

RGB-D SLAM using vanishing point and door plate information in corridor environment

Yonghoon Ji¹ · Atsushi Yamashita¹ · Hajime Asama¹

Received: 3 January 2015 / Accepted: 2 March 2015 / Published online: 22 March 2015
© Springer-Verlag Berlin Heidelberg 2015

Abstract This paper proposes a novel approach to an RGB-D simultaneous localization and mapping (SLAM) system that uses the vanishing point and door plates in a corridor environment simultaneously for navigation. To increase the stability of the SLAM process in such an environment, the vanishing point and door plates are utilized as landmarks for extended Kalman filter-based SLAM (i.e., EKF SLAM). The vanishing point is a semi-global unique feature usually observed in the corridor frontage, and a door plate has strong signature information (i.e., the room number) for the data association process. Thus, using these types of reliable features as landmarks maintains the stability of the SLAM process. A dense 3D map is represented by an octree structure that includes room number information, which can be useful semantic information. The experimental results showed that the proposed scheme yields a better performance than previous SLAM systems.

Keywords SLAM · RGB-D sensor · Vanishing point · Door plate · Mobile robot

1 Introduction

Mapping an unknown environment is a very important task for a mobile robot in the navigation field and includes such details as localization and path planning. To build a reliable map, the robot pose should also be accurately estimated at the same time. Various simultaneous localization and mapping (SLAM) schemes have been proposed to deal with this

problem [1]. Extracting robust features for matching is the key issue for estimating the robot pose accurately. Visual-based SLAM commonly uses low-level features such as corners or lines [2,3]. However, with low-level features, the data association process to match features can often fail because the features are indistinguishable from each other, which decreases the SLAM accuracy.

In contrast to low-level features, a global feature that can be used as a unique landmark in an environment makes it unnecessary to consider the above indistinguishable problem for data association. In addition, using features with easily recognizable differences eliminates ambiguity in the data association process, which leads to reliable SLAM.

In previous research on using global features, several SLAM schemes have been developed that use the vanishing point as a feature. Bosse et al. used the vanishing point extracted from an omni-directional camera image to group parallel line features and applied it as landmark information for SLAM [4]. Kawanishi et al. [5] also used the vanishing point to implement structure from motion (SfM) in a textureless environment. However, these methods can only be applied with omni-directional cameras and cannot make use of information from other sensors. Furthermore, these methods cannot build a dense map for mobile robot navigation. Lee et al. used the vanishing point extracted from laser range data to correct the rotational error of the robot pose; however, their approach cannot correct translational errors of the robot pose [6]. Zhang and Suh [7] developed a SLAM scheme to build a line-based map using not only the vanishing point, but also line features. However, their constructed line-based map does not include dense information on the environment, and errors in the data association process are inevitable because of the indistinguishable line features.

To overcome these disadvantages, we propose a novel approach for reliable SLAM that uses both the vanishing

✉ Yonghoon Ji
ji@robot.t.u-tokyo.ac.jp

¹ Graduate School of Engineering, The University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Fig. 1 Data from RGB-D sensor: **a** RGB image, **b** range image, and **c** point cloud

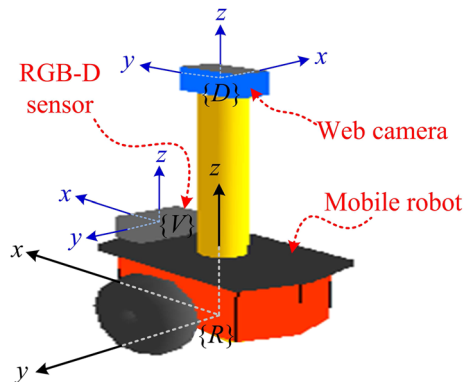
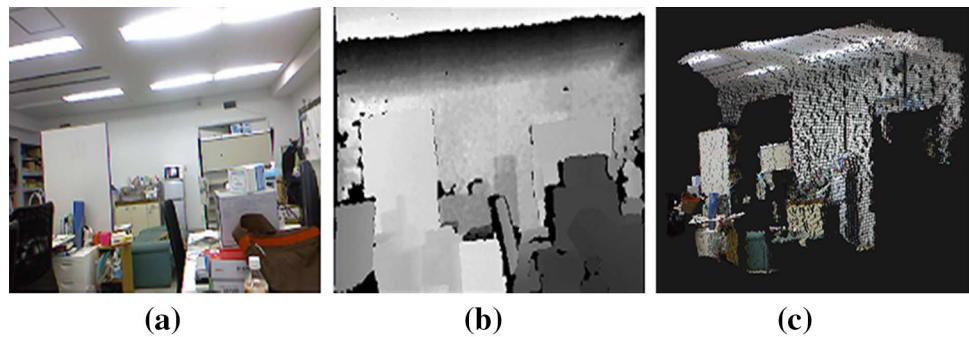


Fig. 2 Local coordinate system

point and door plates as features. The vanishing point is used to correct the rotational pose error, and the door plate is used to reduce the uncertainty of the translational pose error with respect to the longitudinal direction of the corridor. Because we can obtain a robust signature (i.e., the room number information) from the door plate, it is possible to clearly distinguish between features. This eliminates the ambiguity between features, so robust matching can be performed. Moreover, registering recognized room numbers in the map information allows a semantic map containing spatial information can be built.

In this study, an RGB-D sensor to easily acquire 3D information and Web camera to recognize door plates were used to generate a dense 3D semantic map for mobile robot navigation. To handle the nonlinearities of the system and estimate the robot state and feature states, extended Kalman filter-based SLAM (i.e., EKF SLAM) was applied. The RGB-D sensor can acquire an RGB image and depth image continuously, as shown in Fig. 1a, b. Processing these images allows a 3D point cloud data to be generated, as shown in Fig. 1c.

Figure 2 illustrates the coordinate system adopted for the experiments in this study: the robot coordinates $\{R\}$, RGB-D sensor coordinates $\{V\}$ to observe the vanishing point, and Web camera coordinates $\{D\}$ to recognize the door plates.

