Marija Đakulović, Šandor Ileš, Ivan Petrović

ISSN 0005-1144
ATKAFF 52(2), 118–131(2011)

Exploration and Mapping of Unknown Polygonal Environments Based on Uncertain Range Data

UDK 004.93'11
IFAC 2.5

Original scientific paper

We consider problem of exploration and mapping of unknown indoor environments using laser range finder. We assume a setup with a resolved localization problem and known uncertainty sensor models. Most exploration algorithms are based on detection of a boundary between explored and unexplored regions. They are, however, not efficient in practice due to uncertainties in measurement, localization and map building. The exploration and mapping algorithm is proposed that extends Ekman's exploration algorithm by removing rigid constraints on the range sensor and robot localization. The proposed algorithm includes line extraction algorithm developed by Pfister, which incorporates noise models of the range sensor and robot's pose uncertainty. A line representation of the range data is used for creating polygon that represents explored region from each measurement pose. The polygon edges that do not correspond to real environmental features are candidates for a new measurement pose. A general polygon clipping algorithm is used to obtain the total explored region as the union of polygons from different measurement poses. The proposed algorithm is tested and compared to the Ekman's algorithm by simulations and experimentally on a Pioneer 3DX mobile robot equipped with SICK LMS-200 laser range finder.

Key words: Exploration, Mapping, Line extraction, Mobile robot

Istraživanje i modeliranje nepoznatog poligonalnog prostora zasnovano na nesigurnim podacima udaljenosti. Razmatramo problem istraživanja i izgradnje karte nepoznatog unutarnjeg prostora koristeći laserski senzor udaljenosti. Pretpostavljamo riješenu lokalizaciju robota i poznati model nesigurnosti senzora. Većina se algoritama istraživanja zasniva na otkrivanju granica istraženog i neistraženog područja. Međutim, u praksi nisu učinkoviti zbog nesigurnosti mjerenja, lokalizacije i izgradnje karte. Razvijen je algoritam istraživanja i izgradnje karte koji proširuje Ekmanov algoritam uklanjanjem strogih ograničenja na senzor udaljenosti i lokalizaciju robota. Razvijeni algoritam uključuje algoritam izdvajanja linijskih segmenata prema Pfisteru, koji uzima u obzir utjecaje zašumljenosti senzora i nesigurnosti položaja mobilnog robota. Linijska reprezentacija podataka udaljenosti koristi se za stvaranje poligona koji predstavlja istraženo područje iz svakog mjernog položaja. Bridovi poligona koji se ne podudaraju sa stvarnim značajkama prostora su kandidati za novi mjerni položaj. Algoritam općenitog isijecanja poligona korišten je za dobivanje ukupnog istraženog područja kao unija poligona iz različitih mjernih položaja. Razvijeni algoritam testiran je i uspoređen s izvornim Ekmanovim algoritmom simulacijski i eksperimentalno na mobilnom robotu Pioneer 3DX opremljenim laserskim senzorom udaljenosti SICK LMS-200.

Ključne riječi: istraživanje, izgradnja karte, izdvajanje linijskih segmenata, mobilni robot

1 INTRODUCTION

The environment exploration problem is a part of a more general problem of robot motion planning. A robot equipped with a range sensor is required to autonomously navigate in a bounded unknown environment with the purpose of building its complete map. During exploration the robot is expected to traverse the shortest possible path. Therefore, a path planning algorithm must be used to calculate the optimal paths to the destination points that are provided by the exploration algorithm. Similarly, a path following algorithm, which directly controls the robot's motion, must be used to ensure that kinematic and dynamic constraints of the robot are fullfilled and to avoid the obstacles in the path.

Exploration strategies usually assume that a mobile robot is a single point in the unknown environment with a finite number of arbitrarily placed obstacles of different sizes and shapes. The problem of autonomous exploration of an unknown environment is considered separately from the problem of simultaneously localization in the same environment, i.e. the localization is assumed solved. Most exploration algorithms are not reliable when applied under the real conditions due to uncertainty of measurements, localization and map building.

This paper introduces a complete solution for exploration and mapping of an unknown environment. The exploration and mapping algorithm is proposed as an extension of the exploration strategy of polygonal environments developed by Ekman *et al.* [1], which assumes ideal range sensor and no pose uncertainty, combined with the line extraction algorithm developed by Pfister *et al.* [2], which incorporates noise models of the range sensor and robot's pose uncertainty.

The paper is structured as follows. Section 2 provides a brief review of existing exploration strategies with a more detailed description of the Ekman's algorithm. Section 3 describes the line fitting algorithm developed by Pfister *et al*. Section 4 presents the proposed exploration and mapping algorithm of an unknown environment, which comprises sensor noise model and robot's pose uncertainty and assures algorithm's convergence under the real conditions. In Section 5 experimental results of our algorithm are compared to the ones obtained by the original Ekman's algorithm under the same conditions. Finally, Section 6 gives the conclusions of the paper.

2 OVERVIEW OF THE EXPLORATION STRATEGIES

The papers by Brooks [3] and Oommen [4] are among the first ones to tackle the problem of exploration of unknown environments, also known as the problem of terrain acquisition.

Some exploration algorithms use assumptions on the shape of obstacles. Oommen [4] assumes that obstacles are convex polygonal objects. His exploration strategy consists of two elementary robot's actions: scanning to identify all visible vertices of the obstacles from the current measurement position, and moving the robot along a straight line. The robot moves from one to another vertex of the obstacle. To explore an environment with n obstacle vertices, the algorithm requires n scanning operations and, at most, $2(n-1)$ motions between the vertices.

The sightseer strategy developed by Lumelsky *et al.* [5] is based on visiting the obstacles of arbitrary shapes by moving around them. The strategy can be compared to the wall following strategy [6]. However, the sightseer's strategy performs poorly in complex indoor environments.

Most exploration strategies push the robot on the border between explored and unexplored region of the environment. Ekman *et al.* [1] present an exploration strategy for arbitrary polygonal environments that assumes a range sensor of finite angular resolution and thus provides a sampled version of the visibility polygon [7]. The exploration strategy is based on detection of discontinuities in the range data. These so-called jump edges separate explored and unexplored regions of the environment. Therefore, if robot takes measurements in front of the jump edge, a new region in the environment will be explored. A lack of jump edges indicates that the environment had been completely explored. Under certain conditions, resembling the Shannon sampling theorem, it is possible to generate a faithful map from the range data in a finite number of measurements. However, the exploration strategy [1] uses robot with no positional uncertainty and assumes that the range sensor is perfect.

Tovar *et al.* [8] present exploration strategy for polygonal environments that is also based on the discontinuities in the range data called *gaps*. A special dynamic data structure *gap navigation trees* is constructed. The idea is to minimize necessary information that the robot needs to collect to solve his task. The so-called gap sensor is used, which can be constructed from the range sensor, and it is assumed that the gap sensor is the only sensor used by the robot. All exploration actions are reduced to "chasing gaps". The map representation is encoded according to the robot's position as the tree of critical events – gaps can disappear, appear, split or merge due to robot's motion. By analyzing the changes of critical events in the gap sensor the gap navigation tree is constructed, which denotes optimal movements in the environment, although the precise measurements of the environment are not taken. The strategy does not require exact localization and complete map building. However, some practical aspects remain unsolved: it can happened that the robot is chasing the wrong gap due to the noisy gap sensor readings.

Yamauchi *et al.* [9] called their strategy the frontier based exploration strategy. The strategy does not make assumption on the shape of the obstacles and is applicable under the real conditions. The occupancy grid map is used for storing new information about the environment, and computer vision technique is used for extraction of frontiers between explored and unexplored regions. The strategy determines the closest point in frontiers as the next pose of the robot. The main drawback is the high computational cost and memory usage for determining frontiers.

The exploration strategy developed in [10] is based on the detection of occlusion points in range data, which represent borders between explored and unexplored regions. An infinite range line of sight sensor and polygonal obstacles are assumed. The map representation is composed only of occlusion points. Robot needs to visit all occlusion points by moving around them. Thereby, by storing and detecting only occlusion points the computational cost is lowered significantly. However, in cases when robot is surrounded by obstacles with no visible non-visited occlusion point around the strategy may end up in a local minima. A similar strategy is the "view improvement" strategy [11], which is also based on the detection of occlusion points and assumes ideal localization and measurements.

Since the Ekman's strategy makes the basis of our exploration and mapping algorithm we describe it in more detail.

