

Decomposition of Line Segments into Corner and Statistical Grown Line Features in an EKF-SLAM Framework

Christian P. Connette**, Oliver Meister*, Martin Hägele**, Gert F. Trommer*

Abstract—Robots are emerging from industrial plants toward every people's daily life. Thus, navigation in and understanding of human related environments becomes a prerequisite for the systems of tomorrow. Most such environments can be efficiently described using line segments. However, incorporation of extent information is often difficult, as line segments are seldom observed completely and erroneous data-association may corrupt the information associated to a certain segment.

To reduce such problems this paper proposes a statistically driven description of line segments. The corresponding parameters are decomposed into line and corner features, which are separately tracked through an Extended Kalman Filter (EKF). Information about the extent of the lines is encoded statistically. Therefore, we use a method to recursively incorporate the information gained through a time-series of measurements. Thus, the covariance matrix belonging to a line segment grows as new regions of the corresponding line are discovered. Experimental results obtained by implementation on the mobile platform *ITrike* show the validity of our algorithm.

I. INTRODUCTION

A. Motivation

The ability of a machine to obtain an image of its surrounding world is one of the basic preliminaries for even simple applications of mobile systems. Accordingly autonomous navigation – especially the simultaneous localization and map building problem (SLAM) – has become a vividly examined and discussed topic in the field of mobile robotics.

The different kinds of robot generated embodiments of the environment can roughly be distinguished into semantic, topological and metrical maps (ordered by descending degree of abstraction) [1]. Metrical maps again are divided into so called grid maps and feature maps. Grid maps divide the surrounding world into separate cells. As the robot moves around the probability of being occupied or unoccupied is then calculated for every of these cells. Seminal works are [2], [3], [4]. Feature maps, which have attracted much attention especially in the last years, aim on condensing the spatial information about the environment on prominent objects or structures and their relative position and orientation ([5] to [8]).

As especially human related environments, e.g. office buildings or flats, often are dominated by linear and rectangular structures, many efforts have been taken and various methods have been developed to implement SLAM algorithms whose environmental representations base on

*Institute of Theory and Systems Optimization in Electrical Engineering (ITE), University of Karlsruhe, 76131 Karlsruhe - Germany

**Fraunhofer Institute for Manufacturing Engineering and Automation (IPA), 70504 Stuttgart - Germany

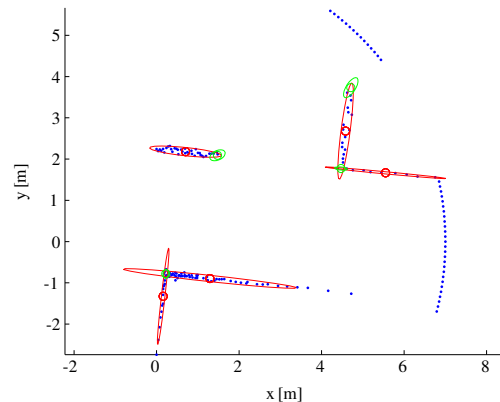


Fig. 1. Lines (red) and corners (green) extracted with their covariances from an simulated range image (blue)

this geometric primitives. These methods usually include a coregistering of lines and line-endpoints not only to update the line parameters but also to associate observed and known features. The solution of this so called data-association problem is of crucial importance for the robustness of a SLAM algorithm.

The method proposed in this paper aims on providing a line-based environment representation which is intrinsically local (Fig. 1) and thus allows for simplified data-association. With respect to this, one goal is to enable the application of standard data-association methods which are widely known and used in context of point-based environment representations [13]. Another goal is to derive a line representation that bears the potential of reduced sensitivity of the extent estimation against erroneous data-association by incorporating corner features which represent redundant information.

B. Related Work

Altermatt et.al. propose an approach which aims on tracking the relative position and orientation of corner features [14]. By adding an orientation parameter to the point-like features it becomes possible to implicitly describe walls. Yet, the generation of a map valid for tasks like path planning depends on the correct association of corner features which belong to the same wall.

In [9] to [12] line features are implemented by tracking their polar coordinates. However, polar coordinates intrinsically describe an infinite line. Accordingly, different line segments which are coaxial can not be discerned using only such a parameterization. Therefore in [10] and [11] the clusters of

the associated measurements are projected on the extracted line and the positions of the extremal points are determined together with their covariances to distinguish different line segments. Those extremal points are then separately tracked. Other methods, like in [9], represent the entire line segment just through these extremal points. [12] proposes the use of block features which store the orientation of a line segment together with the lower and upper bounds for its extension and width. However, standard data-association methods seem to be infeasible for these kinds of representations, as [9], [10] and [12] use additional similarity measures besides pure application of the mahalanobis distance. It seems as if above mentioned representations show a sensitivity against partial occlusion while associating repeated measurements of the same feature from different point of views.

