SLAM with Salient Line Feature Extraction in Indoor Environments

Su-Yong An, Jeong-Gwan Kang, Lae-Kyoung Lee, and Se-Young Oh
Department of Electrical and Electronic Engineering
Pohang University of Science and Technology (POSTECH)
Pohang, Gyeongbuk, Korea
{grasshop, naroo1, laeklee, syoh}@postech.ac.kr

Abstract—This paper presents a simultaneous localization and mapping (SLAM) of a large indoor environment using Rao-Blackwellized particle filter (RBPF) along with line segments as the landmarks. To represent the environment as a compact form, we use only two end points of the line segment, reducing computational cost in modeling line uncertainty. With a modified scan point clustering method, the proposed adaptive iterative end point fitting (IEPF) plays an important role in estimating line parameters by taking a noisy scan point near end points into account. Thus, by line-segment matching the robot is localized well in a local frame. We also introduce an online global optimization of a map, which provides more consistent map by removing spurious lines and merging collinear lines. Each of our approaches is efficiently integrated into the proposed RBPF-SLAM framework. Experiments with well-known data set demonstrate that the proposed method provides a reliable SLAM performance along with a compact map representation.

Keywords—Line segment, Localization, Mapping, Iterative End Point Fitting (IEPF)

I. INTRODUCTION

A Simultaneous Localization And Mapping (SLAM) problem is to estimate the pose of a robot and a map simultaneously while the robot navigates in unknown environment. There has been plenty of work for solving SLAM problem over the past decades [2-5]. The SLAM algorithm can be classified according to the method of how to estimate states of the robot and the map, and how to represent the map.

Among estimation methods, Extended Kalman Filter (EKF) has occupied the attention of mobile robot researchers and is still used for most mobile robot applications. However, its growing computational cost with increasing number of landmarks causes a problem in a large indoor mapping. As an alternative to EKF, Murphy introduced Rao-Blackwellized particle filter (RBPF) to SLAM problem, and recently this approach has been extended by several SLAM researchers [1], [4]. RBPF-SLAM has advantages over EKF-SLAM in that it reduces computational cost through factorizing posterior distribution so as to remove correlations between the robot pose and landmarks, and it is also robust to a false data association because of a resampling process.

For mapping of an indoor environment, various works using RBPF are based on the environment representation consisting of a point feature which is detectable by the robot sensory system [6]. This point feature-based representation has advantage in its compactness, however, the point feature such as the corner may not be extracted enough for acceptable SLAM performance, especially when we use a range sensor (e.g., sonar rings, a laser scanner). Another type of the map representation is a grid-based map, which can represent arbitrary features and provide detailed information of the environment. However, such grid-based representation is computationally expensive and also requires a huge amount of memory, especially for large environments [5].

In this respect, we consider a line segment-based representation in that it requires extremely small amount of memory and a lot of line segments exist in a typical indoor environment. As the first step for mapping, the line segment extraction stage is necessary and thus to compare a performance of several line segment extraction methods using 2D range data, Nguyen conducted extensive work, reporting that Split-and-Merge and Incremental have best performance due to their superior speed and correctness [12]. Mapping methods using line segment is proposed in [7-9]. In these approaches, all the scan points which contribute a line segment are used to calculate the line parameter and its covariance matrix which indicates uncertainty of the observed line. However, to compute the line uncertainty, each point uncertainty should be taken into account and this involves many Jacobian calculations. Nguyen proposed OrthoSLAM using line segments, exploiting orthogonality constraint (or assumption) existing in most indoor environments that lines are parallel or orthogonal to each other [10]. However, we cannot assure this assumption is satisfied in all indoor environments. For robotic vacuum cleaners, Choi proposed EKF-SLAM using line features extracted from efficient management of sparse sonar readings [11]. However, same orthogonal constraint is imposed to achieve reasonable SLAM performance, moreover mapping region is limited (about 5 m by 5 m). Zhang proposed a closed line segment (CLS) map, introducing virtual line segment which is useful for an exploration of unknown environment [19]. However, the line correspondence method considers only two measurements; a line orientation and a distance between each center of gravity (COG). Sohn proposed VecSLAM, where a map is

This work was supported by National Strategic R&D Program for Industrial Technology of the Korean Ministry of Knowledge Economy.

