An Efficient Ceiling-view SLAM Using Relational Constraints Between Landmarks

Regular Paper

Hyukdoo Choi¹, Ryunseok Kim¹ and Euntai Kim¹,*

1 School of Electrical and Electronic Engineering at Yonsei University
* Corresponding author E-mail: etkim@yonsei.ac.kr

Received 21 Apr 2013; Accepted 09 Oct 2013

DOI: 10.5772/57225

© 2014 The Author(s). Licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract In this paper, we present a new indoor ‘simultaneous localization and mapping’ (SLAM) technique based on an upward-looking ceiling camera. Adapted from our previous work [17], the proposed method employs sparsely-distributed line and point landmarks in an indoor environment to aid with data association and reduce extended Kalman filter computation as compared with earlier techniques. Further, the proposed method exploits geometric relationships between the two types of landmarks to provide added information about the environment. This geometric information is measured with an upward-looking ceiling camera and is used as a constraint in Kalman filtering. The performance of the proposed ceiling-view (CV) SLAM is demonstrated through simulations and experiments. The proposed method performs localization and mapping more accurately than those methods that use the two types of landmarks without taking into account their relative geometries.

Keywords Extended Kalman Filter, Relational Constraint, Ceiling-view, Visual Simultaneous Localization and Mapping (vSLAM)

1. Introduction

Since the inception of simultaneous localization and mapping (SLAM), a great deal of research has been conducted on indoor SLAM [1–5] using a variety of sensors [6–8], features [9, 10] and filters [2, 8, 10]. Thus, practical SLAM problems can be solved using many possible combinations of sensors and filtering methods. In our experience, a monocular camera combined with an extended Kalman filter (EKF) provides one of the most practical, low-cost solutions. A monocular camera is lightweight, cheap and low in power consumption in comparison to other sensors, such as laser scanners. An EKF is also a good option for indoor SLAM because the number of landmarks in an indoor environment, generally, is not too large.

An implementation of a monocular camera SLAM was demonstrated by Davison et al. in a study of MonoSLAM in [3] using a six DOF handheld camera. The camera extracted salient visual patches at corner points to register them as point landmarks. The camera’s motion was predicted by the constant velocity model, and the landmarks were initialized by inverse depth estimation [11] and tracked by EKF. Frintrop et al.
[12] proposed an active gaze-control system that tracked salient features using a pan-tilt camera. This system actively tracked existing features and prepared for loop closures in advance. Loop closures were carried out by matching SIFT [13] descriptors detected on salient features.

In other implementations, a monocular upward-looking camera was mounted on a mobile platform [10, 14-18]. Jeong et al. first proposed the ceiling-vision-based SLAM [14, 15] where corner points are detected by the Harris corner detector and where their orientations as well as their positions are estimated. Later, Hwang et al. developed a new monocular-vision-based SLAM [16], which detects corners and lamps on ceilings and doors from ceiling images and builds a map of multiple types of landmarks.

Ceiling images have the advantages that they are extremely static, rarely occluded and well-structured, but compared to front images in forward-looking cameras, they have more repetitive patterns and detect far fewer landmarks. Thus, ceiling-view (CV) SLAM is less time-demanding than forward-view (FV) SLAM, but CV-SLAM’s performance may be degraded due to the lack of landmarks. Therefore, the key issue in CV-SLAM is to extract as much information as possible from a small number of landmarks.

Extended Kalman filtering is a popular option for estimation in SLAM and is used in many applications. However, because EKF’s computational complexity is proportional to the square of the number of landmarks, for maps with large numbers of landmarks or for large target environments, the EKF-SLAM system is unable to operate in real-time. Thus, an EKF is a good choice only for a reasonable number of landmarks that are consistently detected.

In this paper, we present a new SLAM system using a monocular camera. The monocular camera looks upwards toward the ceiling of the environment and an EKF is employed to build a landmark-based map. In the proposed system, the monocular camera extracts line and point features on the ceiling, which are used as landmarks. The two types of landmarks are detected repeatedly and consistently for long periods of time, and their number is moderate in usual environments. The line landmarks are detected from ceiling boundaries [17], and the point landmarks are detected from the centre points of circles and lamps. However, the number of landmarks on the ceiling is likely to be insufficient for good performance, leading to the major challenge in CV-SLAM: to extract as much information as possible from a reasonable number of landmarks in order to build a map efficiently.