2.1 Ekman's strategy

The information about environment is presented at two abstraction levels. At the lower level, a polygonal representation of the environment is used, and at the higher level, a graph representation (*exploration graph*) is used for path planning.

2.1.1 Polygonal representation

Polygon is a planar object composed of N, $N \geq 3$, straight line segments $\overline{v_1v_2}, \overline{v_2v_3}, \ldots, \overline{v_Nv_1}$. Points and line segments will be referred to as vertices and edges, respectively. Two edges with common vertex are adjacent. A polygon is simple if only adjacent edges intersect. A polygon is simply-connected if every simple closed curve in the polygon can be continuously shrunk to a point without leaving the polygon. Otherwise, the polygon is multiply-connected (i.e., the polygon contains holes). A faithful polygonal description of the environment is noted by P_F. Two points p_i and p_j in a polygon are *visible* if and only if all points along the straight line between p_i and p_j belong to the polygon. This binary relation will be referred to as the *visibility* relation, denoted by *Visible*(p_i, p_j). By treating p_i or p_j as a variable, the visibility polygon is obtained. The visibility polygon of p_i can be expressed as $VP_i = \{p_j \mid \textit{Visible}(p_i, p_j)\}$.

From each measurement pose p_i (position plus orientation) range is measured in n uniformly distributed directions. The laser range finder measures distance to the nearest obstacle in $n = 361$ directions ranging from -180° to 180°. Let the range data sequence obtained from a measuring pose p_i be denoted as $\mathcal{R} = \{r_k\}$, $k = 1, \ldots, n$. The scan points coordinates are written in the local (robot) coordinate frame in the polar form as (d_k, Θ_k), or in Cartesian coordinates as:

$$r_k = d_k \begin{bmatrix} \cos(\Theta_k) \\ \sin(\Theta_k) \end{bmatrix}, \tag{1}$$

where d_k is the real distance to the obstacle and Θ_k is angle referring to x-coordinate axis of the local frame. The range data sequence $\mathcal{R}$ is an ordered set, where $\Theta_0 = -180^\circ$ and $\Theta_n = 180^\circ$. Since the range data sequence is a discrete representation of the continuous visibility polygon, a criterion is derived resembling the Shannon sampling theorem [1]. Let $S_m = \{s_1, s_2, \ldots, s_m\}$, $m \geq 3$, denote an ordered subset of $\mathcal{R}$. A subset, where all sample points are collinear, is said to have the *collinearity property*. Any two consecutive sample points s_i, s_{i+1} are said to be *close* if $\|s_i - s_{i+1}\| < \Delta s$, where Δs is some preset value. Otherwise, two points are said to be *distant*. A subset, where every two consecutive sample points are close, is said to have the *proximity property*. The conjunction of the Proximity and Collinearity properties is called the *PC* property. Accordingly, subsets having the *PC* property are called *PC*-subsets. In environments that are incompatible with the value of Δs, spurious *PC*-subsets may appear (Fig. 1). Let d_c denotes the Euclidean distance between the two closest nonadjacent edges in the environment. If the environment satisfies the constraint $d_c \geq 2\Delta s$ spurious *PC*-subsets will never appear [1]. Preset value Δs gives constraint on the size of the detected objects in the environment.

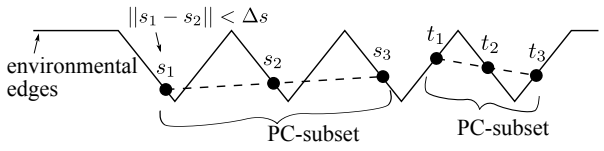

Fig. 1. Two spurious PC-subsets.

The *measurement polygon* P is created from the range data $\mathcal{R}$ taken at the pose p_i, and is represented by the tuple (L, B), where $L = \{l_j\}$ is the set of explored environmental edge sections (l-edges), and $B = \{b_k\}$ is the set of bounding edges (b-edges), which connect sample points and measurement position. The measurement polygon is created from the range data in two steps: first, the range data set is transformed into a star-shaped polygon, then, certain operations of expansion and fusion are performed to transform it into the measurement polygon, by which the l and b-edges are determined. These operations are not used in our work and the details can be found in [1].

The *extended measurement polygon EP* is created from the measurement polygon P, and is represented by the tuple (L, B, H), where $H = \{h_j\}$ contains jump edges (h-edges) defined between the non-common endpoints of every b-edge pair from P.

The *exploration polygon* $P_E(i)$ is represented by the tuple $(L_E(i), B_E(i))$ and is created by adding the corresponding edges of the measurement polygon P to the exploration polygon $P_E(i-1)$ from the previous measurement position p_{i-1}, where $P_E(0)$ is the empty polygon. The faithful exploration polygon $P_E(F)$ (at final step $i = F$) has an empty set $B_E(F)$, i.e., it is represented by $P_E(F) = (L_E(F), \emptyset)$. If a b-edge from $P_E(i-1)$ intersects a b-edge from P, four b-edge pairs are defined, where all eight b-edges have a common point, and operations of expansion and fusion are performed again, by which the new l and h-edges are defined.

The *extended exploration polygon* $EP_E(i)$ is represented by the tuple $(L_E(i), B_E(i), H_E(i))$ and is created

by introducing jump edges between the non-common endpoints of every b-edge pair from $P_E(i)$. If the nonempty extended exploration polygon $EP_E(i)$ contains no jump edges, then it is equal to the faithful polygonal description $P_E(F)$ [1].

2.1.2 Graph representation

The *exploration graph* $\mathcal{G}_E(i) = (\mathcal{N}, \mathcal{E})$ is a connected, undirected graph, where $\mathcal{N} = \{n\}$ is a set of *nodes* and $\mathcal{E} = \{e_{i,j}\}$ is a set of *edges*. Nodes correspond to measurement poses while edges are defined between visible nodes. Every node n has an assigned gain value g, and every edge e_{ij} has an assigned cost value c_{ij}. The gain value represents the predicted information gain of a measurement from the corresponding measurement pose, while the cost value represents the cost of moving between the positions corresponding to the edge endpoints. The exploration graph $\mathcal{G}_E(i)$ is created by adding new nodes and edges to the exploration graph $\mathcal{G}_E(i-1)$, where $\mathcal{G}_E(0) = \{S, \emptyset\}$ (the node S is the start node of the exploration algorithm) for each h-edge in the extended exploration polygon $EP_E(i)$. Namely, a point near the mid point of the h-edge is defined as the new candidate measurement node and added to $\mathcal{G}_E(i)$. Thereafter, the visibility relation is verified for all nodes in $\mathcal{G}_E(i)$ and new edges are accordingly introduced. Finally, cost and gain values in $\mathcal{G}_E(i)$ are updated.

Next measurement pose p_{i+1} is chosen from the exploration graph $\mathcal{G}_E(i)$ created at the current measurement pose p_i according to the selection criterion, which is defined as:

$$C(p_i, p_{i+1}) = \frac{g_{i+1}}{c_{i,i+1}}, \tag{2}$$

where $c_{i,i+1}$ is the cost of the edge $e_{i,i+1}$ if nodes p_i and p_{i+1} are visible, otherwise $c_{i,i+1}$ is the sum of the edge costs along the least expensive path between nodes p_i and p_{i+1}, and g_{i+1} is the gain value of a candidate node p_{i+1} defined by all jump edges visible form the candidate node p_{i+1}:

$$g_{i+1} = \sum_k \alpha_{i+1,k}, \tag{3}$$

where $\alpha_{i+1,k}$ is the angle at the node p_{i+1} in the triangle defined by k-th jump edge (h_k) and the node p_{i+1}. This is illustrated in Fig. 2, where measurement polygon P contains two h-edges (h_1 and h_2), and two candidate measurement poses (p_1 and p_2). The node with the highest C value is selected as the new measurement pose p_{i+1}.

Pseudo code of the exploration strategy is given by Algorithm 1. During the first iteration of the algorithm, the nodes p_0 and p_1 are the start pose S and procedure **move**() does nothing. Exploration polygon is initialized to the empty polygon, and the exploration graph is initialized to the initial pose p_0. The measurement $\mathcal{R}$ from the first pose p_1 is taken and the measurement polygon P is created. The exploration polygon is created from the measurement polygon P, and then, the extended exploration polygon is created from the exploration polygon by adding jump edges. The exploration graph is created from the extended exploration polygon and the next measurement pose is determined according to the highest C value. The exploration algorithm terminates after the k-th measurement if all nodes have zero gain values.