The remainder of this paper is organized as follows. Section 2 presents the method for extracting the vanishing point as the global feature, and Sect. 3 presents the method for

recognizing door plates with obvious signature information. Section 4 describes the EKF SLAM process for using the vanishing point and door plates as features. Section 5 describes how a 3D semantic map that includes both volumetric and feature information can be built. Section 6 presents the experimental results from a real environment. Finally, Sect. 7 presents the conclusions and future work.

2 Vanishing point extraction

A vanishing point is the intersection of projections from a set of parallel lines in space onto an image plane. Theoretically, this point exists at an infinite distance [8]. Several methods have been proposed to extract the vanishing point from an artificial environment [9]. The vanishing point can be a very useful feature in a corridor environment consisting of a long straight passage because this environment is characterized by converging to a unique vanishing point. Figure 3 shows the procedure for extracting the vanishing point in a corridor environment. First, a binary image is generated from the original image using the Canny edge detection process, as shown in Fig. 3a, b. Candidate lines that may converge to the vanishing point are then extracted through the Hough transform, as shown in Fig. 3c. The lines that do not converge to the vanishing point are eliminated using the condition that these lines are almost perpendicular to the (u, v) axis because of the characteristics of artificial structures. Finally, the unique vanishing point can be extracted by calculating the intersection of lines that are not eliminated. The equation of these lines is given by

$$a_i u + b_i v = c_i, \quad (1)$$

where $i = 1, 2, \dots, M$ and M is the number of the lines. (a_i, b_i, c_i) are coefficients of each equation of the lines. (u, v) represent the image coordinates. By substituting the coefficients into Eq. (1), the form of the matrix vector equation can be defined as follows:

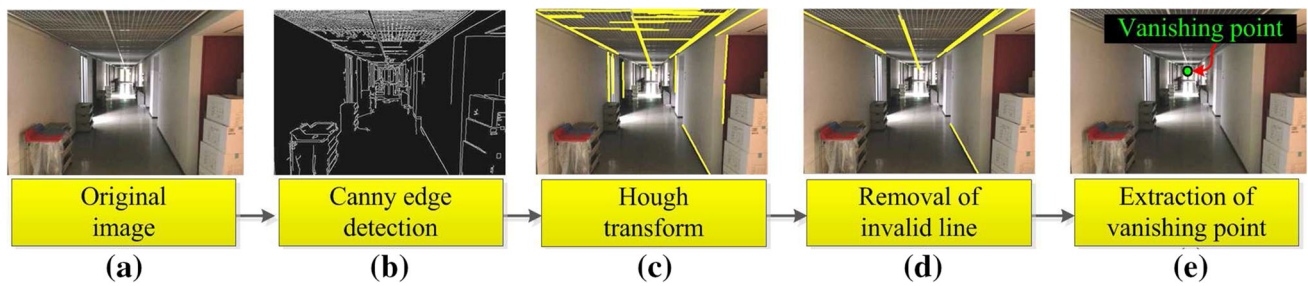


Fig. 3 Procedure for line-based vanishing point extraction

$$Au = c, \tag{2}$$

$$\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \\ \vdots & \vdots \\ a_M & b_M \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_M \end{bmatrix}. \tag{3}$$

Here, the estimated vanishing point $\hat{u} = (\hat{u}, \hat{v})$ is calculated with the least square methods using a Penrose pseudo-inverse matrix as follows:

$$\hat{u} = A^\dagger c, \tag{4}$$

where $A^\dagger = (A^T A)^{-1} A^T$. As shown in Fig. 3e, the unique vanishing point for the landmark is extracted with the above method.

3 Door plates recognition

Because door plates can frequently be observed in a corridor environment, they can provide very useful feature information. Reliable SLAM requires a data association process that tracks the same landmarks during a robot’s navigation. If we use accurately recognized door plates as landmarks, the data association process is greatly simplified because clear room number information is obtained. In particular, these features can handle translational pose errors that cannot be corrected when only the vanishing point is used as a feature. This section describes the method for door plate recognition.

The room number information on a door plate is recognized by using an optical character recognition (OCR)-based method using a support vector machine (SVM) and artificial neural network (ANN) [10]. This process is divided into two steps: SVM-based plate detection and ANN-based plate recognition.

A sufficient number of training samples is required to ensure reliable SLAM because a pre-training process is needed where the recognition rate strongly depends on the samples. The plate detection step involves detecting candidate image patches corresponding to the room number using image data from the camera mounted on a robot. The plate

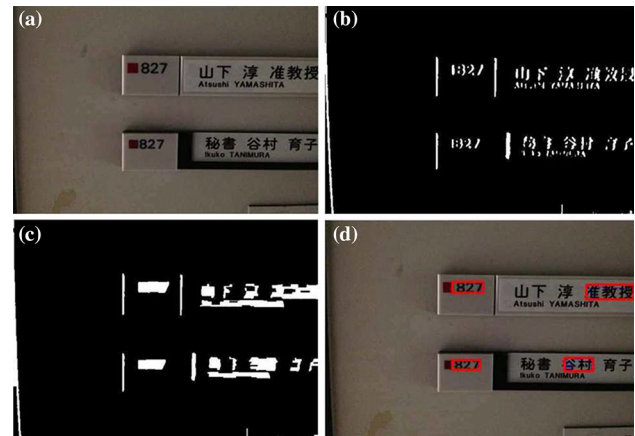


Fig. 4 Plate detection process: a original image, b after Sobel filter application, c after morphology closing filter application, and d detected candidate plates (rectangular blocks)

recognition step involves recognizing the image of the room number as a numeral.

3.1 Plate detection

The plate detection step involves detecting all candidate parts that correspond to the door plate in the input image. This task is also divided into two steps: segmentation and classification. During segmentation, an input image is divided into multiple segments to make feature extraction easier. First, we apply a Sobel filter and threshold filter in sequence to find vertical edges on the input image, as shown in Fig. 4b. This is because one important characteristic of door plate segmentation is the high number of vertical edges in letters. Second, applying a close morphology filter helps fill the black spaces between each vertical edge line and connect all regions with a high number of edges. After these processes are applied, regions in the image that may contain the room number information can be detected, as shown in Fig. 4c. Finally, a predefined aspect ratio (i.e., region width divided by region height) is applied to remove the improper regions, and candidate image patches are detected, as shown in Fig. 4d.