Thus, in this paper we propose a new and effective method to increase the amount of mapping information. In addition to the positions of the landmarks, we also employ the relationships among the landmarks to increase the accuracy of SLAM. More specifically, the observed distance between a point landmark and a line landmark becomes a geometric constraint that is used as an additional clue for the SLAM. In this way, the proposed system improves the SLAM accuracy by using the limited number of landmarks with their geometric constraints, while keeping the computational load low.

The remainder of this paper is organized as follows. The next section provides an overview of the system and its models. Our own contribution is presented in Section 3, which applies the geometric constraints to the SLAM framework. Section 4 describes the performance of the proposed method based on simulations and real-world experiments. Concluding remarks then are given in Section 5.

2. Mathematical Formulation of CV-SLAM

Let us suppose that a robot is equipped with an upward-looking camera, as shown in Figure 1. The camera scans the ceiling of the indoor environment and detects line and point landmarks on the ceiling. Thus, in this paper we propose a new and effective method to increase the amount of mapping information. In addition to the positions of the landmarks, we also employ the relationships among the landmarks to increase the accuracy of SLAM. More specifically, the observed distance between a point landmark and a line landmark becomes a geometric constraint that is used as an additional clue for the SLAM. In this way, the proposed system improves the SLAM accuracy by using the limited number of landmarks with their geometric constraints, while keeping the computational load low.

The remainder of this paper is organized as follows. The next section provides an overview of the system and its models. Our own contribution is presented in Section 3, which applies the geometric constraints to the SLAM framework. Section 4 describes the performance of the proposed method based on simulations and real-world experiments. Concluding remarks then are given in Section 5.

2. Mathematical Formulation of CV-SLAM

Let us suppose that a robot is equipped with an upward-looking camera, as shown in Figure 1. The camera scans the ceiling of the indoor environment and detects line and point landmarks on the ceiling.
For example, line landmarks include the boundaries of the ceiling while point landmarks include the lights and fire sensors. In this paper, both the line landmarks $L^*_{n}$ and point landmarks $P^*_{m}$ are included in a state $y$, with the robot pose $x$, as represented in Eq. (1), and the CV-SLAM is formulated as the statistical estimation of the augmented state:

$$y = \left[ x^T \left( L^*_1 \right)^T \left( L^*_2 \right)^T \cdots \left( P^*_1 \right)^T \left( P^*_2 \right)^T \cdots \right]^T$$ (1)

where the sub-index $t$ refers to the time index; $x = [x_1 \ y_1 \ b_1]$ denotes the x-y coordinates and bearing of the robot; $L^*_n = \left[ L^*_{n,1} \ L^*_{n,2} \right]$ denotes the $n$th line landmark, represented by a distance $d_{t}$ and an angle $b_{t}$ from the origin; $P^*_{m} = \left[ P^*_{m,1} \ P^*_{m,2} \right]$ denotes the position of the $m$th point landmark in the global x-y plane, as shown in Figure 1. In this paper, let us assume certain typical conditions such that the ceiling in the indoor environment is flat and its height is known as $h_{c}$. Since a monocular camera is a bearing-only sensor, this assumption greatly simplifies the three-dimensional estimation into a two-dimensional estimation and enables the immediate initialization of landmarks.

### 2.1 System Models

To estimate the metric state vector, we must define the system models. A two-wheeled robot’s motion always follows a straight or circular trajectory, which is modelled by a linear and an angular velocity. The motion model is defined as follows:

$$x = g(u, x_{i-1}) + N(0, R)$$

$$= x_{i-1} + \begin{bmatrix} \frac{v}{w} \sin b_{i-1} + \frac{v}{w} \sin(b_{i-1} + w \cdot \Delta t) \\ \frac{v}{w} \cos b_{i-1} - \frac{v}{w} \cos(b_{i-1} + w \cdot \Delta t) \end{bmatrix} + N(0, R)$$ (2)

where $u = [v \ w]^T$ represents the linear and angular velocity of a robot, and $R$ is the noise covariance of the motion model. Thus, for the augmented SLAM state $y = \left[ x^T \ L^*_1 \cdots P^*_1 \right]^T$, its transition equation becomes:

$$\begin{bmatrix} x \\ L^*_1 \\ P^*_1 \end{bmatrix} = \begin{bmatrix} g(u, x_{i-1}) \\ L^*_1 \end{bmatrix} + \begin{bmatrix} N(0, R) \\ 0 \\ 0 \end{bmatrix}$$ (3)