In [15] Jeong and Lee present a three dimensional description of the environment based on local line and corner features. However, no relationships between the tracked lines and corners are stored. The extent of the lines is encoded, similarly to the above mentioned works, by additionally estimating segment endpoints.

In [16] the environment is modelled by 2d line segments extracted from 3d data. To represent these 2d lines the SPmodel is used, as it promises advantages over the polar coordinate representation as far as the transformation of the according covariances from one coordinate frame to another is concerned.

Brunskill and Roy use in [17] principal component analysis (PCA) to estimate orientation, center of gravity and extent of a line in an particle filter framework. Their work has basically the same underlying idea of using a probabilistic, local description of line features as the approach proposed in this paper. However, the implementation in an EKF framework, where position and orientation is estimated separately from extent information, leads to a different method for calculation and representation of the probabilistic information. Moreover by introducing additional corner features we gain the possibility to decide when it is advantageous to stop the line growing process.

C. Outline

The approach described in this paper is twofold. First, a parameterization of lines which is intrinsically local and thus point-like is chosen. Therefore, instead of implementing the often used polar coordinates, a line l_i is represented by the Cartesian coordinates of its center of gravity (x_{l_i}, y_{l_i}) and its orientation φ_{l_i} . The extension of a line is encoded by the covariance matrix resulting directly from the measurements associated to that line. A formula is provided to update this covariance matrix recursively as new measurements are associated to a line. This description principally allows for the use of standard algorithms to perform data-association.

Second, the range scan is searched for corner points on basis of the extracted lines. We present a combination of two methods which together ensure robust extraction of corners. These corners are represented through their Cartesian coordinates (x_{c_i}, y_{c_i}) which are estimated in an Extended

Kalman Filter (EKF) along with the parameters of the lines. That essentially decomposes line segments into local line and independent corner features. By investigating the relationship between corners and lines it becomes possible to identify the physical endpoints of walls or to interconnect separate lines based on the knowledge about common corners. This means that the line-extent information is encoded redundantly, promising an increased robustness against errors during the map-building process. The knowledge about line corner relations will also be used to stop the statistical growing of the lines.

II. FEATURE EXTRACTION

Before presenting the method used to detect corner features based on extracted lines, the line extraction algorithm itself will shortly be reviewed. A more detailed description of the line extraction can be found in [19].

A. Line Extraction

The first step is to apply an Iterative Endpoint Fit (IEPF) on the measurements, to segment the range data. The IEPF exploits that a range scanner delivers measurements in an sequential order. Based on the Cartesian coordinates (x, y) of the first point $p_{l_i,b}$ and the last point $p_{l_i,e}$ of a potential line segment l_i a first guess for the polar parameters

$$\varphi_{l_i} = \arctan\left(\frac{y_{l_i,e} - y_{l_i,b}}{x_{l_i,e} - x_{l_i,b}}\right) \quad (1)$$

$$r_{l_i} = x_{l_i,e} \cdot \cos(\varphi_{l_i}) + y_{l_i,e} \cdot \sin(\varphi_{l_i}) \quad (2)$$

of the according line is calculated. Then, for every point $p_{l_i,j}$ between $p_{l_i,b}$ and $p_{l_i,e}$ the distance

$$d_{i,j} = r_{l_i} - (x_{l_i,j} \cdot \cos(\varphi_{l_i}) + y_{l_i,j} \cdot \sin(\varphi_{l_i})) \quad (3)$$

to the line l_i is calculated. If the distance $d_{i,j}$ to the line l_i is below a certain threshold d_{max} for all points, the segment is accepted. If not, the current segment is split at the point with the biggest distance to l_i and the procedure is repeated for the resulting new segments. As a result of noise the IEPF may sometimes fail to detect gaps. To avoid this, a split and merge step was appended to the IEPF, testing the segments and derived parameters on their validity. Based on the calculated parameters (r_{l_i}, φ_{l_i}) of the expected line the gap between neighbouring points is predicted. If the actual gap is significantly bigger than the prediction, the segment is split at this point. To avoid oversegmentation the new segments are compared with their neighbouring ones and merged were appropriate.