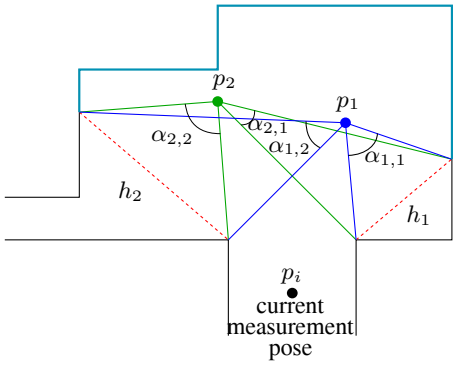

Fig. 2. Selection criterion based on the gain values.

Algorithm 1 Ekman's exploration strategy: **explore**(S)

Input: Starting point S
Output: A sequence of move commands

1. $p_0 \leftarrow S$ *//Initialization*
2. $p_1 \leftarrow S$
3. $P_E(0) \leftarrow (\emptyset, \emptyset)$
4. $\mathcal{G}_E(0) \leftarrow (p_0, \emptyset)$
5. $i \leftarrow 1$
6. **repeat**
7. **move**(p_{i-1}, p_i)
8. $\mathcal{R} \leftarrow$ **new-measurement**(p_i)
9. $P \leftarrow$ **measurement-polygon**($\mathcal{R}$)
10. $P_E(i) \leftarrow$ **exploration-polygon**($P_E(i-1)$, P)
11. $EP_E(i) \leftarrow$ **extended-exploration-polygon**($P_E(i)$)
12. $\mathcal{G}_E(i) \leftarrow$ **exploration-graph**($\mathcal{G}_E(i-1)$, $EP_E(i)$)
13. $p_{i+1} \leftarrow$ **select-node**(p_i, $\mathcal{G}_E(i)$)
14. $i \leftarrow i+1$
15. **until** $g_i = 0$

3 LINE SEGMENT EXTRACTION FROM RANGE DATA

Ekman's algorithm assumes ideal localization and ideal laser range finder. However, when working with real noisy range data, collinearity property is hard to fulfill. Ekman's algorithm also assumes that visibility polygons from

two adjacent measurement positions are perfectly matched. Therefore, a statistically sound estimation of line segments and merging line segments from different measurement poses based on sensor and localization uncertainty must be performed.

Feature extraction is an important task in localization and mapping of unknown environments. The line segment representation reduces the map representation size and complexity. Various method for line extraction from range data can be found in literature, [12], [13], [14]. Some authors use the Hough transform for line extraction from range or sonar data [15]. Although this method can be used for detecting a wide range of features, for map building purpose it is used as a line detector. However, the Hough transform does not take into account sensor noise and uncertainty. A Kalman filter based approach in [16] allows only uniform range data weighting. Pfister *et al.* [2] consider a weighted line fitting of range data with uncertainty.

In this paper Phister's weighted line fitting algorithm is used, with range data and localization uncertainty taken into account. Algorithm is based on the Hough transform, weighted least square and χ^2 test. Instead of the standard Hough transform we used the Progressive Probabilistic Hough Transform (PPHT) with post-processing [17], since it uses just a fraction of points for line detection. Our method finds line segments instead of lines, and has less false detections. To reduce number of lines, similar lines, e.g. the lines that represent same common walls, are detected. Various methods for merging partially overlapping line segments are presented in [18]. Uncertainty modeling is a key problem in grouping and merging similar features. For hypothesis validation a χ^2 test is used. Its effectiveness depends on a correct uncertainty modeling.

The main goal of the line segments extraction algorithm is detection and grouping of range data $\mathcal{R} = \{r_k \mid k = 1, \ldots, n\}$ into M subsets, where number M is not known a-priori. Every subset should contain almost collinear points within some error limit. For every subset the optimal line segment S, with corresponding covariance matrix Q_S is obtained. An iterative procedure for extracting line segments from range data is defined as follows:

1. **Range sensor noise modeling.** Range data points are modeled with Gaussian distribution.
2. **Initial line segment estimation.** For a given range data set $\mathcal{R}$ the initial line segment estimation, $\hat{S}_i$, is found using the Probabilistic Hough Transform.
3. **Data grouping.** For every line segment $\hat{S}_i$, obtained in the step 2, subset of points $\mathcal{R}^i_{\hat{S}}$, where $\mathcal{R}^i_{\hat{S}} \in \mathcal{R}$ is determined as the set of points along the line segment $\hat{S}_i$.
4. **Weighted line fitting algorithm.** For every set $\mathcal{R}^i_{\hat{S}}$ the optimal line segment $S_i = [\alpha, \rho, \psi_a, \psi_b]$ is computed using the maximum likelihood formulation [2].
5. **Line covariance matrix estimations.** For every line segment S_i covariance matrix Q_{S_i} is computed.
6. **Merging similar line segments.** To reduce number of line segments, similar line segments are merged. For hypothesis validation χ^2-test is used.

3.1 Range sensor noise modeling

Given the range data set $\mathcal{R} = \{r_k\}$ given by (1), the measured distance $\hat{d}_k$ is computed by the noise ϵ_{d_k}:

$$\hat{d}_k = d_k + \epsilon_{d_k}. \tag{4}$$

Similarly, the measured angle $\hat{\Theta}_k$ is computed by the noise ϵ_{Θ_k}:

$$\hat{\Theta}_k = \Theta_k + \epsilon_{\Theta_k}. \tag{5}$$

Therefore, the measured point can be written as:

$$\hat{r}_k = (d_k + \epsilon_{d_k}) \begin{bmatrix} \cos(\Theta_k + \epsilon_{\Theta_k}) \\ \sin(\Theta_k + \epsilon_{\Theta_k}) \end{bmatrix}. \tag{6}$$

We assume that every error has a Gaussian distribution with the mean value equal zero and corresponding covariance σ_d^2 i σ_Θ^2. Assuming that $\epsilon_{\Theta_k} \ll 1^\circ$ then the following approximations can be used:

$$\begin{aligned} \sin(\epsilon_{\Theta_k}) &\approx \epsilon_{\Theta_k}, \\ \cos(\epsilon_{\Theta_k}) &\approx 1, \\ \epsilon_{\Theta_k} \cdot \epsilon_{d_k} &\approx 0. \end{aligned} \tag{7}$$

Assuming that ϵ_{d_k} and ϵ_{Θ_k} are independent, covariance matrix of a measured point r_k is given as:

$$\begin{aligned} Q_{r_k} = &\frac{d_k^2 \sigma_\Theta^2}{2} \begin{bmatrix} 2\sin^2(\Theta_k)) & -\sin(2\Theta_k) \\ -\sin(2\Theta_k) & 2\cos^2(\Theta_k) \end{bmatrix} \\ &+ \frac{\sigma_d^2}{2} \begin{bmatrix} 2\cos^2(\Theta_k) & \sin(2\Theta_k) \\ \sin(2\Theta_k) & 2\sin^2(\Theta_k) \end{bmatrix}. \end{aligned} \tag{8}$$

For practical computations $\hat{\Theta}_k$ and $\hat{d}_k$ are used instead of Θ_k i d_k for point covariance matrix estimation.

3.2 Initial line segment estimation

Polar representation of a line allows easy comparison of line orientation and normal position. The line can be written in polar form as follows:

$$L = \begin{bmatrix} \rho & \alpha \end{bmatrix}^T, \tag{9}$$

where $\rho \geq 0$ defines the normal distance to the origin and $-\pi \leq \alpha \leq \pi$ defines orientation of the line, see Fig. 3.

The line is given in the Cartesian coordinate system as follows:

$$x\cos(\alpha) + y\sin(\alpha) = \rho. \tag{10}$$

Line segment representation builds upon the line representation by trimming the line at extreme endpoints. For a given line with orientation α, coordinate frame ρ_c-ψ_c is obtained by rotating the origin frame for angle α, see Fig. 3. The ends of a line segment are represented as a scalar values ψ_a and ψ_b in the ρ_c-ψ_c coordinate frame. The line segment is defined as follows:

$$S = \begin{bmatrix} \rho & \alpha & \psi_a & \psi_b \end{bmatrix}^T. \tag{11}$$

Note that segment S can be augmented with additional endpoint pairs to represent multiple segments that share the same underlying infinite line.

$$S = \begin{bmatrix} \rho & \alpha & \psi_{a1} & \psi_{b1} & \cdots & \psi_{an} & \psi_{bn} \end{bmatrix}^T, \tag{12}$$

where n is the number of endpoint pairs.