Now, let us think about the observation model. First, we will consider the line landmarks, which are detected on the ceiling boundaries between the ceiling and the wall. Figure 2 gives an example of the line-feature extraction process. A ceiling image is binarized into the ceiling region and the rest by expanding the ceiling region from the image centre, based on the fact that the centre of a ceiling image always belongs to the ceiling region. The ceiling boundary is divided into line segments and the segments are parameterized to obtain line measurements. The detailed extraction process is described in our previous work [17]. The line observation model transforms the distance and angle of a line landmark in the global frame into the robot’s frame, as follows:

$$z^i_{ld, t} = \begin{bmatrix} z^i_{ld, t} \\ z^i_{lb, t} \end{bmatrix} = v_{L} \left[ x, L^*_{n} \right] + N(0, Q_{L})$$

$$z_{ld, t} = \begin{bmatrix} z_{ld, t} \\ z_{lb, t} \end{bmatrix} = v_{P} \left[ x, P^*_{m} \right] + N(0, Q_{P})$$ (5)

where $z^i_{ld, t} = \begin{bmatrix} z^i_{ld, t} \\ z^i_{lb, t} \end{bmatrix}$ represents the relative distance and angle of the $n$th landmark from a robot, and $Q_{L}$ is the noise covariance matrix of the line observation model. The equation is divided into two cases so that the distance measurement is always positive.

Next, let us think about a point landmark and its observation model. The point observation model transforms the coordinates of point landmarks in the global frame into the robot’s frame, as follows:

$$z^i_{j} = \begin{bmatrix} z^i_{j} \\ z^i_{p, t} \end{bmatrix} = v_{P} \left[ x, P^*_{m} \right] + N(0, Q_{P})$$

$$= \begin{bmatrix} \sqrt{(P^*_{m, x} - x)^2 + (P^*_{m, y} - y)^2} \\ atan2(P^*_{m, y} - y, P^*_{m, x} - x) - b_{t} \end{bmatrix} + N(0, Q_{P})$$ (4)

**Figure 2.** Feature extraction process: original image (left), binarized image (centre), refined image, and ceiling boundary (right) [16]
are the relative distance and angle from the \(n\)th point landmark, respectively, which is marked by the small circle in Figure 1(b). The uncertainty of the observation is modelled as Gaussian noise with a covariance \(Q\).

The point landmarks consist of the centre points of electric lamps and circles on the ceiling. Electric lamps are detected by thresholding a ceiling image to extract bright blobs, and circles are detected by a combination of the Canny edge and Hough circle transforms. The point landmark detection methods are detailed in [18]. The feature extraction samples are presented in Figure 3. The three rows show how circular lights, light tubes and circles are detected, respectively, from top to bottom. The left column represents the original images and the right represents the detection results. The two top binary images show bright regions in which to find electric lamps and the bottom is the Canny edge image for circle detection. In the mid-central image, the right blob is detected from reflected light. This blob is falsified by checking whether the blob is inside the ceiling region. The bottom-central image shows four circular edges, but those circles that have a radius less than the threshold are abandoned, as shown in the bottom-left image.

2.2 EKF SLAM

In this subsection, the state defined in Eq. (1) is estimated by the EKF. In the EKF, all the states \(y_{i} = [x_{i}^{T} \ L_{i}^{T} \ P_{i}^{T}]^{T}\) are assumed to be Gaussian and their distributions are completely characterized by the mean vector \(\bar{y}_{i}\) and the covariance matrix \(C_{y_{i}}\):

\[
\bar{y}_{i} = \begin{bmatrix} x_{i} \\ L_{i} \\ P_{i} \end{bmatrix}
\]

\[
C_{y_{i}} = \begin{bmatrix} C_{xx} & C_{xl} & C_{xp} \\ C_{lx} & C_{ll} & C_{lp} \\ C_{px} & C_{pl} & C_{pp} \end{bmatrix}
\]