978-1-4244-7815-6/10/$26.00 ©2010 IEEE
constructed by vectors [20]. Vector merging is performed using recursive least squares method, and simple loop closing detection is implemented with vector matching algorithm. However, uncertainty of the map vector is not presented explicitly, and previous robot poses are sequentially modified after a loop closing, thus it may take a long time to handle a loop closing.

In this paper, we present RBPF-SLAM using line segments for mapping of an indoor environment. For clustering of raw scan points, we propose a dual breakpoint detector (DBD) which makes the most of both a distance and an angle of two consecutive scan points. Due to speed and correctness reported by [12], we adopt iterative end point fitting (IEPF) [16], which is kind of Split-and-Merge algorithm, to extract line segment in 2D clustered range data. Moreover, we modify traditional IEPF by introducing an adaptive splitting criterion, improving quality of line orientation estimation under a cluttered environment. This provides a significant improvement in odometry error compensation. To pursue computationally inexpensive algorithm, we use only two end points of a line segment for representing and calculating line parameter and its uncertainty, reducing computational complexity. We do not impose any restrictions such as the orthogonal constraint on the structure of indoor environments. Finally, an online global optimization of a line feature map is proposed to ensure a consistent map with a compact representation of an indoor environment.

To sum up, our contribution is two-fold. (i) We improve the clustering and the line extraction algorithm so as to extract reliable line parameter in the cluttered environment using only two end points. (ii) Each proposed method is well integrated into the proposed RBPF-SLAM framework by making the most of line segment attributes, reducing computational cost.

II. LINE SEGMENT REPRESENTATION AND EXTRACTION

A. Line Representation

As is common in SLAM literature, a line segment \( l_i = (r_i, \psi_i) \) in a global frame is parameterized by its distance \( r_i \) from the origin and the direction \( \psi_i \) of the normal passing through the origin. Let us denote the robot pose by \( x(k) = [x_r(k), y_r(k), \theta_r(k)]^T \), which consists of its position and heading angle at time \( k \).

Suppose that the robot can observe the \( i^{th} \) line segment \( l_i \) using its sensory system at time \( k \), then the observed line segment can be transformed to the robot centered frame (i.e. a local frame) through a following measurement function:

\[
\mu_i(x(k), \ell_i) = \begin{bmatrix}
-\rho_i + x_r(k) \cos \psi_i + y_r(k) \sin \psi_i \\
\psi_i - \theta_r(k) - \frac{\pi}{2} \\
\rho_i - x_r(k) \cos \psi_i - y_r(k) \sin \psi_i \\
\psi_i - \theta_r(k) + \frac{\pi}{2}
\end{bmatrix}
\]

if intersect

\[\mu_i(x(k), \ell_i) \quad \text{else}\]

B. Scan Point Clustering With a Dual Breakpoint Detector

We assume that the robot is equipped with a proximity sensor such as a laser scanner, so it provides \( N \) range and bearing measurements:

\[
p_i = (d_i, \phi_i) = (d_i \cos \phi_i, d_i \sin \phi_i), \quad i = 1, 2, ..., N,
\]

where \( d_i \) is a distance to a sensed object and \( \phi_i \) is a direction of scan point w.r.t. \( x \)-axis in the robot centered frame. To extract a line segment from these raw scan points, several line extraction algorithms are proposed such as RANSAC-based, Hough transform-based and EM-based method [12]. However, all these methods use a dense set of scan points directly instead of undergoing a preprocessing step, and hence computational cost is rather high compared to a method which includes a segmentation of raw scan points.

In this paper, as a preprocessing step, we first detect a breakpoint which is defined as a discontinuity in a sequence of raw measurements. Most of the breakpoint detection algorithms are based on a distance between two consecutive points \( p_i, p_{i+1} \):

\[
\|p_i - p_{i+1}\| < D_{\text{thresh}},
\]

where \( D_{\text{thresh}} \) is a predefined threshold. If (4) is satisfied, then \( p_i \) and \( p_{i+1} \) are considered as being in the same cluster. Otherwise, \( p_{i+1} \) is supposed be a first element of a new cluster, while \( p_i \) and \( p_{i+1} \) turn out to be breakpoints. Therefore, a careful selection of the threshold condition \( D_{\text{thresh}} \) is crucial for the clustering of scan points.