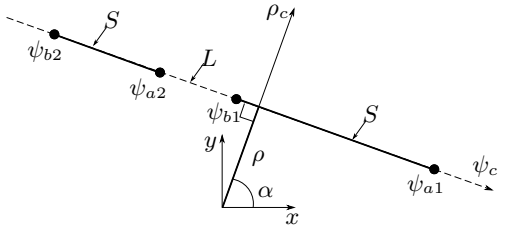

Fig. 3. Two line segments S on the same line L.

For initial line segment estimation PPHT is used. Repeatedly, a new random point is selected for voting, after voting, points that are supporting evidence for the detected features are being removed. The computation stops when all the points have either voted or have been assigned to a feature. Algorithm can be interrupted anytime still providing salient features, which can be useful in real-time applications.

3.3 Data grouping

Let initial line segment estimation be denoted as $\hat{S}$, with corresponding parameters $\hat{\rho}$ and $\hat{\alpha}$. The set of points that correspond to that line segment is denoted as $\mathcal{R}_{\hat{S}}$. The set $\mathcal{R}_{\hat{S}}$ is determined by computing the distance between every measurement point $\hat{r}_k \in \mathcal{R}$ and the line segment

$$\delta\rho_k = \hat{d}_k \cos(\hat{\alpha} - \hat{\Theta}_k) - \hat{\rho}. \tag{13}$$

The subset of points $\mathcal{R}_{\hat{S}}$ is determined as

$$\mathcal{R}^i_{\hat{S}} := \{x \in \mathcal{R} \,|\delta\rho_k < \Delta\rho\}, \tag{14}$$

where $\Delta\rho$ is a discretization step in ρ coordinate of the Hough space, see Fig. 4.

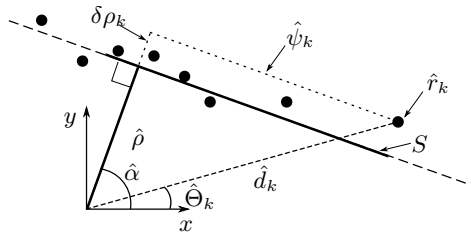

Fig. 4. Candidate line segment S and range measurement points.

3.4 Weighted line fitting algorithm

In the following we summarize the line estimation from [2, 19, 20]. After initial line estimation using Hough transform, line parameters are computed using maximum likelihood formulation:

$$\mathcal{L}(\alpha, \rho) = \sum_{k=1}^{n} \frac{(\delta\rho_k)^2}{Q_{\delta\rho_k}}, \tag{15}$$

where $\delta\rho_k$ is given by (13), and the covariance matrix $Q_{\delta\rho_k}$ of the error distance $\delta\rho_k$ is derived from the covariance matrix Q_{r_k} of a measured point r_k, given by (8), as:

$$Q_{\delta\rho_k} = \begin{bmatrix} \cos\hat{\alpha} & \sin\hat{\alpha} \end{bmatrix} Q_{r_k} \begin{bmatrix} \cos\hat{\alpha} & \sin\hat{\alpha} \end{bmatrix}^T. \tag{16}$$

The estimate of the parameter ρ is derived by minimizing (15):

$$\hat{\rho} = \frac{\sum_{k=1}^{n} \dfrac{\hat{d}_k \cos(\hat{\alpha} - \hat{\Theta}_k)}{Q_{\delta\rho_k}}}{\sum_{k=1}^{n} \dfrac{1}{Q_{\delta\rho_k}}}. \tag{17}$$

There is no exact solution for the parameter α since the minimization of (15) is an nonlinear optimization problem. In [2] developed the second order iterative solution. The calculation of the line heading α depends on the reference frame in which the calculation is performed. In [19] the calculation is performed in the reference frame at the *center of rotational uncertainty* ψ_P in which angle uncertainty ϵ_α and distance uncertainty ϵ_ρ are independent random variables, see Fig. 5. The value ψ_P is calculated such that about that value, the weighted measurement points are balanced:

$$\hat{\psi}_P = \frac{\sum_{k=1}^{n} \dfrac{\hat{\psi}_k}{Q_{\delta\rho_k}}}{\sum_{k=1}^{n} \dfrac{1}{Q_{\delta\rho_k}}}, \tag{18}$$

where $\hat{\psi}_k$ is the measurement point position along the line calculated as

$$\hat{\psi}_k = \hat{d}_k \sin(\hat{\alpha} - \hat{\Theta}_k). \tag{19}$$

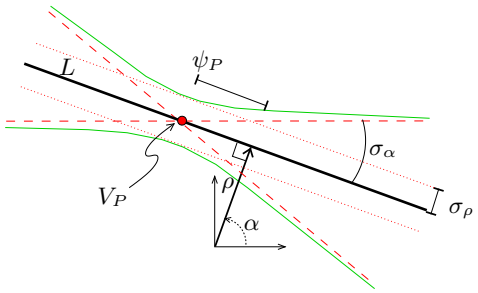

Fig. 5. Covariance representation of the line and the center of rotational uncertainty ψ_P.

Let

$$\delta\psi_k = \hat{\psi}_k - \hat{\psi}_P \tag{20}$$

be the distance between k-th measured point and estimated center of rotational uncertainty. The estimate for the parameter α is updated as $\alpha = \hat{\alpha} + \delta\alpha$, where

$$\delta\alpha = -\frac{\sum_{k=1}^{n} \frac{\delta\rho_k \delta\psi_k}{Q_{\delta\rho_k}}}{\sum_{k=1}^{n} \frac{(\delta\psi_k)^2}{Q_{\delta\rho_k}}}, \tag{21}$$

with $\delta\rho_k$, $Q_{\delta\rho_k}$ and $\hat{\psi}_k$ defined by (13), (16), and (19), respectively. The optimal line segments are determined by projection of initial estimation of endpoints to the estimated line. For simplicity the endpoint uncertainty is ignored.

3.5 Line covariance matrix estimations

The estimated line orientation $\hat{\alpha}$ is computed by the noise ϵ_α:

$$\hat{\alpha} = \alpha + \epsilon_\alpha. \tag{22}$$

Similarly, the estimated normal distance $\hat{\rho}$ is computed by the noise ϵ_ρ:

$$\hat{\rho} = \rho + \epsilon_\rho. \tag{23}$$

We assume that ϵ_ρ and ϵ_α are zero-mean, Gaussian random variables with corresponding variances σ_ρ^2 and σ_α^2. The line covariance matrix is defined as follows:

$$Q_L = \mathrm{E}\left[(\epsilon_L)(\epsilon_L)^T\right], \tag{24}$$

where E is the expectation operator and $\epsilon_L = \begin{bmatrix} \epsilon_\alpha & \epsilon_\rho \end{bmatrix}^T$. Consequently, the line covariance matrix can be computed as:

$$Q_L = \begin{bmatrix} \mathrm{E}\left[\epsilon_\alpha^2\right] & \mathrm{E}\left[\epsilon_\alpha\epsilon_\rho\right] \\ \mathrm{E}\left[\epsilon_\alpha\epsilon_\rho\right] & \mathrm{E}\left[\epsilon_\rho^2\right] \end{bmatrix} = \begin{bmatrix} Q_{\rho\rho} & Q_{\rho\alpha} \\ Q_{\rho\alpha} & Q_{\alpha\alpha} \end{bmatrix}, \tag{25}$$

where $Q_{\rho\rho}$ is the distance variance, $Q_{\alpha\alpha}$ is the line orientation variance and $Q_{\rho\alpha}$ and $Q_{\alpha\rho}$ are corresponding covariances. By definition covariance matrix is symmetric positive definite: $Q_{\rho\alpha} = Q_{\alpha\rho}$. According to the weighted line fitting algorithm, line covariance matrix is computed as follows:

$$Q_{\rho\rho} = \frac{1}{\sum_{k=1}^{n} \frac{1}{Q_{\delta\rho_k}}}, \tag{26}$$

$$Q_{\alpha\alpha} = \frac{1}{\sum_{k=1}^{n} \frac{(\delta\psi_k)^2}{Q_{\delta\rho_k}}}, \tag{27}$$

$$Q_{\rho\alpha} = -Q_{\alpha\alpha} Q_{\rho\rho} \sum_{k=1}^{n} \frac{\delta\psi_k}{Q_{\delta\rho_k}}, \tag{28}$$

where $\delta\rho_k$, $\delta\psi_k$ and $Q_{\delta\rho_k}$ are given by (13), (20) and (16).