After the different line segments are identified the according center of gravity (x_{l_i}, y_{l_i}) and covariance matrix C_{l_i} are calculated based on the N_{l_i} measurements $(x_{l_i,k}, y_{l_i,k})$ of

this segment:

$$\begin{pmatrix} x_{l_i} \\ y_{l_i} \end{pmatrix} = \frac{1}{N_{l_i}} \sum_{k=1}^{N_{l_i}} \begin{pmatrix} x_{l_i,k} \\ y_{l_i,k} \end{pmatrix} \quad (4)$$

$$C_{l_i} = \begin{pmatrix} C_{xx,l_i} & C_{xy,l_i} \\ C_{yx,l_i} & C_{yy,l_i} \end{pmatrix} \quad (5)$$

$$C_{xx,l_i} = \frac{1}{N_{l_i} - 1} \sum_{k=1}^{N_{l_i}} (x_{l_i,k} - x_{l_i})^2 \quad (6)$$

$$C_{yy,l_i} = \frac{1}{N_{l_i} - 1} \sum_{k=1}^{N_{l_i}} (y_{l_i,k} - y_{l_i})^2 \quad (7)$$

$$C_{xy,l_i} = \frac{1}{N_{l_i} - 1} \sum_{k=1}^{N_{l_i}} (x_{l_i,k} - x_{l_i}) \cdot (y_{l_i,k} - y_{l_i}) \quad (8)$$

$$C_{yx,l_i} = C_{xy,l_i}^T \quad (9)$$

The final orientation φ_{l_i} and distance r_{l_i} of the line and their covariance C_{φ,l_i} is calculated using Haralick's Method [20].

Unlike [19] the parameter vector characterising one single line is composed by the calculated center of gravity and orientation

$$l_i = \begin{pmatrix} x_{l_i} \\ y_{l_i} \\ \varphi_{l_i} \end{pmatrix} \quad (10)$$

$$P_{l_i} = \begin{pmatrix} C_{xx,l_i} & C_{xy,l_i} & 0 \\ C_{yx,l_i} & C_{yy,l_i} & 0 \\ 0 & 0 & C_{\varphi,l_i} \end{pmatrix}, \quad (11)$$

to achieve a point-like parameterization for representation of the measured lines.

B. Corner Extraction

Now, based on the extracted line segments, the corners are extracted. We discern two kinds of corners: The intersection points of neighbouring lines, which resemble the corners of a room and the endpoints of lines, e.g. open doorways. The second type can easily be detected if in a range scan measurements belonging to a line segment are followed by measurements at the maximum range of the sensor. In this case a prediction for the expected distance of the next measurement is calculated based on the parameters of the extracted line. If the prediction is significantly smaller as the de facto measured distance, a corner of type two has been found. If not, the line just vanishes at the edges of the robots sensor-range (see Fig. 1 for both cases).

For type 1 corners, where one line is directly followed by another, it has to be tested if the line intersects with the other or just occludes it. In this case we distinguish occlusion and intersection by two measures. First, we calculate the Mahalanobis Distance

$$D_{MH} = (p_e - p_b)^T (C_e + C_b)^{-1} (p_e - p_b) \quad (12)$$

between the two neighbouring points of the segments. If the distance is above a certain threshold, the system decides on occlusion. However, this threshold is arbitrary chosen and may fail e.g. when a corner is far away from the sensor and

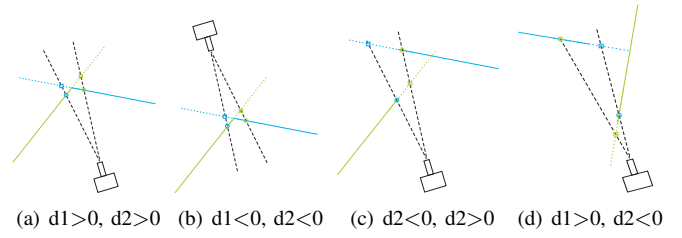


Fig. 2. Base types of possible relative line to line relations. Lines in 2(a) and 2(b) intersect while lines in 2(c) and 2(d) occlude each other.

thus the density of the measured points is very low and their distance from each other large compared to their uncertainty. Thus, we also calculate a measure for the relative position of the neighbouring lines. Therefore, we calculate for both line segments the prediction

$$d_{s,i} = \sqrt{x_{s,i}^2 + y_{s,i}^2} \quad (13)$$

for the next distance measurement if the considered line would be continued beyond the expected intersection point. Here $x_{s,i}$ and $y_{s,i}$ denote the Cartesian coordinates of the expected intersection point of the next laser-ray in the range scan and the considered line segment l_i (Fig. 2). If for both lines the differences d_1 and d_2 between expected and actual measurement (r_l)

$$d_1 = d_{s,i} - r_{l_{i+1},b} \quad (14)$$

$$d_2 = d_{s,i+1} - r_{l_i,e} \quad (15)$$

have the same sign, the lines do intersect. This is a sufficient but not a necessary condition. E.g. if the corner is nearby and thus the density of the measured points is very high and their distance from each other small compared to their uncertainty, the sign of d_1 or d_2 may be corrupted by noise. That means, that the second method is very robust when the first one, calculation of Mahalanobis Distance, is sensitive to errors. Vice versa method one is very robust in situations, where method two is sensitive to errors.