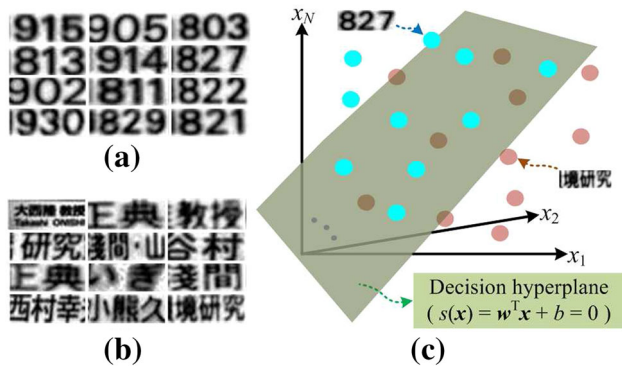


Fig. 5 Plate detection process: **a** training samples ($s(x) > 0$, label = 1), **b** training samples ($s(x) < 0$, label = -1), and **c** learned SVM (support vector machine)

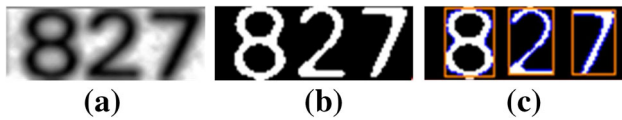


Fig. 6 Preprocessing for number recognition: **a** original image patch, **b** binary image patch, and **c** segmented objects (rectangular blocks)

The classification step determines whether or not each candidate patch has the room number information with the SVM, which is a widely used supervised machine learning algorithm for binary classification and has a relatively small number of parameters. Fig. 5a, b shows the labeled training samples that include and do not include the room number information. The linear SVM determines the hyperplane parameters w and b that are used to classify the data:

$$s(x) = w^T x + b = 0, \tag{5}$$

$$y = \text{sgn}(f(x)), \tag{6}$$

where $x = [x_1, \dots, x_k]^T$, $w = [w_1, \dots, w_k]^T$ and k represents the total number of pixels for each image patch. x is a feature vector that describes all of the pixels of the image patch. After the training of the SVM is completed, the image patches corresponding to the room number can clearly be distinguished in the feature space from all candidate image patches, as shown in Fig. 5c. Therefore, the labels of new input image patches are determined through the sign function in Eq. (6), and only image patches of the room number are detected.

3.2 Plate recognition

In the plate recognition step, OCR is used to determine the numeral from the image patches corresponding to the room number. An accurately recognized room number can act as a strong signature for the data association process in EKF SLAM. First, a binary image is generated through a threshold

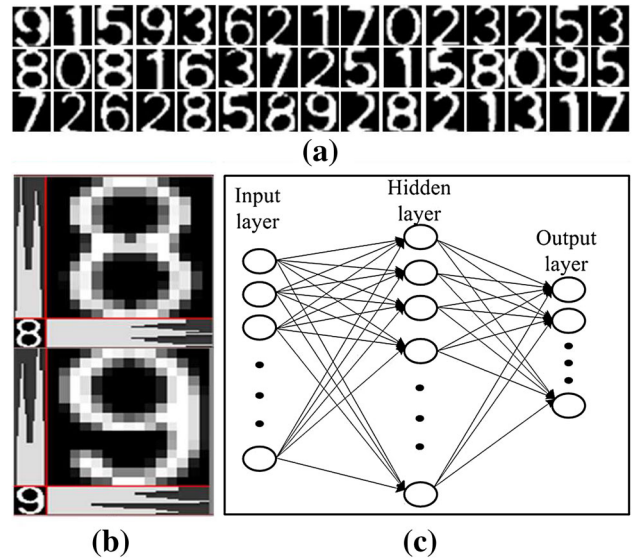


Fig. 7 Number recognition process: **a** training samples, **b** histogram features, and **c** ANN (artificial neural network)

filter for pre-processing, as shown in Fig. 6b. Second, a contour detection algorithm is performed, and each number information object is segmented as shown in Fig. 6c. After the pre-processing, each segmented number is separately recognized as 0–9 s through an ANN, which is another supervised machine learning algorithm. Figure 7a shows the training samples for ANN training. To train the ANN as shown in Fig. 7c, we use the features of horizontal and vertical accumulation histograms, as shown in Fig. 7b.

After the ANN is trained, all weight parameters of each layer are determined; therefore, if a new number object is an input, the corresponding number information object among 0–9 s can be recognized. In addition, a probability value (i.e., value similarity between histograms as shown in Fig. 7b) can be obtained for each digit, which can be used for the recognition score to determine the reliability. Figure 8 shows the final recognition results. If valid recognition is performed, the recognition score should be very close to 1.0. On the other hand, a recognition score from false recognition should be low (e.g., 0.7 or less). These false recognition results would not be registered as landmarks for SLAM. Because the unique value of a room number with a high recognition score can be used to clearly detect the location of each room, it can be used for robust signatures.

4 EKF SLAM process

Figure 9 shows a flowchart of the overall proposed SLAM process. The robot pose and landmark positions are estimated by EKF SLAM to handle nonlinearities related to odometry and measurement data (i.e., the vanishing point and door

Fig. 8 Recognized door plates: **a** correct recognition result (recognition scores: (0.9787, 0.9804, 0.9581) and (0.8518, 0.9804, 0.9425)) and **b** incorrect recognition result (recognition scores: (0.9682, 0.9856, 0.6180))