3.5.1 Odometry noise model

The robot's pose referring to the fixed reference frame p_0 is given by:

$$p_i = \begin{bmatrix} x_i \\ y_i \\ \vartheta_i \end{bmatrix}, \tag{29}$$

Pair (x_i, y_i) determines position and ϑ_i robot's orientation referring to positive direction on x-axis. The odometry system estimates relative changes in the robot's pose by integrating over the internally measured wheel motion. If the robot starts at the pose p_i and moves to the pose p_j the resulting local displacement measurement with respect to pose p_i is denoted as p_{ij}. In general, the covariance matrix of p_{ij} is given as follows:

$$Q_{p_{ij}} = \begin{bmatrix} Q_{xx} & Q_{xy} & Q_{x\vartheta} \\ Q_{yx} & Q_{yy} & Q_{y\vartheta} \\ Q_{\vartheta x} & Q_{\vartheta y} & Q_{\vartheta\vartheta} \end{bmatrix}. \tag{30}$$

The form of the actual covariance matrix depends on the model of the odometry method being used. Assuming small displacement p_{ij} and noise in x, y and ϑ are independent random variables:

$$Q_{p_{ij}} = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\vartheta^2 \end{bmatrix}, \tag{31}$$

where σ_x^2, σ_y^2 and σ_ϑ^2 corresponding position variance x and y and orientation variance ϑ.

Let the initial position be p_i, and corresponding covariance matrix Q_{p_i}, and local displacement p_{ij} with corresponding local covariance matrix $Q_{p_{ij}}$. The combined covariance Q_{p_j} in the global frame can be calculated as:

$$Q_{p_j} = G Q_{p_i} G^T + K Q_{p_{ij}} K^T, \tag{32}$$

where

$$G = \begin{bmatrix} 1 & 0 & -y_{ij}\cos\vartheta_i - x_{ij}\sin\vartheta_i \\ 0 & 1 & -y_{ij}\sin\vartheta_i + x_{ij}\cos\vartheta_i \\ 0 & 0 & 1 \end{bmatrix} \tag{33}$$

and

$$K = \begin{bmatrix} \cos\vartheta_i & -\sin\vartheta_i & 0 \\ \sin\vartheta_i & \cos\vartheta_i & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{34}$$

3.5.2 Covariance transformations

Consider an infinite line measured in i-th local reference frame $p_i = \begin{bmatrix} x_i & y_i & \vartheta_i \end{bmatrix}^T$, where $L_i = \begin{bmatrix} \alpha_i & \rho_i \end{bmatrix}^T$. The line transformed into global reference frame is denoted as L_0,

$$L_0 = \begin{bmatrix} \alpha_0 \\ \rho_0 \end{bmatrix} = \begin{bmatrix} \alpha_i + \vartheta_i \\ \rho_i + \delta\rho_i \end{bmatrix}, \tag{35}$$

and the corresponding covariance matrix is denoted as Q_{L_0}, which is given as a sum of transformed line covariance matrix and projection of the pose covariance matrix:

$$Q_{L_0} = H_i Q_{L_i} H_i^T + K_i Q_{p_i} K_i^T, \tag{36}$$

given by the following expressions:

$$H_i = \begin{bmatrix} 1 & 0 \\ \delta\psi_i & 1 \end{bmatrix}, \tag{37}$$

$$K_i = \begin{bmatrix} 0 & 0 & 1 \\ \cos(\alpha_i + \vartheta_i) & \sin(\alpha_i + \vartheta_i) & 0 \end{bmatrix}, \tag{38}$$

$$\delta\rho_i = x_i\cos(\alpha_i + \vartheta_i) + y_i\sin(\alpha_i + \vartheta_i), \tag{39}$$

$$\delta\psi_i = y_i\cos(\alpha_i + \vartheta_i) - x_i\sin(\alpha_i + \vartheta_i), \tag{40}$$

Fig. 6 shows line covariance transformed into global coordinate frame, and Fig. 7 shows pose covariance ellipse Q_{p_i} transformed into projection in the direction of the line normal $Q_{\rho\rho}$.

It is important to note that transformed line covariance matrix into the global coordinate frame by a transformation H_i is a translation transformation and that the eigenvalues of the covariance matrix are independent of the reference pose. For any line covariance matrix Q_L there exists a special value of H_P, which diagonalizes the covariance matrix Q_L as follows:

$$H_P = \begin{bmatrix} 1 & 0 \\ \psi_P & 1 \end{bmatrix}, \tag{41}$$

$$Q_L = \begin{bmatrix} 1 & 0 \\ \psi_P & 1 \end{bmatrix} \begin{bmatrix} \sigma_\alpha^2 & 0 \\ 0 & \sigma_\rho^2 \end{bmatrix} \begin{bmatrix} 1 & \psi_P \\ 0 & 1 \end{bmatrix}. \tag{42}$$

where ψ_P is the center of rotational uncertainty defined by

$$\psi_P = -\frac{Q_{\rho\alpha}}{Q_{\alpha\alpha}}, \tag{43}$$

which estimated value is given by (18).

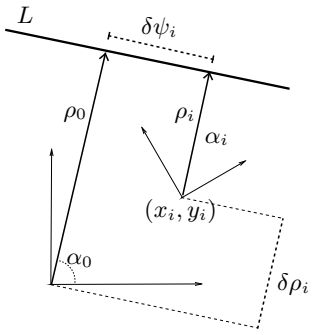

Fig. 6. The line referring to global and local reference frames. Values $\delta\rho$ and $\delta\psi$ represent reference frame displacement referring to ρ-ψ reference frame.

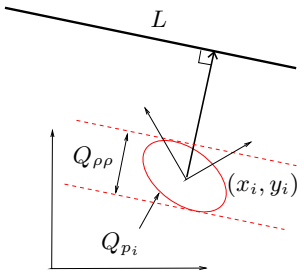

Fig. 7. Position uncertainty projected onto line uncertainty.

3.6 Merging similar line segments

Only most robust parameters of the line segment are compared: orientation α and normal distance ρ. A χ^2 test is used as the basis for the hypothesis test. The hypothesis say that two line segments lie on the same underlying line. Given two line segments S_0^i and S_0^j, first the underlying lines L_0^i and L_0^j are extracted from (11). Relative uncertainty of two line difference is calculated as follows:

$$Q_{\delta L} = Q_{L_0^i} + Q_{L_0^j}. \tag{44}$$

The lines are represented with respect to a reference frame from which the cross-coupling is minimized, i.e., at the center of rotational uncertainty defined by (18). Lines L_0^i and L_0^j and corresponding covariance matrix $Q_{\delta L}$ are transformed into $\tilde{L}_0^i$, $\tilde{L}_0^j$ and $\tilde{Q}_{\delta L}$ according to (41). Then, the following test is performed:

$$\delta\tilde{L} = \begin{bmatrix} \delta\tilde{\alpha} \\ \delta\tilde{\rho} \end{bmatrix} = \begin{bmatrix} \tilde{\alpha}_0^i - \tilde{\alpha}_0^j \\ \tilde{\rho}_0^i - \tilde{\rho}_0^j \end{bmatrix}, \tag{45}$$

$$\tilde{\chi}_2^2 = (\delta\tilde{L})^T(\tilde{Q}_{\delta L})^{-1}(\delta\tilde{L}) < 3. \tag{46}$$

If condition (46) is true it can be determined within chosen probability p that the difference between lines L_0^i and L_0^j can be explained by the modeled noise and the hypothesis

is true. A maximum likelihood approach is then used to determine the best estimate of the line pairs to be merged:

$$Q_{Ls} = ((Q_{L_0^i})^{-1} + (Q_{L_0^j})^{-1})^{-1}, \quad (47)$$

$$L_s = Q_{Ls}((Q_{L_0^i})^{-1} L_0^i + (Q_{L_0^j})^{-1} L_0^j). \quad (48)$$

Endpoints are obtained by projecting existing endpoints on the new line. Since, for simplicity, the uncertainty of endpoints is ignored, two line segments sharing the same underlying line whose endpoints have distance less than $3\sigma_d$ are merged into one line segment.

4 POLYGONAL ENVIRONMENT EXPLORATION ALGORITHM

The proposed exploration algorithm is an extension of the Ekman's algorithm [1]. The extension includes changed selection criterion and the Pfister's line extraction algorithm for polygon creation [2], which incorporates noise models of the range sensor and robot's pose uncertainty.