(7)

where \(C_{xx}\) is the covariance matrix of the variable \(x_{i}\), and \(C_{ab}\) is the correlation matrix between \(A\) and \(B\). The SLAM based on the EKF first predicts the robot’s pose using the odometry and then corrects the total state using the detected features. First, let us consider the prediction step. When the velocity \(u_{i} = [v_{i} \ w_{i}]^{T}\) is given, the EKF predicts the state by the following equations:

\[
\begin{bmatrix} \bar{x}_{i} \\ \bar{L}_{i} \\ \bar{P}_{i} \end{bmatrix} = \begin{bmatrix} g(u_{i}, \bar{x}_{i-1}) \\ L_{i-1} \\ P_{i-1} \end{bmatrix}
\]

\[
C_{y_{i}} = C_{y_{i-1}} + G_{i}^{T}\ R\ G_{i}
\]

(8)

\[
G_{i} = \frac{\partial g(u_{i}, \bar{x}_{i-1})}{\partial \bar{y}_{i}} = \begin{bmatrix} 1 & 0 & -\frac{v_{i}}{w_{i}} \cos h_{i} + \frac{v_{i}}{w_{i}} \cos (h_{i} + w_{i}) \\ 0 & 1 & 1 - \frac{v_{i}}{w_{i}} \sin h_{i} + \frac{v_{i}}{w_{i}} \sin (h_{i} + w_{i}) \theta_{s(v_{i} + w_{i})} \end{bmatrix}
\]

(9)

where \(G_{i}\) is the Jacobian of the motion model with respect to the predicted state \(\bar{y}_{i}\), and \(R\) is the motion noise covariance in Eq. (2).

Next, let us explain the correction step. The uncertainty of the state enlarged in the prediction stage is reduced in the following correction stages. When the line measurement \(z_{l,i}\) is detected from the \(n\)th line landmark, the state correction is formulated as follows:

\[
V_{l,i}^{y} = \frac{\partial y_{l}}{\partial \bar{y}_{i}}
\]

If \((T_{l,i}^{y} - x_{i} \cos(T_{l,i}^{\theta}) - y_{i} \sin(T_{l,i}^{\theta})) > 0\),

\[
\begin{bmatrix} \cos(T_{l,i}^{\theta}) & \sin(T_{l,i}^{\theta}) & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 1 & y_{i} \sin(T_{l,i}^{\theta}) - y_{i} \cos(T_{l,i}^{\theta}) & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \end{bmatrix}
\]

(10)

else

\[
\begin{bmatrix} \cos(T_{l,i}^{\theta}) & \sin(T_{l,i}^{\theta}) & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ -1 & -y_{i} \sin(T_{l,i}^{\theta}) + y_{i} \cos(T_{l,i}^{\theta}) & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \end{bmatrix}
\]
where $V^m_{Pj}$ is the Jacobian of the line measurement model with respect to $P^m_j$, and $Q_L$ is the noise covariance of the line measurements. Similarly, the correction by point features is formulated by the following equations:

$$V^m_{Pj} = \frac{\partial v_y(\mathbf{x}_y, P^m_y)}{\partial P^m_j} + \mathbf{0} \quad (15)$$

$$d_y = \mathbf{y}_y - P^m_j,$$  

$$K^m_{Pj} = C_{y,Pj} V^m_{Pj} \left(V^m_{Pj} C_{y,Pj} V^m_{Pj} + Q_L\right)^{-1} \quad (16)$$

$$\mathbf{y}_y = \mathbf{y}_y + K^m_{Pj} \left(z^m_{Pj} - v_y(\mathbf{x}_y, P^m_y)\right) \quad (17)$$

$$C_{y,Pj} = \left(I - K^m_{Pj} V^m_{Pj}\right)C_{y,Pj} \quad (18)$$

The formulations introduced so far are the traditional EKF-based SLAM framework, but the traditional method sometimes loses track of the robot’s true pose, especially when only a few landmarks are within the FoV. Our main contribution is proposed so as to improve the accuracy of SLAM in such situations. It is described in the next section.