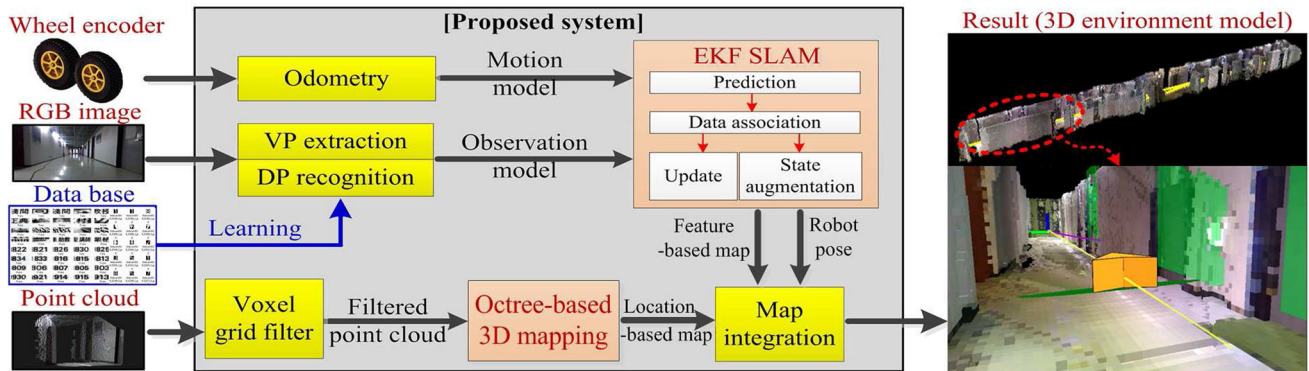


Fig. 9 Overview of the proposed EKF SLAM using vanishing point (VP) and door plates (DPs)

plates). An octree-based dense 3D map (i.e., location-based map) is built from the 3D point cloud data, and a feature-based map is generated from the estimated positions of the landmarks. Finally, a dense 3D semantic map is constructed by combining the information of both maps. Further details are given in Sect. 5.

The EKF SLAM procedure consists of the prediction and update steps. The control input data for odometry are used to define a motion model at the prediction step. The vanishing point and door plate information are used to define a measurement model in the update step. The state vector \mathbf{x} and corresponding covariance matrix \mathbf{P} in EKF SLAM are defined as follows:

$$\mathbf{x} = [(\mathbf{x}_r)^T \quad (\mathbf{l}_v)^T \quad (\mathbf{l}_{d(1:n)})^T]^T, \tag{7}$$

$$\mathbf{x}_r = [x_r \quad y_r \quad \phi_r]^T, \tag{8}$$

$$\mathbf{l}_v = [x_v \quad y_v]^T, \tag{9}$$

$$\mathbf{l}_{d(1:n)} = [(\mathbf{l}_{d1})^T \quad (\mathbf{l}_{d2})^T \quad \dots \quad (\mathbf{l}_{di})^T \quad \dots \quad (\mathbf{l}_{dn})^T]^T, \tag{10}$$

$$\mathbf{l}_{di} = [x_{di} \quad y_{di} \quad s_{di}]^T, \tag{11}$$

$$\mathbf{P} = \begin{bmatrix} \Phi & \Psi_{r,v} & \Psi_{r,d(1:n)} \\ (\Psi_{r,v})^T & \Theta_v & \Theta_{v,d(1:n)} \\ (\Psi_{r,d(1:n)})^T & (\Theta_{v,d(1:n)})^T & \Theta_{d(1:n)} \end{bmatrix}, \tag{12}$$

where \mathbf{x}_r is the robot pose and $\mathbf{l}_v, \mathbf{l}_{di}$ are the positions of the vanishing point and the i^{th} door plate with a room number

signature at time t , respectively, as shown in Fig. 10. Here, the subscript t representing the current state is omitted. The vanishing point feature represents a unique global landmark, while the door plate features do not. The diagonal elements Φ, Θ_v , and $\Theta_{d(1:n)}$ are covariance matrices corresponding to the states of the robot pose, vanishing point, and door plates, respectively. The off-diagonal elements $\Psi_{r,v}, \Psi_{r,d(1:n)}$, and $\Theta_{v,d(1:n)}$ are the cross-correlation matrices of Φ, Θ_v , and $\Theta_{d(1:n)}$. The state vector and covariance matrix are estimated in the prediction and update steps.

A more detailed EKF SLAM procedure is as follows. First, the system state \mathbf{x} is predicted based on the control input in the prediction step. Then, the data association process is performed based on the measurement data. If the data association processes succeed, the whole system state \mathbf{x} is updated in the update step. If the data association process fails, the whole system state \mathbf{x} is augmented. Further details are given in this section.

4.1 Prediction

In the prediction step of EKF SLAM, a motion model is defined to predict the state vector \mathbf{x}^- and corresponding covariance \mathbf{P}^- using the control input $\mathbf{u} = (s_l, s_r)$ at the time t as follows:

$$\mathbf{x}^- = \mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u})$$

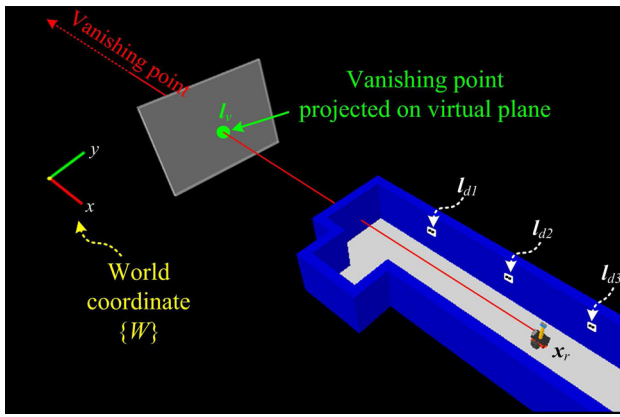


Fig. 10 State variables for EKF SLAM

$$= \mathbb{x}_{t-1} + \begin{bmatrix} \frac{s_r+s_l}{2} \cos(\phi_{r,t-1} + \frac{s_r-s_l}{2B}) \\ \frac{s_r+s_l}{2} \sin(\phi_{r,t-1} + \frac{s_r-s_l}{2B}) \\ \frac{s_r-s_l}{B} \\ \mathbf{0}_{N \times 1} \end{bmatrix}, \tag{13}$$

$$P^- = \mathbb{F}P_{t-1}\mathbb{F}^T + \mathbb{G}\mathbb{Q}\mathbb{G}^T, \tag{14}$$

$$\mathbb{F} = \begin{bmatrix} \mathbf{F} \mathbf{0}_{3 \times N} \\ \mathbf{0}_{N \times 3} \mathbf{I}_N \end{bmatrix}, \tag{15}$$

$$\mathbb{G} = \begin{bmatrix} \mathbf{G} \\ \mathbf{0}_{N \times 3} \end{bmatrix}, \tag{16}$$

$$\mathbb{Q} = \begin{bmatrix} k|s_r| & 0 \\ 0 & k|s_l| \end{bmatrix}, \tag{17}$$

where $f(\cdot)$ is the motion model of the system. This is defined by assuming a two-wheeled differential robot. $N = 3n + 2$ is the size of elements consisting of landmark states in the state vector \mathbb{x} . B denotes the distance between the wheels. \mathbb{Q} is the process noise matrix, and its elements consist of values that are proportional to the distances by the right wheel s_l and left wheel s_r of the control input. $\mathbf{F} = \partial f / \partial \mathbf{x}_{r,t-1}$ and $\mathbf{G} = \partial f / \partial \mathbf{u}$ are Jacobian matrices of the nonlinear function $f(\cdot)$ with respect to the robot state vector and control input, respectively. Note that the landmark state vectors are not affected by the control input because the vanishing point and door plates are considered to be fixed landmarks in the environment. The superscript $g-h$ indicates the predicted state before the measurement data at the time t are taken.