4.1 Selection criterion of the next measurement pose

In previous work [21] we proposed selection criterion for the next measurement pose that uses the ratio of the gain in the candidate node p_i and the cost of the path from the current measurement node p_{i-1} to the node p_i. Division of the gain value by a factor r can be considered as multiplying the path cost by a factor $1/r$. Therefore, it can happen that further candidate node with higher gain value is selected rather than closer candidate node with lower gain. Since all nodes in the exploration graph need to be visited, the robot must later return for visiting nodes with low gain values. In an indoor environments with a lot of doors, desks and passages, this situation could happen very often, which will result in unsatisfactory too long robot's motion.

To prevent such situations, we changed the selection criterion in this work. Instead of using the ratio, we use the sum of gain and cost normed by the length of the visible radius R_v of the sensor range. The gain value g_i assigned to the candidate measurement node p_i can maximally be $180°$ (see Fig. 2) and placing the measurement pose should not be too far from the jump edge due to the limit of the sensor range. We defined the gain value as the length of the jump edge since it greatly determines the size of the unexplored environment. The selection criterion is defined as:

$$C(p_{i-1}, p_i) = \frac{|h_{i-1}|}{R_v} + \frac{R_v}{c_{i-1,i}}, \quad (49)$$

where h_{i-1} is the jump edge in front of the node p_i. This form of selection criterion prefers similarly longer jump edges and shorter paths to the candidate measurement positions.

4.2 Creation of polygons from line segments

The original exploration algorithm [1] creates the measurement polygon P from the range data $\mathcal{R}$ by operations of expansion and fusion of the star-shaped polygon. When working with real noisy range data, the collinearity property is hard to fulfill. In such situations one can expect to detect as many edges as there are range points.

The proposed exploration algorithm uses the line extraction algorithm, which finds statistically correct edges from the noisy sensor readings and includes robot's pose uncertainty. Calculated line segments create the polygon as follows. The ending points of the line segments are sorted according to increasing angle in the polar coordinate system from $[-\pi, \pi]$. Such arranged ending points $s_1, \ldots, s_n$ define extended measurement polygon. Specially, if the robot's pose p_i is not collinear with $\overline{s_n s_1}$, the extended measurement polygon is closed over the robot's pose, i.e., instead of segment $\overline{s_n s_1}$ two end segments are created $\overline{s_n p_i}$ and $\overline{p_i s_1}$. Line segments correspond to l-edges (the set $L_i = \{l_{i,j}\}$) of the extended measurement polygon, and edges needed to enclose the polygon correspond to the h-edges of the extended measurement polygon. For clarity, extended measurement polygon is noted as $\widetilde{EP}$.

If line segments from range data could not be created (e.g. all obstacles are out of range), jump edges are defined in order to construct a triangle whose vertices lie on the circle of radius R_v (maximal sensor range) with the center at the measurement position. One jump edge pass through the center of the circle connecting points at $-\pi$ and π in the polar coordinate system, and the other two jump edges connect ending points of the first with the point on the circle in front of the robot, as shown in Fig. 8.

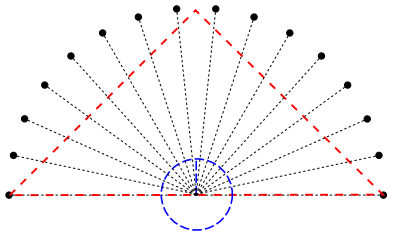

Fig. 8. Creation of jump edges when all obstacles are out of range.

Fig. 9 shows an example of determination of extended measurement polygon of the part of an unknown environment. The real map which is unknown to the robot is shown by dots. Dashed lines represent jump edges, and solid lines represent line segments extracted from the range data. The current measurement pose is p_i, and the chosen next measurement pose is m. The pose m is located in

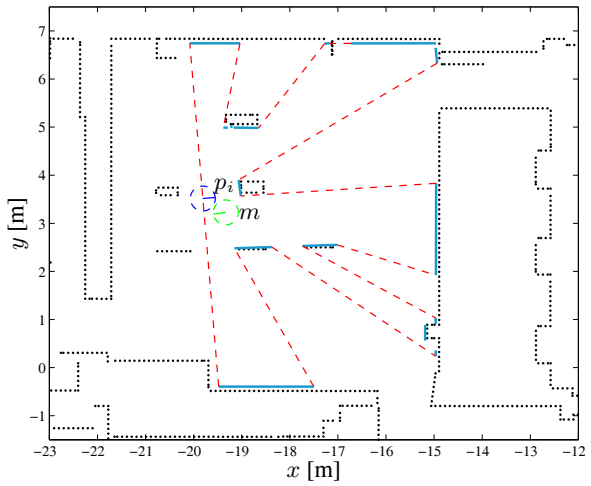

Fig. 9. An example of determination of jump edges (dashed lines) and environmental edges (solid lines) from the range data taken at the pose p_i.

front of the jump edge to ensure partial overlap of measurement polygons. Jump edges together with line segments create the extended measurement polygon $\widetilde{EP}$.

Fig. 10 shows the robot taking the next measurement scan from the best measurement pose according to the selection criterion and configuration presented in Fig. 9. The

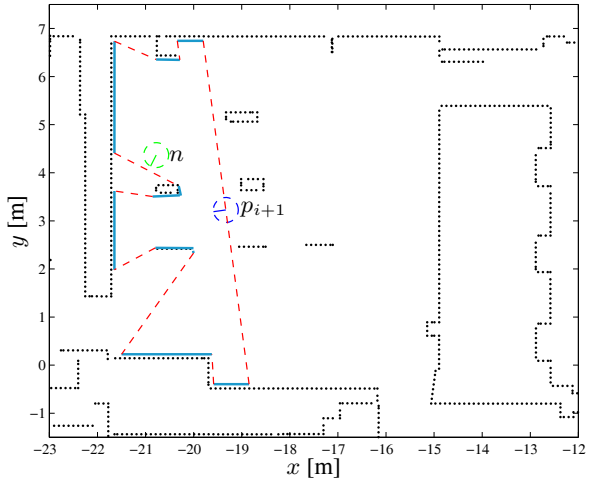

Fig. 10. Determination of jump edges (dashed lines) and environmental edges (solid lines) from the range data taken at the next measurement pose p_{i+1}.

current measurement pose is p_{i+1}, and chosen next measurement pose is n. Jump edges together with line segments create the extended measurement polygon $\widetilde{EP}$.

In the original exploration algorithm the exploration polygon $P_E(i)$ is created by adding the edges of the measurement polygon P to the exploration polygon $P_E(i-1)$ from the earlier measurement pose p_{i-1} by operations of expansion and fusion. Due to the localization error new environmental edges will be created and consequently more jump edges and measurement poses will be created than is really needed. The extended exploration polygon $EP_E(i)$ is created from the exploration polygon $P_E(i)$ by introducing jump edges.

Algorithm 2 Proposed exploration strategy: **explore**(S)

Input: Starting point S
Output: A sequence of move commands

1. $p_0 \leftarrow S$ *//Initialization*
2. $p_1 \leftarrow S$
3. $\widetilde{EP}_E(0) \leftarrow (\emptyset, \emptyset)$
4. $\mathcal{G}_E(0) \leftarrow (p_0, \emptyset)$
5. $i \leftarrow 1$
6. **repeat**
7. **move**(p_{i-1}, p_i)
8. $\mathcal{R} \leftarrow$ **new-measurement**(p_i)
9. $\{S_k\} \leftarrow$ **line-segments**$(\mathcal{R})$
10. **if** $|\{S_k\}| > 0$ **then**
11. $\{S_k^0\} \leftarrow$ **merge-line-segments**$(\{S_k\}, \{S^0\})$
12. $\widetilde{L_k} \leftarrow$ **sort-ending-points**$(\{S_k^0\})$
13. $\widetilde{EP} \leftarrow$ **add-jump-edges**(L_k)
14. **else**
15. $\widetilde{EP} \leftarrow$ **max-range-jump-edges**$(\mathcal{R})$
16. **end if**
17. $\widetilde{EP}_E(i) \leftarrow \widetilde{EP}_E(i-1) \cup \widetilde{EP}$
18. $\mathcal{G}_E(i) \leftarrow$ **exploration-graph**$(\mathcal{G}_E(i-1), \widetilde{EP}_E(i))$
19. $p_{i+1} \leftarrow$ **select-node**$(p_i, \mathcal{G}_E(i))$
20. $i \leftarrow i+1$
21. **until** $g_i = 0$

In the proposed exploration algorithm the extended exploration polygon $\widetilde{EP}_E(i)$ is created from the extended measurement polygon $\widetilde{EP}$ and the extended exploration polygon $\widetilde{EP}_E(i-1)$ from the earlier measurement pose p_{i-1}, where $\widetilde{EP}_E(0)$ is the empty polygon. By using general polygon clipping algorithm [22] it is possible to create union of polygons obtained from different measurement poses and thus creating approximation of explored environment from all measurement positions. The union of the extended measurement polygon $\widetilde{EP}$ and the extended exploration polygon $\widetilde{EP}_E(i-1)$ maintains only the most distant edges in the polygon from the measurement pose, and thus ensures that new jump edges are not created within previously explored region. Jump edges that are larger than the preset value Δr are considered as new measurement candidates. If the nonempty extended exploration polygon $\widetilde{EP}_E(i)$ contains no jump edges, then it is equal to the faithful polygonal description $P_E(F)$. An example of the extended exploration polygon $\widetilde{EP}_E(i+1)$ created as the

union of the extended measurement polygon $\widetilde{EP}$ (Fig. 10) and the extended exploration polygon $\widetilde{EP}_E(i) = \widetilde{EP}$ (for $i = 0$ in Fig. 9) is shown in Fig. 11.