3. Relational Constraints

When a pair of point and line landmarks are simultaneously observed, we can see a geometric relationship between them. This paper proposes to exploit this relationship in order to improve the accuracy of the mapping and localization. More specifically, the relationship is applied as a relational constraint between a pair of landmarks in the constrained-EKF [19] framework. Since the constraint is not obtained from a priori knowledge but rather from an observed geometric relationship, it is considered as a soft constraint with uncertainty $\sigma^2$. When a pair of line and point measurements $z^l_{Pj}$ and $z^p_{Pj}$ is detected, their relational constraint is defined by the distance between them, which is calculated by:

$$z^l_{Pj} = d\left(z^l_{Pj}, z^p_{Pj}\right) = \left|z^l_{Pj} - z^p_{Pj}\cos(z^p_{Pj}) - z^p_{Pj}\sin(z^p_{Pj})\right| \quad (19)$$

Eq. (19) can be derived from Eq. (4) by substituting $(z^l_{Pj}, z^p_{Pj})$ for $(I^l_{Pj}, I^n_{Pj})$ and $(z^p_{Pj} \cos(z^p_{Pj}), z^p_{Pj} \sin(z^p_{Pj}))$ for $(x, y)$, where Eq. (4) calculates the distance between the robot and a line landmark in the global frame, while Eq. (19) calculates the distance between line and point measurements in the robot’s frame. The geometric relationship of the line and point measurements is shown in Figure 4. We utilize this relationship as a soft constraint on EKF-SLAM. The constraint model for a pair of line and point landmarks corresponding to Eq. (19) is defined as:

$$\zeta^l_{ij} \in \mathcal{H}(I^l_{Pj}, I^n_{Pj})$$

$$= \left|I^l_{Pj} - P^m_{Pj}\cos(I^n_{Pj}) - P^m_{Pj}\sin(I^n_{Pj})\right| + N(0, \sigma^2_{ij}) \quad (20)$$

where $z^l_{Pj}$ is obtained from $I^l_{Pj}$, $z^p_{Pj}$ is from $P^m_{Pj}$ and $\sigma^2_{ij}$ is the noise variance of the relational constraint $\zeta^l_{ij}$ that represents uncertainties in the line and point measurements in Eq. (19).

The derivation of Eq. (20) is similar to the derivation of Eq. (19), substituting $(P^m_{Pj}, P^m_{Pj})$ for $(x, y)$ in Eq. (4). The noise covariance matrices of the line and point measurement models are transformed in order to compute $\sigma^2_{ij}$:

$$\text{Figure 4. Geometric description of a relational constraint. The distance between a pair of line and point measurements is applied as a relational constraint on the corresponding line and point landmarks.}$$
The relational constraint with the uncertainty calculated in Eq. (21) is applied in the EKF-SLAM through the following equations:

\[
\begin{align*}
H_i^{(n,m)}(j) &= \frac{\partial h \left( \hat{T}_i^f, \hat{P}_i^m \right)}{\partial y_t} \\
if \left( T_{d,j}^m - P_{d,j}^m \cos(T_{b,j}^m) - P_{d,j}^m \sin(T_{b,j}^m) > 0 \right) &= \begin{bmatrix} 0 & 0 & 1 \ -\cos(T_{b,j}^m) \ -\sin(T_{b,j}^m) \ 0 & 0 & 0 \end{bmatrix} \\
else &= \begin{bmatrix} 0 & 0 & -1 \ -\sin(T_{b,j}^m) \ \cos(T_{b,j}^m) \ 0 & 0 & 0 \end{bmatrix}
\end{align*}
\]

The Jacobian equation of Eq. (22) is divided into the two cases shown here because we take the absolute value in Eq. (20). With equations (22) through to (25), the relational constraint corrects all the related landmarks and the robot’s pose - as well as the paired landmarks - in the EKF frame.

At this point, let us discuss the computational issue. When there are multiple line and point measurements, the possible number of pairs of line and point measurements is \( n_L \times n_P \), where \( n_L \) is the number of line measurements and \( n_P \) is the number of point measurements. If \( n_L \) and \( n_P \) are less than two, as usual, the possible pairs are not overly large for real-time performance. However, when there are several more line and point measurements, namely in the worst cases where \( n_L, n_P > 2 \), applying all the relational constraints of all the possible pairs is computationally heavy and can lead to overconfidence in the state estimation. The overconfidence means that the estimated uncertainty of the state decreases below the actual uncertainty, or even that the covariance matrix can lose its positive definiteness. Therefore, we selectively apply the relational constraints by forming a single pair for each line measurement: a line measurement is paired only with the closest point measurement to the line, whereas a point measurement can be assigned to multiple line measurements. This limitation reduces the computational load for applying constraints when there are more than a couple of measurements, and the performance of SLAM is improved when there are only a few measurements in a sparse map.