4.2 Data association

When a feature (e.g., vanishing point or door plate) is extracted, the data association process is needed to check whether it is a newly observed landmark or has already been registered. To perform this process, the measurement model $h(\cdot)$ based on the predicted state vector represents the

relationship between the global frame and sensor frame. This model is defined as follows:

$${}^r\hat{z}_i = h(\mathbf{x}_r^-, \mathbf{l}_i^-), \tag{18}$$

where ${}^r\hat{z}_i$ is the predicted measurement data of the i th landmark \mathbf{l}_i^- in the sensor frame at the time t from the predicted state vector \mathbf{x}_r^- . The measurement models for the vanishing point and door plates given in Sects. 2, 3 are defined as follows:

$${}^r\hat{z}_v = h_v(\mathbf{x}_r^-, \mathbf{l}_v^-) = \frac{f_v \{ (x_v^- - x_r^-) \sin \phi_r^- - (y_v^- - y_r^-) \cos \phi_r^- \}}{\sqrt{(x_v^- - x_r^-)^2 + (y_v^- - y_r^-)^2}}, \tag{19}$$

$${}^r\hat{z}_{di} = \begin{bmatrix} \hat{u}_{di} \\ \hat{s}_{di} \end{bmatrix} = h_d(\mathbf{x}_r^-, \mathbf{l}_{di}^-) = \begin{bmatrix} \frac{f_d \{ (x_{di}^- - x_r^-) \cos \phi_r^- - (y_{di}^- - y_r^-) \sin \phi_r^- \}}{\sqrt{(x_{di}^- - x_r^-)^2 + (y_{di}^- - y_r^-)^2}} \\ s_{di}^- \end{bmatrix}. \tag{20}$$

Here, Eq. (19) is the measurement model for the vanishing point extracted from the image from the RGB-D sensor. It plays an important role in correcting the rotational error of the robot pose because the vanishing point is observed almost continuously in the corridor environment. $\mathbf{l}_v^- = (x_v^-, y_v^-)$ denotes the state vector of the vanishing point, which has already been registered as a unique landmark. We assumed that a point at the finite distance for which the vanishing point is projected on a virtual plane is defined as the state of the landmark, as shown in Fig. 10. We assumed this because it is difficult to define a measured value as infinity. Equation (19) is the approximation model based on the assumption that the robot navigates along the corridor axis. The error of this measurement model can be ignored under the assumption that the distance between the robot and virtual plane is sufficiently large. This approximation model is valid because the robot moving along the corridor axis assigns the same meaning to observing the vanishing point. For these assumptions, we can consider the vanishing point as a global landmark fixed in space.

Equation (20) is the measurement model for the door plates extracted from the image from the camera mounted in the lateral direction. s_{di}^- denotes a signature that characterizes a landmark type. In this paper, we define the signature as an integer value of the recognized room number. Here, it is possible to greatly reduce the data association error by processing reliable measurement data. This can only occur if the recognition scores for all digits are above the threshold.

f_v and f_d are the focal lengths of the RGB-D sensor and Web camera, respectively. Figure 11a illustrates the image

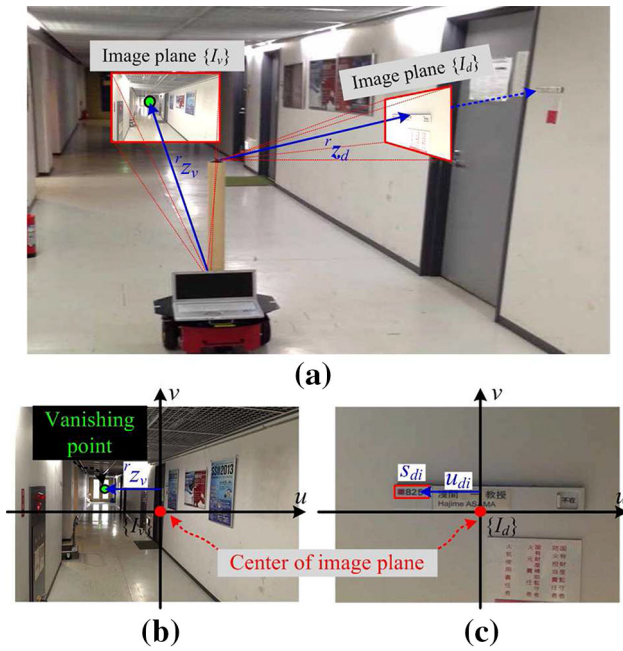


Fig. 11 Observation models for EKF SLAM: **a** local coordinate system, **b** image planes, **c** observation model of vanishing point, and **d** observation model of door plates

planes; these include the measured r_{z_v} corresponding to the vanishing point and r_{z_d} corresponding to the door plate. r_{z_v} represents the predicted measured value of the vanishing point based on the robot state vector x_r^- at the time t . It is defined as the pixel coordinate with respect to the u axis, as shown in Fig. 11b. r_{z_d} also represents the predicted measured value of the door plate based on the robot state vector. It is also defined in the same manner as the vanishing point, and the integer value of the room number signature is added as shown in Fig. 11c.