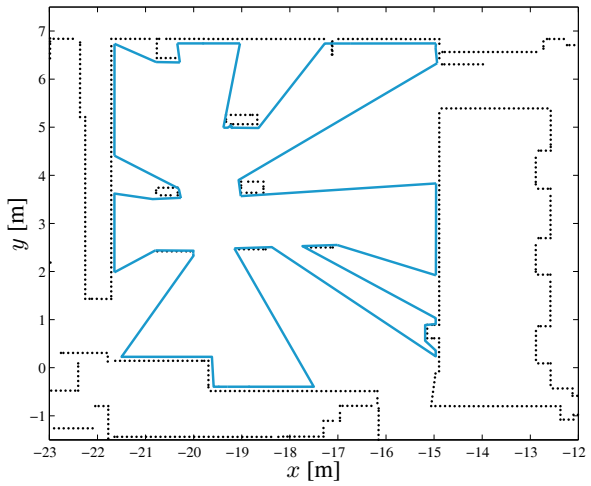

Fig. 11. The extended exploration polygon created as the union of polygons $\widetilde{EP}_{i+1}$ and $\widetilde{EP}_i$.

Pseudo code of the proposed exploration algorithm is given by Algorithm 2. The exploration algorithm terminates after the i-th measurement if all nodes have zero gain values.

5 TEST RESULTS

Ekman's exploration strategy and proposed algorithm are implemented in *Player/Stage* (a free software tool, www.playerstage.sourceforge.net) and experimentally verified on a Pioneer 3DX mobile robot. The laser range finder SICK LMS200 mounted on the robot was used for environment perception. It scans uniformly the environment in radial range of $\pm 180°$ with resolution of $0.5°$ and sends to the robot 361 values for distances to the detected obstacles every 200 ms. The CAD model map was used for algorithms verification. While exploring, the robot has to navigate between measurement poses in unknown environment. Motion planning algorithm described in [23] is used. In order to avoid obstacles while navigating between measurement poses, re-planing is done when necessary.

In simulation tests the localization with small uncertainty is chosen, and laser range finder has almost no noise, so both algorithms performed similarly and completely explored the environment in finite time. Robot trajectory while exploring the environment by the proposed algorithm is shown in Fig. 12. Obtained polygon shown in Fig. 13 almost completely matches given map due to small uncertainty of range sensor and localization.

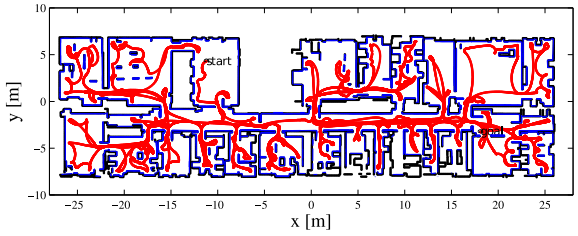

Fig. 12. Robot trajectory while exploring the environment by the proposed algorithm – simulation results.

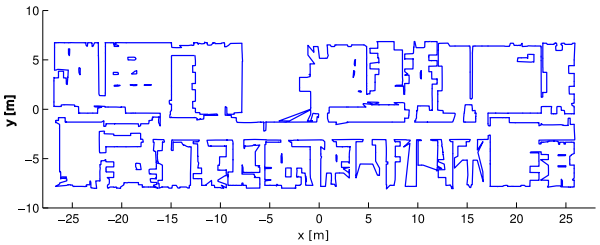

Fig. 13. Polygonal map obtained by the proposed algorithm – simulation results.

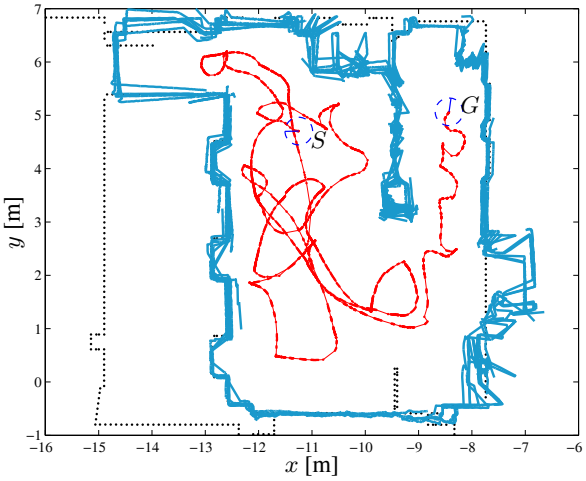

Fig. 14. Robot trajectory while exploring the environment and obtained polygonal map by Ekman's algorithm – experimental results.

Experiments were done in small part of the experimental environment. In order to show effectiveness of algorithm, the simulation map is shown by dots. However, the simulation map does not represent the real environment. For example, in real experiment all doors were closed, stairs were blocked by obstacles for safety reasons and some other non-mapped obstacles were in the room. Since solved localization problem is assumed the AMCL localization (Adaptive Monte Carlo Localization) was used, which uses the simulation map. In Fig. 14 ex-

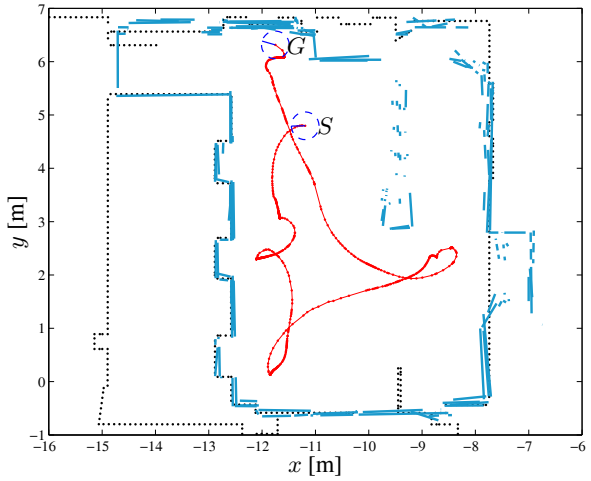

Fig. 15. Robot trajectory while exploring the environment and extracted line segments by the proposed algorithm – experimental results.

perimental results for Ekman's algorithm is shown, where 11547 line segments are obtained from 62 measurement positions. Figure 15 shows experimental results for proposed algorithm, where only 162 line segments are obtained from 13 measurement positions, which is significant improvement of the original algorithm. The beginning of exploring is noted by S, and the final position in which the environment is completely explored is noted by G. Obtained polygon is shown in Fig. 16.

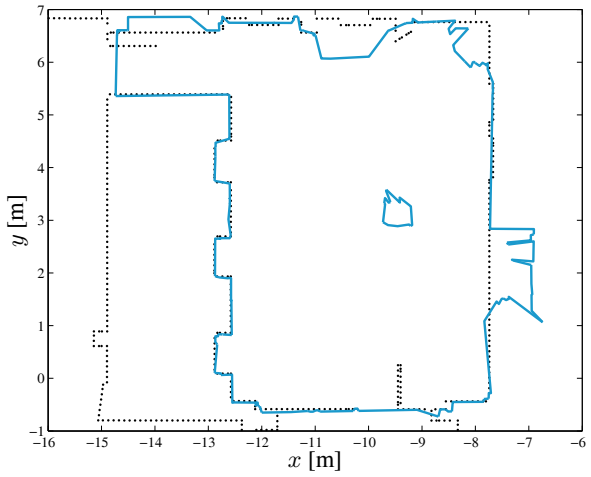

Fig. 16. Obtained polygon by the proposed algorithm – experimental results.