4. Experiments

The proposed SLAM was tested in both simulations and a real-world experiment, and we compare its performance here with that of a traditional SLAM. By ‘traditional’ SLAM, we mean the SLAM described up until Section 2. The traditional and proposed SLAMs are the same except that the proposed SLAM utilizes the relational constraints given in Section 3. Four measures are used to evaluate the SLAM performance: trajectory error, line mapping error, number of outlying line landmarks and point mapping error. Because the same map might look different if it uses different origins and orientations, translations and rotations are applied to the coordinates of the trajectory and map prior to assessing the performance of the SLAM. The translations and rotations are represented by a vector:

\[
t = \begin{bmatrix} t_x \\ t_y \\ t_b \end{bmatrix}
\]

and the trajectory and the map are transformed by:

\[
\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(t_b) & \sin(t_b) \\ -\sin(t_b) & \cos(t_b) \end{bmatrix} \begin{bmatrix} x \\ y \\ t \end{bmatrix}
\]

The transformation \( t \) is selected to minimize the sum of the four performance measures of error, defined by:

\[
\text{TotalErr} = \text{TrjErr}(t) + \text{LineOL}(t) + \alpha_{OL} \cdot \text{LineOL}(t) + \text{PointErr}(t)
\]

where \( \text{TrjErr} \), \( \text{LineOL} \) and \( \text{PointErr} \) denote the trajectory error, line mapping error and point mapping error, respectively, such that they all depend on \( t \). LineOL is the number of outlying line landmarks that are not associated with line landmarks in the true map. \( \alpha_{OL} \) is the weight for LineOL and is set to 0.1 in this experiment. The trajectory error is simply computed by the mean of the distances between the trajectories of the ground truth and the SLAM at each time step. The point mapping error is calculated by the average distance between a point landmark obtained by SLAM and the corresponding true point landmark, where a point landmark detected by a ceiling camera is associated with the closest point landmark in the map. The line mapping error is the average difference between the line feature in a map and the associated true line landmark, and the difference between the line landmarks are defined as:
\[ D_L = \left( L_d \cos(L_d) - L_d \cos(L_d) \right)^2 + \left( L_d \sin(L_d) - L_d \sin(L_d) \right)^2 \right)^{1/2} \]  

(29)

where \( L \) is a true line landmark and \( L \) is a SLAM line landmark. A SLAM line landmark is associated with the closest true line landmark if two landmarks overlap. The following subsections describe the experimental environments and conditions and provide the results of the performance evaluation.

4.1 Simulation Results

First, we conducted simulations to test the performance of the proposed SLAM in a synthetic environment. A number of line and point features were installed in the synthetic environment, and the environment was more complex than our real-world experimental conditions, as described in the next subsection. Two different environments were used that differed in size, trajectory lengths and the number of landmarks, as shown in Figure 5. The first map - called M1 - consists of four rooms with hallways around them. It has 84 line landmarks and 34 point landmarks. The robot travelled along 20 different trajectories for 500 time-steps. The second map - M2 - includes 76 line landmarks and 51 point landmarks and has a relatively large and complex structure. Again, the robot travelled along 20 different trajectories for 1,000 time steps in M2.

Since the inputs suffer from Gaussian random noise, the simulation results are not deterministic (even through the same trajectory in the same environment). In the simulations, the noise conditions are set as \( R = \text{diag}(0.01, 0.01, 0.001) \) and \( Q_e = Q_r = \text{diag}(0.01, 0.001) \). To examine the statistical reliability of the performance evaluation, 20 runs were made with different trajectories. The trajectory errors are evaluated for each trial and the mean and standard deviation are abstracted from the results. The mean value measures the accuracy of the state estimation and the standard deviation evaluates the precision. The trajectory depicted by the dotted line in Figure 5 is an example of the trajectories used in the simulation. The square mark represents the robot’s starting point.