Several breakpoint detectors were proposed in [14] and [15]. This kind of point distance-based (PDB) method is adequate for dense scan points set measured from a laser scanner with a high angular resolution. Even so, PDB method cannot deal with the case in which a distance between the robot and the sensed object is long enough, even though the distance is taken into account (Fig. 2a).
To cope with this case, we propose a dual breakpoint detector (DBD) that combines PDB with an angle-based detection. The angle-based detection is similar to incremental detector (DBD) that combines PDB with an angle-based clustering algorithm, or Line-Tracking [12], however, to reduce computational cost our approach does not update line parameters but store only previous line angle.

In this paper, we adopt a well-known recursive line extraction algorithm, or iterative end point fitting (IEPF) proposed by Duda and Hart [16]. This algorithm recursively splits the cluster of scan points can be assigned to one of two types according to its clustering criterion; one is the distance-based cluster and the other is the angle-based cluster (Fig. 2b).

Suppose that the distance-based cluster \( S_k = \{ p_i | i=1, ..., N_k \} \) has \( N_k \) number of scan points, then the next scan point \( p_j \) is tested in conjunction with the last scan point \( p_{N_j} \) to determine whether \( p_j \) belongs to \( S_k \) or not. If \( p_j \) do not satisfy (4), then the \( p_j \) and \( p_{N_j} \) form a new cluster, which is assigned to one of two types.

C. Line Extraction using an Adaptive IEPF Algorithm

Suppose that we have \( M \) number of clusters \( S^\prime = \{ S_i | i=1, ..., M \} \) through the clustering step. These clusters should be further processed to be divided into several line segments. In this paper, we adopt a well-known recursive line extraction algorithm, or iterative end point fitting (IEPF) proposed by Duda and Hart [16]. This algorithm recursively splits the cluster \( S_i \) into two subsets \( S_i^1, S_i^2 \) until a validation criterion is not satisfied (Fig. 3a). The criterion validates lines with \( \delta_j \leq \delta_{th} \). This simple criterion works well in most applications.

In this paper, along with IEPF, we only use two end points instead of using all scan points to estimate line parameters, so as to pursue computationally inexpensive algorithm.

In traditional IEPF, threshold value \( \delta_{th} \) is a constant irrespective of the position of a point \( p_j \); however, the constant \( \delta_{th} \) which forms a constant validation window may give false end point fitting (Fig. 3b). Therefore, we propose an adaptive IEPF which can vary \( \delta_{th} \) according to the projected position \( \bar{p}_j \) of \( p_j \). This varying \( \delta_{th} \) forms an ellipse-shaped adaptive validation window (Fig. 3c), and we model \( \delta_{th} \) as a function of \( d_{ij} \) and \( d_{2j} \), where each indicates a distance between the end point and \( p_j \):

\[
\begin{align*}
     d_{ij} = 0.5(d_{ij} + d_{2j}), \\
     \min_j = \min(d_{ij}, d_{2j}), \\
     h_{max} = 0.015\ln(2d_{ij}) + 0.05, \\
     \delta_{th} = h_{max} \sqrt{1 - \left( \frac{d_{ij} - \min_j}{d_{ij}} \right)} - \frac{h_{max}}{2d_{ij}} \sqrt{\min_j(2d_{ij} - \min_j)},
\end{align*}
\]

where \( h_{max} \) is a maximum threshold when \( \bar{p}_j \) is at the center of the line \( l_j \). The other constant values are empirically determined. This approach provides a reduction of a noise arising near end points, and hence giving better estimates of line parameters (Fig. 3d).

III. LINE SEGMENT-BASED SCAN MATCHING

A. Line Segment-Based Scan Matching for Odometry Error Compensation

In this paper, the purpose of a scan matching is to compensate odometry error occurred in short travel distance so the robot is localized well in a local frame. This process alleviates the need of large number of particles, because it helps to reduce a covariance of a proposal distribution in a sampling step. Let us denote a set of the robot pose and line segments extracted at time \( k \) by \( T(k) = \{ x(k), l_1(k), ..., l_N(k) \} \), where \( N \) is the number of line segments. The previous set \( T(k-1) \) can be transformed to the current robot frame \( \hat{T}(k) = x(k) \Pi T(k-1) \). Therefore, all the line segments of both sets are in the same reference frame, matched lines are likely to be almost parallel or intersect at some point. The matched line segments only differ in translation and rotation due to the fact that the robot obeys a rigid motion in 2-dimensional space.