6 CONCLUSION

In this paper an autonomous exploration of unknown polygonal environment based on Ekman's algorithm is proposed. The proposed algorithm takes into consideration uncertainties as opposed to the original Ekman's algorithm, which assumes ideal localization and ideal laser range finder. Ekman's algorithm creates *visibility polygon* from range data, and proposed algorithm uses weighted line fitting algorithm proposed by Pfister *et al.* Pfister's algorithm provides statistically sound estimation of line segments from range data based on sensor and localization uncertainty. Edges of visibility polygon that do not correspond to environment edges are called *jump edges*. The exploration strategy explores on the fact that jump edges show directions for probably unexplored area. In proposed algorithm approximation of jump edges is computed by enclosing line segments into polygon. Ekman's algorithm assumes that visibility polygons from two adjacent measurement poses are perfectly matched. In proposed algorithm statistical method for merging line segments from two measurement positions is used. Ekman's algorithm eliminates jump edges in explored area by matching visibility polygons. In proposed algorithm a general polygon clipping algorithm is used to eliminate such jump edges. Ekman's algorithm has stopping criterion when there is no jump edges, proposed algorithm inherits that property. Obtained experimental results show that the line fitting with uncertainty taken into account ensures faster convergence of the algorithm and results in less line segments that represent the same environment, and thus lowers the map representation complexity.

REFERENCES

[1] A. Ekman, A. Torne, and D. Stromberg, "Exploration of polygonal environments using range data," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 27, no. 2, pp. 250–255, 1997.

[2] S. Pfister, S. Roumeliotis, and J. Burdick, "Weighted line fitting algorithms for mobile robot map building and efficient data representation," in *IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA'03*, vol. 1, 2003.

[3] R. Brooks, "Visual map making for a mobile robot," in *1985 IEEE International Conference on Robotics and Automation. Proceedings*, vol. 2, pp. 824–829, 1985.

[4] B. Oommen, S. Iyengar, N. Rao, and R. Kashyap, "Robot navigation in unknown terrains using learned visibility graphs. part i: The disjoint convex obstacle case," *Robotics and Automation, IEEE Journal of*, vol. 3, pp. 672–681, December 1987.

[5] V. Lumelsky, S. Mukhopadhyay, and K. Sun, "Dynamic path planning in sensor-based terrain acquisition," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 4, pp. 462–472, 1990.

[6] M. Mataric, "Integration of representation into goal-driven behavior-based robots," *IEEE Transactions on robotics and automation*, vol. 8, no. 3, pp. 304–312, 1992.

[7] J. O'Rourke, H. Booth, and R. Washington, "Connect-the-dots: A new heuristic," *Computer Vision, Graphics, and Image Processing*, vol. 39, no. 2, pp. 258 – 266, 1987.

[8] B. Tovar, R. Murrieta-Cid, and S. LaValle, "Distance-optimal navigation in an unknown environment without sensing distances," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 506–518, 2007.

[9] B. Yamauchi, A. Schultz, and W. Adams, "Integrating exploration and localization for mobile robots," *Adaptive Behavior*, vol. 7, no. 2, p. 217, 1999.

[10] T. Bandyopadhyay, Z. Liu, M. Ang, and W. Seah, "Visibility-based exploration in unknown environment containing structured obstacles," in *Proceedings of the 12th International Conference on Advanced Robotics (ICAR)*, pp. 484–491, 2005.

[11] R. Gartshore and P. Palmer, "Exploration of an unknown 2D environment using a view improvement strategy," *Towards Autonomous Robotic Systems*, pp. 57–64, 2005.

[12] G. Borges and M. Aldon, "A split-and-merge segmentation algorithm for line extraction in 2Drange images," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 1, 2000.

[13] A. Siadat, A. Kaske, S. Klausmann, M. Dufaut, and R. Husson, "An optimized segmentation method for a 2D laser-scanner applied to mobile robot navigation," in *Proceedings of the 3rd IFAC symposium on intelligent components and instruments for control applications*, 1997.

[14] D. Sack and W. Burgard, "A comparison of methods for line extraction from range data," in *Proceedings of the 5th IFAC Symposium on Intelligent Autonomous Vehicles (IAV)*, Citeseer, 2004.

[15] J. Illingworth and J. Kittler, "A survey of the hough transform," *Computer Vision, Graphics, and Image Processing*, vol. 44, no. 1, pp. 87 – 116, 1988.

[16] S. Roumeliotis and G. Bekey, "SEGMENTS: a layered, dual-Kalman filter algorithm for indoorfeature extraction," in *2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings*, vol. 1, 2000.

[17] J. Matas, C. Galambos, and J. Kittler, "Progressive probabilistic hough transform," in *Proc British Machine Vision Conference BMVC98*, pp. 256–265, 1998.

[18] J. Tavares and A. Padilha, "A new approach for merging edge line segments," in *RecPad'95-7th Portuguese Conference on Pattern Recognition*, 1995.

[19] S. Pfister, *Algorithms for mobile robot localization and mapping, incorporating detailed noise modeling and multi-scale feature extraction*. PhD thesis, California Institute of Technology, 2006.

[20] S. Pfister, "Weighted line fitting and merging," *California Institute of Technology, Tech. Rep*, 2002.

[21] S. Ileš, M. Seder, and I. Petrović, "Improvement of map building during the exploration of polygonal environments using the range data," in *Proceedings of the International Conference on Electrical Drives and Power Electronics (EDPE 2009)*, 2009.

[22] B. Vatti, "A generic solution to polygon clipping," *Communications of the ACM*, vol. 35, no. 7, pp. 56–63, 1992.

[23] M. Seder, K. Maček, and I. Petrović, "An integrated approach to real-time mobile robot control in partially known indoor environments," in *Proceedings of the 31st Annual Conference of the IEEE Industrial Electronics Society (IECON 2005)*, pp. 1785–1790, 2005.

Marija Đakulović received the B.Sc. degree in 2004 and Ph.D. degree in 2010, all in Electrical Engineering from the Faculty of Electrical Engineering and Computing (FER Zagreb), University of Zagreb, Croatia. Since 2004 she has been with the Department of Control and Computer Engineering at FER Zagreb, where she is currently working as a teaching assistant and a post-doc researcher. Her main research interests are: mobile robotics (especially path planning, obstacle avoidance and motion planning). During her undergraduate and graduate studies she was awarded with two faculty prizes, Vice-chancellor award and with scholarships from the Croatian ministry of science.

Šandor Ileš received dipl. ing. (in Croatia interpreted as M. Eng.) degree in electrical engineering from the University of Zagreb, Croatia in 2009. Currently he is a Ph.D. student working as an assistant at the Department of Electric Machines, Drives and Automation, Faculty of Electrical Engineering and Computing, University of Zagreb. His main research interests are control of mechatronic systems, application of linear matrix inequalities in control and polytopic system representation.

Ivan Petrović received B.Sc. degree in 1983, M.Sc. degree in 1989 and Ph.D. degree in 1998, all in Electrical Engineering from the Faculty of Electrical Engineering and Computing (FER Zagreb), University of Zagreb, Croatia. He had been employed as an R&D engineer at the Institute of Electrical Engineering of the KonÄŤar Corporation in Zagreb from 1985 to 1994. Since 1994 he has been with FER Zagreb, where he is currently a full professor and the head of the Department of Control and Computer Engineering. He teaches a number of undergraduate and graduate courses in the field of control systems and mobile robotics. His research interests include various advanced control strategies and their applications to control of complex systems and mobile robots navigation. Results of his research effort have been implemented in several industrial products. He is a member of IEEE, IFAC – TC on Robotics and FIRA – Executive committee. He is a collaborating member of the Croatian Academy of Engineering.

AUTHORS' ADDRESSES
Marija Đakulović
Ivan Petrović
Department of Control and Computer Engineering,
Šandor Ileš
Department of Electric Machines, Drives and Automation,
Faculty of Electrical Engineering and Computing,
University of Zagreb,
Unska 3, 10000 Zagreb, Croatia
email: marija.dakulovic@fer.hr, sandor.iles@fer.hr,
ivan.petrovic@fer.hr

Received: 2010-11-08
Accepted: 2011-02-02