets at point P. After split, we utilize least square method to fit corresponding lines.

We can consider the fitting problem as a regression problem under polar coordinate using the raw laser data. So we can describe line as

$$L : \rho = x \cos(\alpha) + y \sin(\alpha), \tag{1}$$

where ρ is the perpendicular distance from the origin to the line and α is the angle between the x axis and the normal of the line. So the mean-squared error of distance is obtained as follows:

$$\sum_{i=1}^{N} d_i^2 = \sum_{i=1}^{N} \left(\rho - (x_i cos(\alpha) + y_i sin(\alpha)\right)^2, \tag{2}$$

where d_i is the distance from point (x_i, y_i) to the line L. x_i and y_i, respectively, are the positions of samples in Cartesian coordinates. α and ρ can be computed as follows [14]

$$\tan(2\alpha) = \frac{2\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{N}\left[(x_i - \bar{x})^2 - (y_i - \bar{y})^2\right]}, \tag{3}$$

$$\rho = \bar{x}\cos\alpha + \bar{y}\sin\alpha, \tag{4}$$

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i. \tag{5}$$

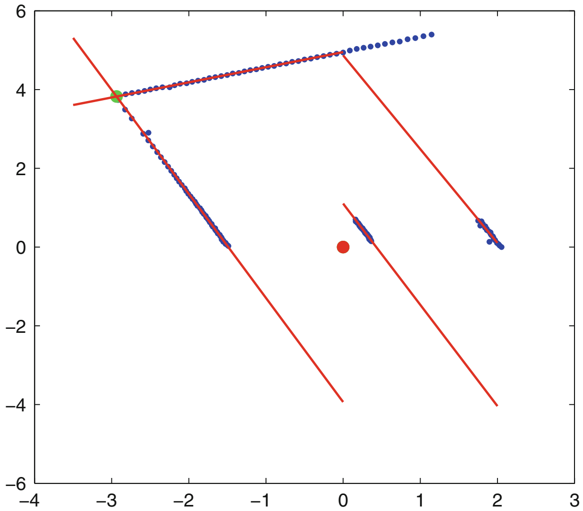
Combining with these two methods, we obtain the extracted line features. Through computing the cross dot of every two lines and comparing its position with the laser data, we can decide which cross dot could be corner. Moreover, we make some constraints, for example, a line should not be extracted if its length is under 1 meter or the number of contained dot is lesser than 10. The extracted results are shown in Figs. 2 and 3.

In Fig. 2, the blue dots are the raw laser data, the red circle represents robot, the red lines are extracted lines, and the green dot is the computed corner.

3.2 Data Association

In SLAM applications, when the robot detects a feature, it must be associated with a landmark in the existed map or incorporated as a new landmark. Data association

Fig. 2 The result without constraints



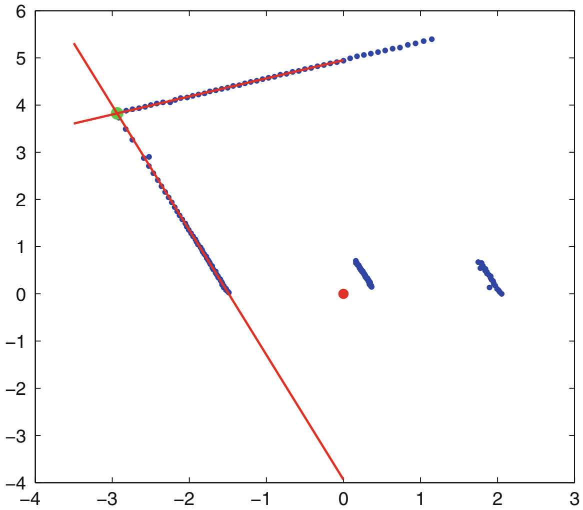


Fig. 3 The result with constraints

is the process that deals with the matching problem between observations and existed landmarks. A typical data association algorithm is composed of two elements: compatibility test to find potential pairs between observation and landmarks and a selection rule to choose the best matchings among the set of compatible matchings.