Thus, we first find a rotational difference of the matched line segment, and then we calculate a translational difference
simply by computing the difference of line parameters \( \hat{\rho}_i(k) - \rho_i(k) \) since a rotation of line segment cannot affect \( \rho_i \):

\[
\Delta \rho_i(k) = \left[ \hat{\rho}_i(k) - \rho_i(k) \right], \quad \Delta \alpha_i(k) = \left[ \hat{\alpha}_i(k) - \alpha_i(k) \right],
\]

(7)

\[
\Xi(k) = \hat{x}(k) + \Delta \xi = \left[ \begin{array}{c}
x_i(k) \\
y_i(k) \\
\theta_i(k)
\end{array} \right] + \frac{1}{N_m} \sum_{i=1}^{N_m} \Delta \rho_i(k) \cos \alpha_i(k) + \frac{1}{N_m} \sum_{i=1}^{N_m} \Delta \rho_i(k) \sin \alpha_i(k),
\]

(8)

where \( N_m \) is the number of matched pair, \( \Xi \) is the compensated robot pose by line segment-based scan matching (Fig. 4).

B. Data Association

Before compensating an odometry error, we have to make an exact corresponding pair among several line segments. As an intuitive method, one would compare the line parameters directly to see a matching degree. However, different lines in the environment may lie on the same line and this often happens in common indoor environment such as a hallway [7].

In this paper, we consider an overlapping rate of two line segments as well as a difference of line parameters. When the angle between two lines is below a specified threshold, the lines are considered as candidates for possible matching pair [13]. Having filtered line segments which may form parallel or collinear pairs using an angular difference criterion alone, we are given a pair of line segments \( l_i = \{\rho_i, \alpha_i, \rho_i^{m}, \alpha_i^{m} \} \) and \( l_j \), where \( \rho_i \) and \( \alpha_i \) are line parameters, successive three coordinate pairs \((x, y)\) denotes the position of two end points and a middle point, \( d_i \) is a length of the line, and \( n_i \) is the number of scan points which compose the line segment. We can further examine how much this pair overlaps, denoting overlapping factor \( \alpha \).

We first find the reference line \( L_r \) onto which the end points are supposed to be projected. \( L_r \) is defined by its orientation \( \alpha_{r} \) and the point \( p_m \) which it passes (Fig. 5):

\[
\alpha_r = \frac{d_i \alpha_i + d_j \alpha_j}{d_i + d_j}, \quad p_m = \frac{d_i p_i^{m} + d_j p_j^{m}}{d_i + d_j}.
\]

(9)

We can now project the end points of two line segment on the reference line, denoting by \( \hat{p}_i^{x}, \hat{p}_i^{y}, \hat{p}_j^{x}, \hat{p}_j^{y} \) and \( \hat{p}_i^{2x}, \hat{p}_j^{2x} \), respectively. A length \( d \) of the reference line is defined by the pair of points which form a largest distance among any pair of the projected points. Therefore, the overlapping factor \( Q \) is calculated as follows:

\[
Q = \frac{\| \hat{p}_i^{x} - \hat{p}_j^{x} \| + \| \hat{p}_i^{y} - \hat{p}_j^{y} \|}{d_j}.
\]

(10)

This definition ensures that \( Q \) is always in the range (0, 2]. A value of one can be a threshold that can determine whether or not two line segments overlap. For example, if \( Q \) is less than one, then it implies that there is no overlapping region between two line segments. Otherwise, it is considered that there is an overlapping region with a degree of \( Q \). A perfect overlapping occurs when \( Q \) is exactly two. For a reliable data association, we also need a disparity between line segments, because the overlapping factor cannot provide how far these line segments are away. Here, we define a disparity \( D_{ij} \) of two line segments using an average distance between the end point and its projected point onto the reference line:

\[
D_{ij} = \sum_{k=1}^{2} \frac{\| \hat{p}_i^{x} - \hat{p}_j^{x} \| + \| \hat{p}_i^{y} - \hat{p}_j^{y} \|}{2}.
\]