The simulation results for M1 are provided in Figure 6 and Table 1. Figure 6 shows the results for one of the 20 simulation runs. In the figure, the dotted and solid trajectories depict the true and estimated trajectories, respectively. Equally, the dotted and solid lines represent the true and estimated line landmarks, respectively. Finally, the triangular and circular marks represent the true and estimated point landmarks, respectively. As the figure shows, the results of using the proposed SLAM are closer to the ground truth than those using the traditional SLAM.
Table 1. Comparison of SLAM errors in M1

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Traditional SLAM</th>
<th>Proposed SLAM</th>
<th>Reduction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TrjErr (m)</td>
<td>mean 0.391</td>
<td>0.322</td>
<td>17.6</td>
</tr>
<tr>
<td></td>
<td>std 0.180</td>
<td>0.120</td>
<td>33.0</td>
</tr>
<tr>
<td>LineErr (m)</td>
<td>mean 0.409</td>
<td>0.336</td>
<td>18.0</td>
</tr>
<tr>
<td></td>
<td>std 0.184</td>
<td>0.154</td>
<td>16.5</td>
</tr>
<tr>
<td>LineOL</td>
<td>mean 2.35</td>
<td>0.85</td>
<td>63.8</td>
</tr>
<tr>
<td>PointErr (m)</td>
<td>mean 0.660</td>
<td>0.486</td>
<td>26.4</td>
</tr>
<tr>
<td></td>
<td>std 0.467</td>
<td>0.378</td>
<td>18.9</td>
</tr>
</tbody>
</table>

The numerical results of all 20 simulations are summarized in Table 1. The table shows the means and standard deviations of the four error measures for each method and their reduction using the proposed SLAM as compared to the traditional SLAM. The trajectory error is reduced by 17.6%, and the line and point mapping errors are reduced by 18.0% and 26.4%, respectively, using the proposed method. The standard deviations of the errors are also substantially reduced. The number of outlying line landmarks is reduced to 0.85, which means that almost every line landmark estimated by the proposed SLAM is associated with the corresponding true line landmark.

Figure 7 and Table 2 provide the simulation results for the environment M2. The figure displays the results for one of the 20 simulation runs. The dotted and solid trajectories, the dotted and solid lines, and the triangular and circular marks, all have the same meanings as in Figure 6. As in the simulation for M1, the proposed SLAM estimates the trajectory and the map much more accurately (i.e., closer to the ground truth) than the traditional SLAM. In Table 2, the simulation results of the two methods are summarized. The trajectory error of the proposed method is much smaller than that of the traditional method, while the mapping errors remain almost the same. The line mapping error may appear degraded in our method, but in fact the traditional method has a larger line mapping error dispersed in the many outlying landmarks. A large error for a line landmark makes it appear unrelated to the corresponding true landmark and thus to appear to be an outlying landmark. With the proposed method, the number of outlying line landmarks is reduced by 33.9% and the standard deviations of all four errors are decreased. The decrease in standard deviations indicates that the relational constraint used in the proposed SLAM stabilizes the performance. From the simulation results of M1 and M2, it can be concluded that the use of relational constraints between features improves the accuracy and the reliability of the SLAM.

Table 2. Comparison of the SLAM errors in M2

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Traditional SLAM (m)</th>
<th>Proposed SLAM (m)</th>
<th>Reduction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TrjErr</td>
<td>mean 0.613</td>
<td>0.470</td>
<td>23.3</td>
</tr>
<tr>
<td></td>
<td>std 0.610</td>
<td>0.161</td>
<td>73.6</td>
</tr>
<tr>
<td>LineErr (m)</td>
<td>mean 0.545</td>
<td>0.559</td>
<td>-2.6</td>
</tr>
<tr>
<td></td>
<td>std 0.291</td>
<td>0.250</td>
<td>14.2</td>
</tr>
<tr>
<td>LineOL</td>
<td>mean 6.35</td>
<td>4.2</td>
<td>33.9</td>
</tr>
<tr>
<td>PointErr (m)</td>
<td>mean 0.805</td>
<td>0.787</td>
<td>2.3</td>
</tr>
<tr>
<td></td>
<td>std 0.591</td>
<td>0.501</td>
<td>15.3</td>
</tr>
</tbody>
</table>

4.2 Experimental Results

In this subsection, we describe the experiment conducted to demonstrate the performance of the proposed method for real-world applications. In this experiment, the robot is set up as shown in Figure 8. It is equipped with an embedded camera on top, and the camera provides grey images with a 320-by-240 pixel resolution. We suggest that the camera should have at least 3 m of ceiling within the field of view for consistent landmark detection. The robot reads a velocity and a ceiling image at each step, and line and point features are extracted from the ceiling images, as described in Section 2.
Figure 8. A mobile platform equipped with a ceiling camera