Generally, the gated nearest neighbor (NN) algorithm is used to associate data. The main process of NN algorithm is compatibility test through utilizing the Mahalanobis distance to determine whether a feature corresponds to a landmarks and then find the one with the minimum Mahalanobis distance. However, in this paper, due to that the landmarks are set exactly, we use a simple method to determine the correspondence between features and landmarks. We compute the degree of matching through a simple computation as follows:

$$\text{Match } (i,j) = |\text{ error }_r| \cos(\text{ error }_b), \tag{6}$$

where error _r and error _b are the error of range and bearing between feature i and landmark j, respectively. Then, we regard the feature with the minimum match as the best match of corresponding landmark.

3.3 Extended Kalman Filter

Extended Kalman filter is a repetitive process of prediction and correction to estimate the system state. In our implementation, we can estimate the pose of robot by observing several set landmarks.

$$x_n(k) = [x(k), y(k), \theta(k)]^T \tag{7}$$

where $x_n(k)$ is the posture of robot, $x(k)$ and $y(k)$ are the current robot positions, $\theta(k)$ is the heading angle, and k is the current time. The developed robot platform is a nonholonomic two-wheeled driven mobile robot, and its velocity motion model is shown in Fig. 4. In Fig. 4, D is the distance between two wheels. So velocity motion model can described as

$$g(x, u, k+1) = \begin{bmatrix} x(k) + \Delta t V(k)\cos(\theta(k)) \\ y(k) + \Delta t V(k)\sin(\theta(k)) \\ \theta(k) + \Delta t \omega(k) \end{bmatrix} \tag{8}$$

where $V(k)$ and $\omega(k)$ are the linear velocity and angular velocity, respectively.

To fit EKF, we should linearize the motion model to acquire the Jacobian matrices $\frac{\mathrm{d}g}{\mathrm{d}x}$ and $\frac{\mathrm{d}g}{\mathrm{d}u}$ as follows:

$$G_x = \frac{\mathrm{d}g}{\mathrm{d}x} = \begin{bmatrix} 1 & 0 & -\Delta t V(k)\sin(\theta(k)) \\ 0 & 1 & \Delta t V(k)\cos(\theta(k)) \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

$$G_u = \frac{\mathrm{d}g}{\mathrm{d}u} = \begin{bmatrix} \frac{R}{2}\Delta t\cos(\theta(k)) & \frac{R}{2}\Delta t\cos(\theta(k)) \\ \frac{R}{2}\Delta t\sin(\theta(k)) & \frac{R}{2}\Delta t\sin(\theta(k)) \\ \frac{R}{2D}\Delta t & -\frac{R}{2D}\Delta t \end{bmatrix}. \tag{10}$$

The global framework of robot is described as shown in Fig. 5.

Then, we can easily obtain the measurement model that shown as

$$h(x) = \begin{bmatrix} \rho \\ \alpha \end{bmatrix} = \begin{bmatrix} \sqrt{(x_l - x)^2 + (y_l - y)^2} \\ \arctan(\frac{y_l - y}{x_l - x}) - (\theta - \frac{\pi}{2}) \end{bmatrix}, \tag{11}$$

Fig. 4 The motion model

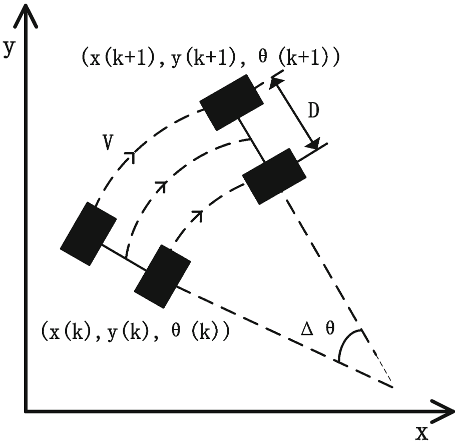
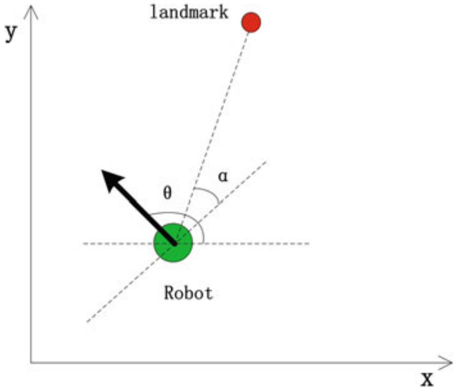