(11)

By integrating all these angular constraint, overlapping factor, and disparity together, we make data association criteria as follows:

\[
| \alpha_i - \alpha_j | < \alpha_{th}, \quad Q < Q_{th}, \quad D_{ij} < D_{th}.
\]

(12)

When all these constraints are satisfied, a pair of line segments is considered to be associated. This data association scheme is used not only for odometry error compensation but also for a global association with a set of different threshold conditions (a global optimization of a map will be discussed in section IV-B). Note that we usually set strict threshold conditions for odometry error compensation compared to a case of global association, because it is often the case that two line segments are initially well aligned due to odometry information though it is not an exact measurement.
IV. LINE UNCERTAINTY MODELING & OPTIMIZATION

A. Measurement Update and Uncertainty Modeling of an Observed Line Segment

In RBPF framework, there is no correlation between any landmarks due to a factorization of posterior distribution, and thus we update only an observed landmark independently. We use EKF to estimate each landmark state as follows [3]:

\[ \mathbf{K} = \mathbf{H}^T \mathbf{Q}_i \mathbf{H} + \mathbf{R}, \]
\[ \hat{\mathbf{x}} = \mathbf{x} + \mathbf{K} (\mathbf{z} - \hat{\mathbf{z}}), \]
\[ \hat{\mathbf{P}} = (1 - \mathbf{K} \mathbf{H}) \mathbf{P}, \] (13)

where \( \mathbf{P} \) and \( \mathbf{Q}_i \) are landmark states and its covariance, \( \mathbf{H} \) is the derivative of measurement function w.r.t. landmark states, \( \mathbf{K} \) is kalman gain, \( \mathbf{z} \) and \( \hat{\mathbf{z}} \) are actual and predicted measurement, \( \mathbf{Q} = \mathbf{Q}_i + \mathbf{H} \mathbf{P} \mathbf{H}^T \) is a measurement information, respectively. Here, we omit all particle, time and landmark index for brevity.

In accordance with SLAM literature, we adopt line parameters \( t^T = [r \ y]^T \) as landmark states in a global frame. Suppose that the robot observed a line segment \( l_i = \{x, y, r, \theta, \phi, \mu, \sigma^2, d, n_i\} \) with covariance \( \mathbf{Q}_i \) indicates uncertainty of the observed line segment. A modeling of a line covariance \( \mathbf{Q}_i \) is important, because badly approximated covariance affects an update of landmark state severely, and it may cause data association problem in the future association.

In [7] and [8], a modeling of covariance matrix of a line segment is presented, however, it uses all scan points which forms the line segment and becomes computationally expensive. In order to be consistent with the line extraction method proposed in this paper, we use only two end points for modeling of the covariance matrix. Let us denote a line equation in polar coordinate by \( \rho = r \cos \theta + y \sin \theta \). By using two end points, we have explicit form of \( \rho \) and \( \theta \) as follows:

\[
\begin{bmatrix}
\rho_i \\
\theta_i
\end{bmatrix} = \begin{bmatrix}
g_1(x_1^1, y_1^1, x_2^1, y_2^1) \\
g_2(x_1^2, y_1^2, x_2^2, y_2^2)
\end{bmatrix} = \begin{bmatrix}
\frac{1}{2} (x_1^1 + x_2^1) \cos \theta + (y_1^1 + y_2^1) \sin \theta \\
\tan^{-1} \left( \frac{x_1^2 - x_2^2}{y_1^2 - y_2^2} \right)
\end{bmatrix}.
\] (14)