As noted in the introduction, because the measured value r_{z_v} is a unique landmark, it is not included in the data association process. Hence, state augmentation is performed only when a vanishing point is observed for the first time, and subsequent observations are always used for the state update process. On the other hand, for the measurements of the door plates, the simplified data association process is always performed using only the signature s_{d_i} , as given in Table 1. If the matching landmark l_{d_i} with the measured r_{z_d} exists (i.e., the data association succeeds) based on the algorithm given in Table 1, the whole system state is updated. On the other hand, if there is no matching landmark l_{d_i} after all landmarks are searched (i.e., the data association fails), the whole system state is augmented. Therefore, the data association process can simply be conducted by only checking the room number information of the door plates. Note that we only use recognition results with a high recognition score as measurement data; thus, it is possible to greatly reduce the matching error more than with a Mahalanobis distance-based method.

Table 1 Algorithm of simplified data association process

- 1: Data association for door plates landmarks (r_{z_d}):
- 2: for every registered landmark l_{d_i}
- 3: if s_d of $r_{z_d} = s_{d_i}$ of l_{d_i}
- 4: measured r_{z_d} is matched to i th landmark l_{d_i}
- 5: end if
- 6: end for

4.3 Update

In case of the data association success, the predicted state vector x and the corresponding covariance P at time t are updated from the measured r_{z_d} as follow:

$$K = P^{-1}H^T(H^{-1}P^{-1}H^T + R)^{-1}, \tag{21}$$

$$x = x^- + KH(z_d - \hat{z}_{d_i}), \tag{22}$$

$$P = (I_{N+3} - KH)P^-, \tag{23}$$

$$H = [H_r \ 0_{2 \times 3(i-1)} \ H_i \ 0_{2 \times 3(n-i)}] \tag{24}$$

$$R = \begin{bmatrix} \sigma_u^2 & 0 \\ 0 & 0 \end{bmatrix}, \tag{25}$$

where K represents the Kalman gain. $H_r = \partial h_d / \partial x_r^-$ and $H_i = \partial h_d / \partial l_{d_i}$ are the Jacobian matrices of the measurement models with respect to the robot state vector and landmark state vector, respectively. R denotes the measurement noise matrix. Here, the measurement of the room number information is clear and cannot affect the correction of the state vector. Thus, the corresponding noise element of R (i.e., the entry in the second row and second column of the matrix R) should be 0. Equations (21)–(23) represent the update process using the door plate information. The state update process using the vanishing point is also performed in the same manner based on the measured r_{z_v} and its model $h_v(\cdot)$.

4.4 State augmentation

If the data association fails, the state vector and the corresponding covariance at the time t are augmented with the first measurement. The augmented state vector x^+ and its covariance matrix P^+ are given by

$$x^+ = [x^T \ (l_a)^T]^T, \tag{26}$$

$$P^+ = \begin{bmatrix} \Phi & \Psi & \Phi^T(A_r)^T \\ \Psi^T & \Theta & \Psi^T(A_r)^T \\ A_r\Phi & A_r\Psi & A_r\Phi(A_r)^T + A_aR(A_a)^T \end{bmatrix}, \tag{27}$$

where A_r and A_a are the Jacobian matrices of the registered landmarks with respect to the robot state vector and measurement data. The new covariance matrix for the new landmark is

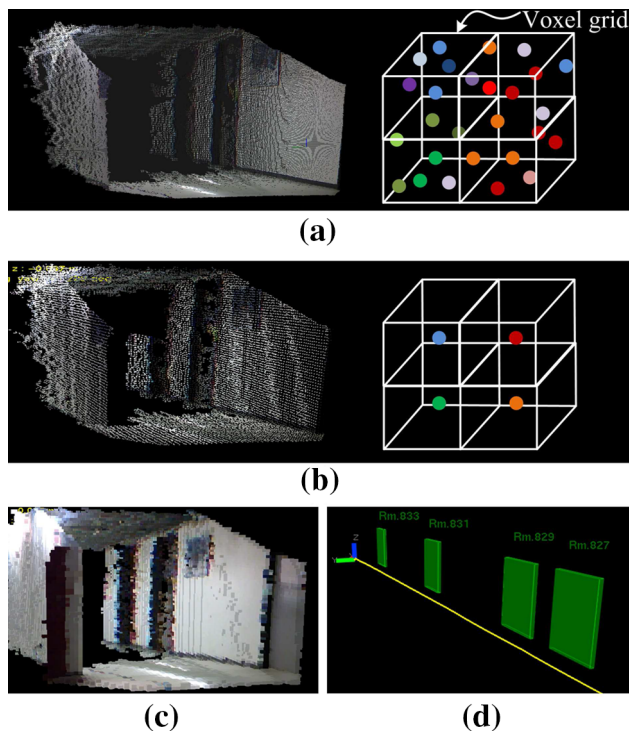


Fig. 12 3D semantic mapping: **a** original point cloud (size: about 300,000 points), **b** after voxel grid filter is applied (size: about 20,000 points), **c** location-based map for volumetric information (OctoMap), and **d** feature-based map for semantic information

thus augmented with all of the uncertainty factors considered (i.e., the covariance of the robot state Φ and the measurement noise R).

5 3D semantic map building

The estimated landmark states of the door plates can be useful semantic information in that they consist of absolute position values and the room number values. Therefore, these landmark states can be directly used for a feature-based map, as shown in Fig. 12d. This semantic information makes it possible to reduce the speed of the robot near the door when there is a risk of collision. Moreover, a goal position can easily be assigned using not specific coordinate information, but the room number information. Using only the feature-based map, however, makes mobile robot navigation impossible. For these reasons, a location-based map that represents the volumetric information was built and combined with the feature-based map in this study. Several ways to represent a 3D environment model have been developed [11–13]. Among them, the OctoMap library, which is based on the probabilistic update of the 3D point cloud, was utilized to build a dense 3D map [13, 14]. The data structure of the OctoMap is based on an octree structure that is very efficient for memory management. However, if we use all of

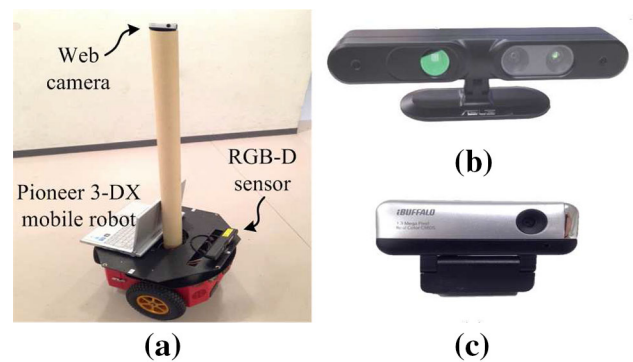


Fig. 13 Experimental setup: **a** whole system, **b** RGB-D sensor (ASUS Xtion Live Pro), and **c** Web camera

the points from the RGB-D sensor, as shown in Fig. 12a, the computational burden would be very high because 307,200 points would be considered in every frame of an image with a resolution of 640×480 pixels. To remedy this problem, a voxel grid filter is applied to reduce the number of points, as shown in Fig. 12b. Thus, the computational burden of the OctoMap building process using a 3D point cloud is effectively reduced. Figure 12c indicates the results of the built OctoMap.