Fig. 5 The global framework of robot



where x_l, y_l are the Cartesian coordinates of landmarks, x and y are the positions of robot, and θ is the heading angle. Similarly, by linearizing the measurement model, we obtain the Jacobian matrix H_x of measurement model that shown as

$$H_x = \begin{bmatrix} \frac{x-x_l}{\rho} & \frac{y-y_l}{\rho} & 0 \\ \frac{y_l-y}{\rho^2} & \frac{x-x_l}{\rho^2} & -1 \end{bmatrix}. \tag{12}$$

3.4 Motion Plan and Obstacle Avoidance

Collision avoidance has been widely studied in autonomous mobile system, and several techniques has been proposed to tackle with collision avoidance. In this paper, we utilize a extended beam curvature method. The details are illustrated as follows:

Obtain the environmental information The laser's range is from 0° to 270° with 1° resolution. As a result, in every step, we obtain 271 values and every value d_i represents the distance of the ith beam.

Evaluate each beam To find the best beam, we evaluate the beams through a equation [11]:

$$f_e(d_i, p_i) = \alpha d_i \cos(|\, dp_i \,|) - (1 - \alpha) \left|\, \frac{dp_i}{\pi} \,\right|, \tag{13}$$

where p_i is defined as the angle between the X axis and the ith beam and p_G is defined as the angle between the X axis and the goal direction in the coordinates system which is illustrated in Fig. 6. $dp_i = p_G - p_i$, $d_i \cos(|\, dp_i \,|)$ may be regarded as the projected distance on the goal direction, and α and β are constants.

Searching the instant goal candidate and safe path In order to find the proper instant goal, it is essential to find the potential safe area. The safe area could be searched by [11]:

$$p_{1\text{safe}} = \max_{(P_i,d_i)} \left(P_i + \frac{d_{\text{safe}}}{d_i}\right), -\pi < P_i < P_{\text{best}}, \tag{14}$$

$$p_{2\text{safe}} = \min_{(P_i,d_i)} \left(P_i - \frac{d_{\text{safe}}}{d_i}\right), P_{\text{best}} < P_i < \pi, \tag{15}$$

where $p_{2\text{safe}}, p_{1\text{safe}}$ are the top angle and the bottom angle of the safe area and d_{safe} is a safe threshold adjusted in experiments. Then, we decide the heading angle through comparing the angle between robot's position and goal's position with the limits of safe area [11]:

$$\theta_{\text{heading}} = \begin{cases} p_0, & p_0 \in [p_{1\text{safe}}, p_{2\text{safe}}] \\ p_{1\text{safe}}, & p_0 < p_{1\text{safe}} \\ p_{2\text{safe}}, & p_0 > p_{2\text{safe}} \end{cases} \tag{16}$$

Determining velocity According to the instant goal and safe path, we determine angular velocity ω such that the heading angle of the robot is approaching the instant goal.

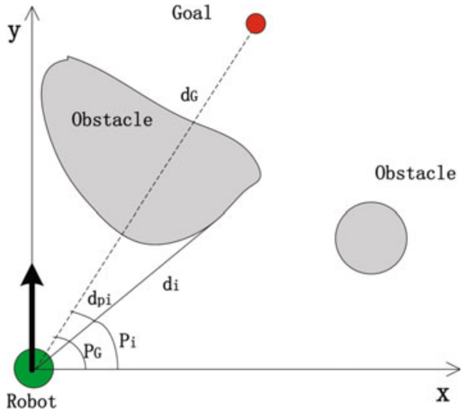
$$\omega = k_\omega(\theta_i - \theta_{\text{heading}}) \tag{17}$$

where k_ω is a adjustable constant, θ_i is the angle between the robot and the instant goal, and θ_{heading} is the heading angle of the robot. Moreover, in order to make sure that the robot will not deviate from the safe path suddenly, it is essential to limit the maximum linear speed υ_{MAX}. The linear velocity υ can be defined as follows:

$$\upsilon = \begin{cases} \upsilon_{\text{MAX}} - k_\upsilon \mid \omega \mid & \upsilon > 0 \\ 0 & \upsilon < 0 \end{cases} \tag{18}$$

$$\upsilon_r = \frac{2\upsilon + \omega D}{2} \tag{19}$$

Fig. 6 The beam appraise



$$\upsilon_l = \frac{2\upsilon - \omega D}{2}, \tag{20}$$

where υ is the linear velocity of the robot and υ_r and υ_l are the forward velocities of the right and left wheels.