Assuming that the errors added to two end points are uncorrelated, the covariance of observed line can be approximated as:

\[ \mathbf{C}_{li} = \mathbf{J}_{li} \mathbf{C}_{n2} \mathbf{J}_{li}^T + \mathbf{J}_{li} \mathbf{C}_{n1} \mathbf{J}_{li}^T, \] (15)

where the matrix \( \mathbf{J}_{li} \) and \( \mathbf{J}_{li} \) are the Jacobians of \( q(\cdot) \) w.r.t. the end points \( (x_1^1, y_1^1) \) and \( (x_2^2, y_2^2) \), respectively. \( \mathbf{C}_{n1} \) and \( \mathbf{C}_{n2} \) are the covariance matrices of the noise affecting the position of the two end points. However, actual raw measurements of two end points are \( (\hat{d}^i, \phi_i^i) \) and \( (\hat{d}^i, \phi_i^i) \), and hence it is transformed to Cartesian coordinate by following simple transformation:

\[ \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} d \cos \phi \\ d \sin \phi \end{bmatrix}, \] (16)

Therefore, \( \mathbf{C}_{n1} \) can be expressed as follows:

\[ \mathbf{C}_{n1} = \mathbf{J}_{li} \mathbf{C}_{n2} \mathbf{J}_{li}^T, \] (17)

where the matrix \( \mathbf{J}_{li} \) is the Jacobian of \( q(\cdot) \) w.r.t. the measurement \( (\hat{d}^i, \phi_i^i) \). Likewise, \( \mathbf{C}_{n2} \) follows a similar calculation with \( \mathbf{C}_{n1} \). \( \mathbf{C}_{n1} \) is the covariance matrix of the noise affecting the raw measurement \( (d, \phi) \). By substituting (17) into (15), we can approximate the covariance matrix of the line segment \( l_i \) as follows:

\[ \mathbf{C}_i = \mathbf{J}_{li} \mathbf{C}_{n1} \mathbf{J}_{li}^T + \mathbf{J}_{li} \mathbf{C}_{n2} \mathbf{J}_{li}^T, \] (18)

where \( \mathbf{C}_{n1} = \begin{bmatrix} \sigma^2_d & \sigma^2_d \\ \sigma^2_d & \sigma^2_d \end{bmatrix} \) (i.e., we assume that there is no correlation between \( d \) and \( \phi \)). Simply, we can put constant value to noise parameter \( (\sigma^2_d, \sigma^2_d) \), however, these raw measurements finally form the line segment and so are closely related to attributes of the line segment such as its length or the number of scan points on it.

Therefore, we vary noise parameter according to a line density \( \lambda \), which is defined by the number of scan points per unit length \( n/d \). The noise parameters are modeled as a function of the line density \( \lambda \):

\[ \begin{bmatrix} \sigma_d \\ \sigma_\phi \end{bmatrix} = \begin{bmatrix} c_1 \lambda \\ c_2 \end{bmatrix}, \] (19)

where \( \sigma_{d\text{max}} \) and \( \sigma_{\phi\text{max}} \) are the maximum standard deviations of noise affecting \( d \) and \( \phi \). The constant value \( c_1 \) and \( c_2 \) are empirically determined (e.g., \( c_1=1.24, \quad c_2=-0.4 \)). This implies that the noise parameters decrease with increasing line density.

B. Global Optimization of a Map

While the robot builds a line feature map during exploration of an environment, several dubious line segments which are actually extracted from a single line segment are likely to be generated. This is because a sample which is inconsistent to a given map is drawn from a proposal distribution. We can take this case into account by giving a high weight to a particle which has less number of line segments, and hence we intend to resample those particles. This is an indirect way of optimizing a map depending on a resampling process. As a direct way of optimizing a map, we merge several line segments which are associated each other in the global frame. We use a same data association scheme as in odometry error compensation with more relaxed thresholds. Let us denote the state and the covariance matrix of associated landmarks by \( (\mu_i, \Sigma_i) \) and \( (\mu_j, \Sigma_j) \), then the merged state and covariance can be computed by Covariance Intersection (CI) method as follows [17]:

\[ \hat{\Sigma}^{-1} = \omega \Sigma_i^{-1} + (1 - \omega) \Sigma_j^{-1} \]
\[ \hat{\mu} = \Sigma (\Sigma_i^{-1} \mu_i + (1 - \omega) \Sigma_j^{-1} \mu_j), \] (20)

where \( \omega \) is a parameter range from zero to one usually chosen to minimize a trace or a determinant of the fused covariance matrix.