6 Experimental results

To verify the proposed SLAM scheme, various experiments were conducted using a Pioneer 3-DX (MobileRobots) mobile robot equipped with an RGB-D sensor (ASUS Xtion Live Pro) and web camera, as shown in Fig. 13. The RGB-D sensor was mounted in the frontal direction; it was used to extract the vanishing point and build the location-based map (i.e., OctoMap). The web camera mounted in the lateral direction was used to recognize the door plates and build the feature-based map. The camera calibration task was done before the experiments. The average speed of the robot was 0.34 m/s, and a laptop computer with a 2.8 GHz quad core CPU was used to execute the proposed SLAM method. The acquisition period of the sensor data was 300 ms. The robot navigated toward the end of the corridor and came back to the starting point, as shown in Fig. 14a. Figures 14b, c illustrate the 3D semantic map, including both the feature-based map and location-based map, constructed with our proposed method.

To verify the effectiveness of the proposed method, comparative experiments were conducted. We conducted experiments under the same conditions using three different methods: odometry, a conventional corner-based EKF SLAM scheme using Harris corners, and the proposed EKF SLAM scheme using the vanishing point and door plates. The estimated trajectories and position errors are illustrated

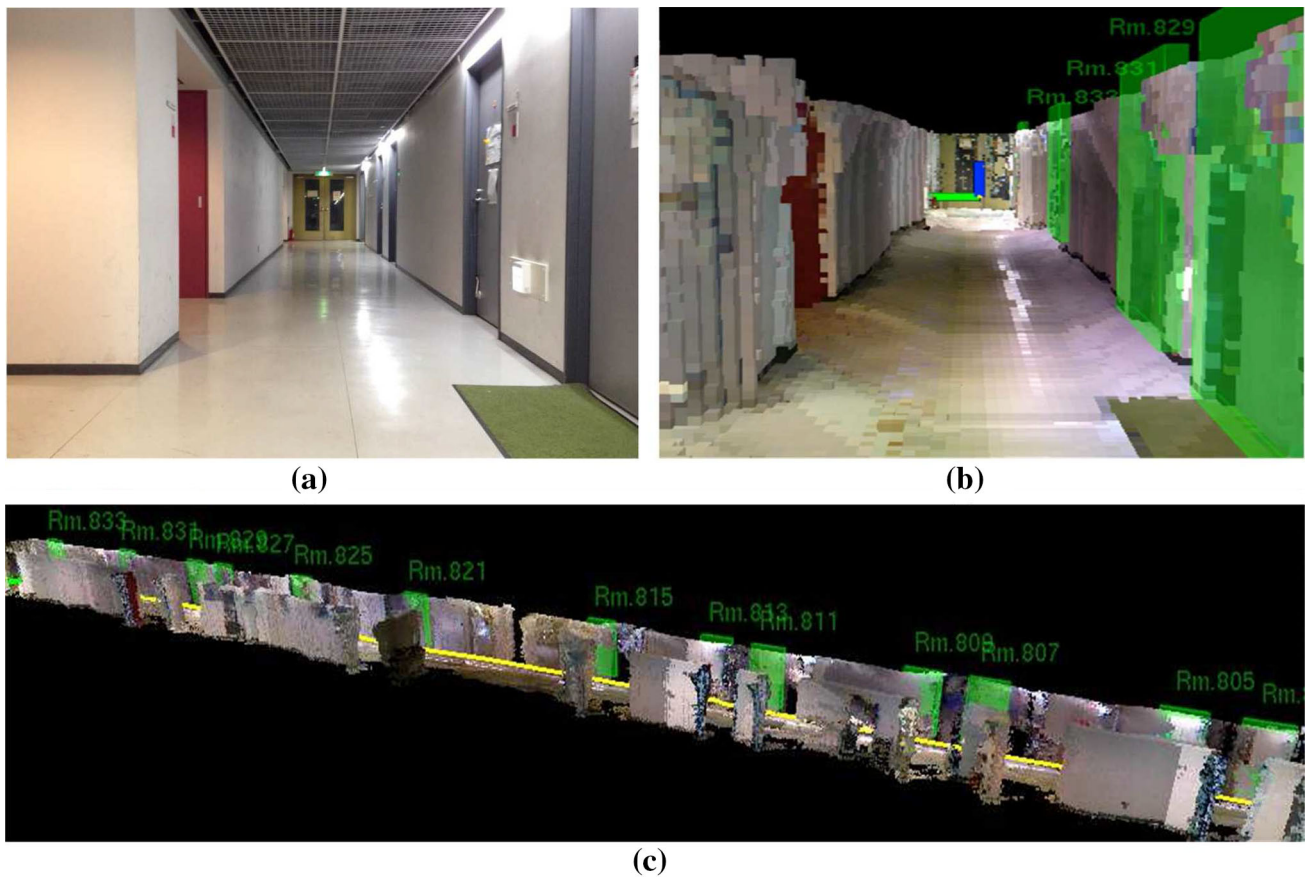


Fig. 14 SLAM results: **a** real corridor environment and **b, c** 3D semantic map, including feature-based map and location-based map

in Fig. 15a, b, respectively. The red dotted line, light solid line, and dark solid line represent the results with odometry, corner feature-based SLAM, and proposed SLAM, respectively. The odometry showed the largest error, even though the robot navigated along a linear path. For corner feature-based SLAM, the position error could not converge to close to 0 because of the indistinguishable features. The proposed method produced the most accurate estimated trajectory and smallest error.