Detecting goal We detect whether robot has arrived the goal position through the distance constraints

$$|x - x_{\text{goal}}| < \sigma_e, \tag{21}$$

$$|y - y_{\text{goal}}| < \sigma_e, \tag{22}$$

where x_{goal}, y_{goal} are the position of goal and σ_e is the error tolerance.

4 Experiments

In the experiments, the parameters of the robot are set as $R = 0.10$ m, $D = 0.6$ m, $k_\upsilon = 1.0$, $k_\omega = 0.4$, $\alpha = 0.05$, $\beta = 0.95$, and $\sigma_e = 0.2$ m, and the sampling interval is ΔT=0.1 s.

4.1 Estimate Start Position

Placing the robot at different initial positions, we begin to estimate the start position, and the result is shown in Table 1.

In the table, x, y are the practical positions, x_e, y_e are the estimated positions, and e_x, e_y are the relative errors. From the table, we can see that the results are acceptable for extended Kalman filter algorithm.

4.2 Obstacle Avoidance

The pulse number per circle is 2048, and if the control input is angular velocity, we should turn it into angular velocity. The laser's accuracy is 0.01 m and 1°.

Table 1 Estimated result

x	x_e	$e_x(\%)$	y	y_e	$e_y(\%)$
−5.4	−5.4865	1.601	0.92	0.9665	5.054
−5.4	−5.4738	1.367	1.20	1.2882	6.846
−4.8	−4.8729	1.518	1.20	1.2715	5.958
−4.8	−4.8830	1.729	0.60	0.6429	7.150

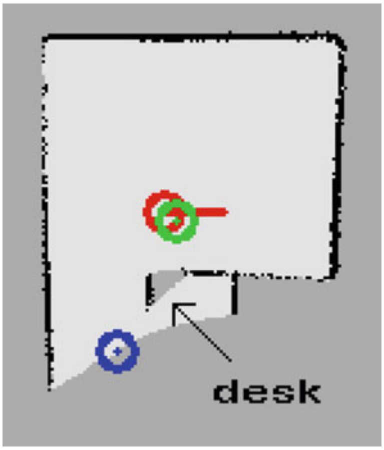


Fig. 7 The built map of SLAM

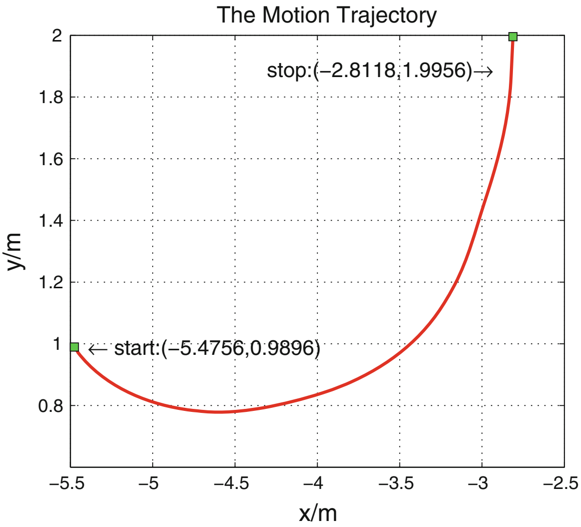


Fig. 8 The motion trajectory of robot

So we initialize the measurement noise covariance matrix Q and control noise covariance matrix R as follows:

$$R = \begin{bmatrix} (1.0/2048/\pi/\Delta T)^2 & 0 \\ 0 & (1.0/2048/\pi/\Delta T)^2 \end{bmatrix}, \; Q = \begin{bmatrix} (0.01)^2 & 0 \\ 0 & (1.0 * \pi/180)^2 \end{bmatrix}.$$

After initialized these matrices, we place the robot at $(-5.4, 1.0)$ and set the goal position at $(-3.0, 2.1)$ and then run our program. Finally, we obtain the map of our experimental environment as shown in Fig. 7.