Figure 9. Floor plan of the experimental site (a) and the ground truth trajectory and map (b)

Figure 9 shows the floor plan of the experimental site and the true map of line and point landmarks. The site is on the fourth floor of the Technopark in the city of Bucheon. The true map (the ground truth of the map) was reconstructed from the floor plan of the building and then manually corrected. The robot moved around the U-shaped hallway in the building for 5,232 sampling times, and the map was about 17 m × 18 m. The true trajectory, shown in Figure 9(b), was obtained from the indoor localization system, named Stargazer [20].

Table 3. Comparison of the SLAM errors in the experiment

<table>
<thead>
<tr>
<th></th>
<th>Traditional SLAM</th>
<th>Proposed SLAM</th>
<th>Reduction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TrjErr (m)</td>
<td>48.23</td>
<td>26.58</td>
<td>44.9</td>
</tr>
<tr>
<td>LineErr (m)</td>
<td>35.17</td>
<td>26.87</td>
<td>23.6</td>
</tr>
<tr>
<td>LineOL</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>PointErr (m)</td>
<td>50.40</td>
<td>32.09</td>
<td>36.3</td>
</tr>
</tbody>
</table>

The results of both SLAM methods are shown in Figure 10. The dotted and solid trajectories, the dotted and solid lines, and the triangular and circular marks, all have the same meaning as in the figures for the simulations. The traditional SLAM updates line and point landmarks independently (i.e., without regard for their relationships), and the resulting estimations of the map and the trajectory are quite different from the ground truth, as shown in Figure 10 (a). In particular, when the robot is far away from the starting location (in the right-hand part of the map), the estimation performance is seriously degraded. In contrast, the proposed SLAM exploits the relational constraints between the line and point landmarks and estimates the map and trajectory extremely closely to the ground truth over the whole
environment. The numerical results are summarized in Table 3. The trajectory error of the proposed SLAM is reduced almost by half compared to the traditional method. In addition, the line and point mapping errors are significantly decreased, by 23.6% and 36.3%, respectively. However, the number of outlying lines remains the same because the environment is not as complex in the experiment as it is in the simulations, and association failures rarely occur in the simpler environment. The three outlying line landmarks arise from line landmarks that do not belong to the ground truth map.

To evaluate the capability of its real-time operation, the processing times for feature extraction and state estimation were measured. The feature extraction process takes 38.4 ms and the state estimator takes 0.3 ms on average over sampling times on a computer with an i5-2500 CPU and 4GB RAM. Therefore, the total mean processing time is 38.7 ms, which is fast enough to operate in real-time.

5. Conclusions

This paper proposes a method that utilizes observed relational constraints between different types of features to improve accuracy of SLAM estimation in environments with sparsely distributed landmarks. CV in an indoor environment provides static and robust features, but those features are not plentiful. Therefore, additional information is needed for good performance. We employ the distance between line and point landmarks as relational constraints, which prevents the line and point maps from becoming misaligned, thereby increasing localization accuracy. We implement this idea using line and point features in the EKF SLAM framework. Simulation and experimental tests of the proposed method showed improved results compared with the traditional method that updates the state vector without using relational constraints. The experimental results provide a real-world demonstration of our method, showing that the trajectory and maps estimated using relational constraints are closely fitted to the ground truth. Our algorithm was also tested through simulations so as to verify the performance of the proposed SLAM in more complex, larger environments, through various trajectories. The simulation results show that our method successfully improves the general performance of SLAM. Furthermore, our approach’s utilization of relational constraints also improves the stability of the estimation filter.

In future work, we plan to apply relational constraints between various kinds of features or between features of the same type for conditions in which SLAM suffers from sparse landmarks. Furthermore, we may need to develop new relational constraints in addition to the distance measure used in this paper, in order to constrain the landmarks’ locations more tightly in multiple dimensions.

6. Acknowledgments

This research was supported by the Basic Science Research Programme through the National Research Foundation of Korea (NRF), funded by the Ministry of Education, Science and Technology (NRF-2013R1A2A2A01015624).

7. References