Figure 16 represents the relationship between the number of features used as landmarks and the computation times of EKF SLAM for each frame. The conventional SLAM scheme, which used corner features, exploited 612 corners during the round-trip navigation in a corridor with a distance of 55 m; thus, it could not handle real-time processing, and the computation time rapidly increased after about 100 landmarks were registered. On the other hand, the proposed SLAM scheme used only one vanishing point and 14 door plates (i.e., 15 total) as landmarks; therefore, the matching process was very fast compared to the conventional SLAM scheme.

In conclusion, the corner feature-based SLAM registered a very large number of features as landmarks; thus, the

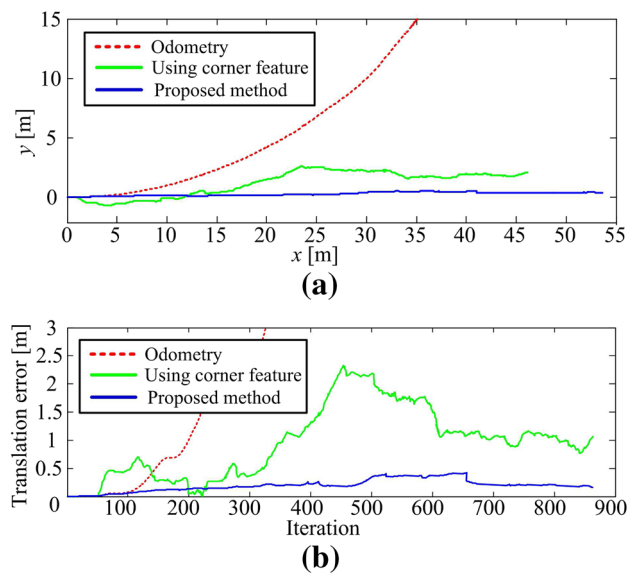


Fig. 15 Comparison of **a** estimated trajectories and **b** translation errors

computational load was significantly increased in proportion to the square with EKF SLAM. Furthermore, the accuracy of SLAM was not reliable. On the other hand, the proposed

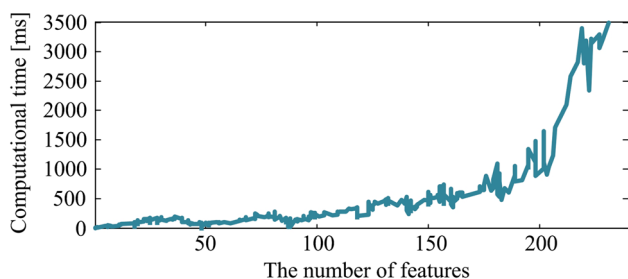


Fig. 16 Comparison of computational time

method built accurate map closely matching the real environment despite using a much smaller number of features as landmarks. The proposed method significantly reduced the computational burden.

7 Conclusion

This paper proposes a reliable SLAM scheme for a corridor environment that uses a vanishing point and door plates on the assumption that the robot navigates by the longitudinal direction of the corridor. The proposed approach was validated in several experiments, and the following conclusions were drawn:

- By using a vanishing point as a unique global landmark, the rotational error of the robot pose can be corrected robustly because the vanishing point can be observed almost continuously in a corridor environment.
- A door plate with room number information can serve as a very obvious signature; therefore, by using it as the landmark, the data association process in EKF SLAM can be simplified, and the translational error of the robot pose can be corrected robustly.
- A dense 3D semantic map can be constructed by combining the location-based map from the 3D point cloud and feature-based map representing the room number information.
- By combining the two above features, reliable SLAM can be performed for a corridor environment.

Future work will involve developing novel features with new measurement models for several types of semantic

information that contain more useful meanings. A semantic map with a variety of information other than the door plate information would be beneficial to various tasks performed by the robot in this environment.

References

1. Thrun S, Burgard W, Fox D (2005) Probabilistic robotics. The MIT Press, New York
2. Lemaire T, Lacroix S, Sola J (2005) A Practical 3D bearing-only SLAM Algorithm. In: Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 2449–2454
3. Smith P, Reid I, Davison A (2006) Real-time monocular SLAM with straight lines. In: Proceedings of the 17th British machine vision conference, pp 17–26
4. Bosse M, Rikoski R, Leonard J, Teller S (2003) Vanishing points and three-dimensional lines from omnidirectional video. *Vis Comput* 19(6):417–430
5. Kawanishi R, Yamashita A, Kaneko T, Asama H (2013) Parallel line-based structure from motion by using omnidirectional camera in textureless scene. *Adv Robot* 27(1):19–32
6. Lee YH, Nam C, Lee KY, Li YS, Yeon SY, Doh NL (2009) VPass: algorithmic compass using vanishing points in indoor environments. In: Proceedings of the 2009 IEEE/RSJ international conference on intelligent robots and systems, pp 936–941
7. Zhang G, Suh IH (2012) A vertical and floor line-based monocular SLAM system for corridor environments. *Int J Control Autom Syst* 10(3):547–557
8. Norling ER (2008) Perspective made easy. BN Publishing, New York
9. Shufelt JA (1999) Performance evaluation and analysis of vanishing point detection techniques. *IEEE Trans Pattern Anal Mach* 21(3):282–288
10. Baggio DL, Emami S, Escrivá DM, Ievgen K, Mahmood N (2012) Mastering openCV with practical computer vision projects. Packt Publishing, New York
11. Triebel R, Pfaff P, Burgard W (2006) Multi-level surface maps for outdoor terrain mapping and loop closing. In: Proceedings of the 2006 IEEE/RSJ international conference on intelligent robots and systems pp 2276–2282
12. Kim S, Kang J, Chung MJ (2013) Probabilistic voxel mapping using an adaptive confidence measure of stereo matching. *Intell Serv Robot* 6(2):89–99
13. Horunge A, Wurm KM, Bennewitz M, Stachniss C, Burgard W (2013) OctoMap: an efficient probabilistic 3D mapping framework based on Octrees. *Autonom Robots* 34(3):189–206
14. OctoMap. (<http://octomap.github.io/>)