In Fig. 7, the blue and green circles are the original position and the goal position, respectively. The red circle represents the robot position. The motion trajectory is shown in Fig. 8. During the overall process, robot moves toward left to acquire the enough diameter to bypass the desk on the right. Then, robot turns right to avoid the desk and moves to the goal. From the motion trajectory and the velocity trajectory, we can see that the whole process is smooth and continuous and the final errors are 0.1882 and 0.1044 m, which are within σ_e.

5 Conclusions

In this paper, a mobile system which integrates SLAM and obstacle avoidance is implemented. In SLAM, through comparing the detected features with landmarks which set previously in global coordinate system, we utilize EKF to estimate the pose of robot. Meanwhile, an obstacle avoidance algorithm is planning the motion path to ensure that the robot will safely arrive the goal position. In the implementation, the results are satisfying.

Acknowledgments This work is supported by the Natural Science Foundation of China under Grants 61174045 and 61111130208, and the International Science and Technology Cooperation Program of China under 2011DFA10950, and the Fundamental Research Funds for the Central Universities (No. 2011ZZ0104) and the Program for New Century Excellent Talents in University No. NCET-12-0195.

References

1. Kalman R (1960) A new approach to linear filtering and prediction problems. Trans ASME J Basic Eng 82:35–45
2. Dellaert F, Fox D, Burgard W, Thrun S (1999) Monte Carlo localization for mobile robots. International conference on robotics and automation, vol 2, pp 1322–1328
3. Hu HS (2000) Landmark based navigation of autonomous robots in industry. Int J of Ind Robot 27:458–467
4. Roumeliotis S, Bekey G (2000) Bayesian estimation and Kalman filtering: a unified framework for mobile robot localization. In: Proceedings of 2000 IEEE international conference on robotics and automation, vol 23, pp 2985–2992
5. Smith RC, Self M, Cheeseman P (1986) Estimating uncertain spatial relationships in robotics. In: Proceedings of the 2nd annual conference on uncertainty in, Artificial Intelligence (UAI-86), pp 267–288
6. Keron Y, Borenstein J (1991) Potential field methods and their inherent limitations for mobile robot navigation. In: Proceedings of the IEEE conference on robotics and automation, vol 2, pp 1398–1404
7. Glasius R, Komoda A, Gielen S (1995) Neural network dynamics for path planning and obstacle avoidance. Neural Networks 8:125–133
8. Ulrich I, Borenstein J (2000) VFH*: local obstacle avoidance with look-ahead verification. In: IEEE international conference on robotics and automation, vol 3, pp 2505–2511
9. Simmons R (1996) The curvature-velocity method for local obstacle avoidance. In: Proceedings of the 1996 IEEE international conference on robotics and automation Minneapolis, vol 4, pp 3375–3382

10. Ko N, Simmons G (1998) The lane-curvature method for local obstacle avoidance. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, vol 3, pp 1615–1621
11. Fernndez JL, Sanz R, Benayas JA, Diguez AR (2005) Improving collision avoidance for mobile robots in partially known environments: the beam curvature method. Rob Auton Syst 46:205–219
12. Nguyen V, Gächter S, Martinelli A, Tomatis N, Siegwart R (2007) A comparison of line extraction algorithms using 2D range data for indoor mobile robotics. Auton Robot 23:97–111
13. Pavlidis T, Horowitz SL (1974) Segmentation of plane curves. IEEE Trans Comput C–23:860–870
14. Kuo BW, Chang HH, Chen YC, Huang SY (2011) A light-and-fast SLAM algorithm for robots in indoor environments using line segmentMap. J Robot
15. Li Z, Xie Z, Ming A (2008) Simultaneously firing sonar ring based high-speed navigation for non-holonomic mobile robots in unstructured environment. Int J Veh Auton Syst 6:172–185
16. Zhu X, Li Z, Ye W, Zhao H (2001) Event-based Tele-presence of a mobile service robot. In: IEEE symposium on haptic audio visual environments and games, pp 118–123