Global optimization of a map is performed on a per-particle basis whenever \( N \), number of resampling steps is done so is an occasional process. In addition to this online optimization, we can also activate an offline global optimization after the whole SLAM process is finished to provide more consistent map in
terms of the alignment of line segments. By adjusting a threshold $Q_{th}$ of overlapping factor, this process merges non-overlapped but well-aligned line segments into one.

V. EXPERIMENTAL RESULTS

The approach proposed in this paper has been implemented and tested on well-known data set, named as Intel Lab [18]. The proposed adaptive IEPF in conjunction with a dual breakpoint detector improved odometry error compensation, so it enabled the filter to solve SLAM problem with small number of particles. We used 40 or less particles to build a reliable line feature map, and hence achieving a performance faster than real-time. The consistency of a line feature map was strengthened due to the interplay of our approaches; a dual breakpoint detector, an adaptive IEPF, a line association scheme, and an online/offline global optimization of a map.

A. Experimental Data Set

Intel Lab: The size of the map is 35 ×40 meters. Each a Pioneer2 DX mobile robot and a SICK LMS 200 laser scanner provided odometry and range measurements. This map is suitable to test our algorithm in that it has several long hallways which provide abundant line segments (Fig. 6).

B. Performance Improvement in Line Segment-Based Scan Matching

The traditional IEPF in conjunction with distance-based clustering of scan points was tested using data set. It resulted in a big discrepancy between initially observed line segment and the line segment in current observation at the point of the first loop closing (Fig. 7a). However, adaptive IEPF with distance and angle-based clustering showed a better performance in that odometry error was continually compensated through good estimates of line parameters (Fig. 7b).

C. RBPF-SLAM Results

We compared the overall performance of the proposed approach with the traditional one as shown in Table I. The minimum number of particle required for the traditional approach is 120, while our approach requires only 40 particles to build a consistent map. Moreover, the number of line segments in the map is 182, which is less than the case of the traditional approach, implying that our approach achieved a more consistent map with small number of particles. Here, the consistent map denotes a map where spurious and duplicated line segments are removed and also wall lines are well-aligned parallel to each other. The initial loop closing was occurred after the robot traveled about 50 m. It is considered that this long travel distance with unreliable estimates of line parameters before the loop closing caused the use of many particles in the traditional approach, whereas our approach succeeded in continual odometry error compensation and removed spurious lines through the online optimization, so it resulted in less number of particles with reduced number of line segments in the map (Fig. 8).
The experiments with known data set demonstrated that the proposed method gave reliable SLAM performance with compact representation of the environment.

REFERENCES


D. Final Map Representation

As a final step, the offline global optimization of the map was performed after the whole SLAM process was finished. Through this step, collinear lines were connected for a map consistency. For example, two well-aligned walls separated by a door were merged while blocking the door. However, data association scheme proposed in this paper is not affected by this block situation (i.e., we used only two end points to represent a line segment); thus it cannot affect to SLAM performance. The effectiveness of a global optimization was validated by showing the number of lines with increasing time index (Fig. 9).

VI. CONCLUSION

In this paper, an RBPF-SLAM using line segments has been presented. By using only two end points for a line representation, we pursued a more compact way of a line uncertainty modeling and an environment representation. With a modified scan point clustering method, an adaptive IEPF played an important role in reducing odometry error, so it paved the way for use of small number of particles without affecting a map consistency. The introduction of online optimization also made a contribution to a map consistency by removing spurious or duplicated lines and merging collinear lines. Along with this, all of the modifications to the traditional RBPF-SLAM proposed in this paper such as a line association were efficiently integrated into our RBPF-SLAM framework.

![Figure 9. The number of line segments in the map with increasing time index. The green square is a point in which global optimization was performed. The abrupt drop at last time index came from the offline global optimization. As a result, the number of line segments were reduced approximately 30% compared to the traditional approach.](image)

Table I. Performance Evaluation of the Proposed Method

<table>
<thead>
<tr>
<th>Data set</th>
<th>Method</th>
<th>Minimum # of particle</th>
<th># of line segments*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Lab</td>
<td>Traditional approach</td>
<td>120</td>
<td>216 (N/A)</td>
</tr>
<tr>
<td>Our approach</td>
<td></td>
<td>40</td>
<td>182 (146)</td>
</tr>
</tbody>
</table>

a. The number in parentheses indicates that the number of line segments after an offline global optimization.