After discerning the segment endpoints in corners of type one or two and vanishing points the Cartesian coordinates (x_c, y_c) of the corners and the corresponding covariance matrices C_c are calculated. This is done by calculating the mean of the center of gravity (x_l, y_l) of the lines weighted by the according covariance matrices C_l :

$$P_{c_j} = \left(\frac{1}{2} C_{l_i}^{-1} + \frac{1}{2} C_{l_{i+1}}^{-1} \right)^{-1}$$

$$\begin{pmatrix} x_{c_j} \\ y_{c_j} \end{pmatrix} = P_{c_j} \left(\frac{1}{2} C_{l_i}^{-1} \begin{pmatrix} x_{l_i} \\ y_{l_i} \end{pmatrix} + \frac{1}{2} C_{l_{i+1}}^{-1} \begin{pmatrix} x_{l_{i+1}} \\ y_{l_{i+1}} \end{pmatrix} \right)$$

This approach resembles the application of a Kalman Filter with inflation of the resulting covariance by factor 2 or the application of the covariance intersection principle

$$C = (\omega A^{-1} + (1 - \omega) B^{-1})^{-1} \quad (16)$$

$$c = C (\omega A^{-1} a + (1 - \omega) B^{-1} b), \quad (17)$$

when ω is set to the constant value 0.5.

C. Composition of measurement vector

The feature vector z and the corresponding covariance matrix R characterising the complete range scan are then composed by the parameters and covariances of all observed lines and corners

$$z = \begin{pmatrix} \vdots \\ l_i \\ \vdots \\ c_j \\ \vdots \end{pmatrix}, \quad R = \begin{pmatrix} \ddots & 0 & \cdots & \cdots & 0 \\ 0 & P_{l_i} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & P_{c_j} & 0 \\ 0 & \cdots & \cdots & 0 & \ddots \end{pmatrix},$$

where we ignore the crosscovariances arising in the sensor coordinate frame as corners are calculated based on the lines.

Furthermore, for every line l_i the covariance matrix P_{l_i} characterising the distribution of the measured points p_{l_i} associated to that line is stored independently. This distribution is a measure for the extent of an observed line segment. Also, for every extracted corner an identifier for the lines from which this corner is calculated is stored.

III. ESTIMATION OF LINE AND CORNER POSE WITH THE EXTENDED KALMAN FILTER

The application of the Extended Kalman Filter to the SLAM problem has been introduced by Dissanayake et al. in [7]. For the sake of completeness a short review is given here, before the measurement equations are adapted to the proposed feature representation.

A. Concept of the Extended Kalman Filter in SLAM

The Kalman Filter (KF), developed by R.E. Kalman [21], is a least squares estimator, which exploits knowledge about the dynamics of a linear technical system. It is composed by an prediction step

$$s_k^- = A s_{k-1}^+ + B u_k \quad (18)$$

$$P_k^- = A P_{k-1}^+ A^T + B U_k B^T + Q_k, \quad (19)$$

and an estimation step

$$K_k = P_k^- H^T (H P_k^- H^T + R_k)^{-1} \quad (20)$$

$$s_k^+ = s_k^- + K_k (z_k - H x_k^-) \quad (21)$$

$$P_k^+ = (I - K_k H) P_k^-. \quad (22)$$

A way to approximate an KF in the case of nonlinear system dynamics $f(s_k, u_k)$ and measurement model $h(s_k)$ is the Extended Kalman Filter (EKF), which linearizes dynamics and measurement equations around the current working point to propagate the probabilities.

The Idea of applying an Extended Kalman Filter (EKF) on the SLAM-Problem [7] is to estimate the relative position and orientation of all features together with the position and orientation of the robot. Therefore, the system state vector $s = (s_{rob}, s_{feat})^T$ is composed by the parameters describing the robot $s_{rob} = (x_{rob}, y_{rob}, \varphi_{rob})^T$ and the parameters describing all features $s_{feat} = (f_1, \dots, f_l)^T$. The parameters

of the features are usually modeled as constants. Thus, the matrices describing the system dynamics become

$$A = \begin{pmatrix} A_{rob,k} & 0 \\ 0 & I \end{pmatrix}, \quad B = \begin{pmatrix} B_{rob,k} \\ 0 \end{pmatrix}, \quad (23)$$

where

$$A_{rob,k} = \nabla_{s_{rob,k}} f_{rob}(s_k, u_k) \quad \text{and} \quad (24)$$

$$B_{rob,k} = \nabla_{u_k} f_{rob}(s_k, u_k) \quad (25)$$

are the linearizations of the system dynamics $f(s_k, u_k)$ with respect to the robot states. The measurement matrix H_k is obtained by concatenation

$$H_k = \begin{pmatrix} H_{f_1,k} \\ \vdots \\ H_{f_l,k} \end{pmatrix} \quad (26)$$

of the linearized measurement equations

$$H_{f_i,k} = (H_{rob,k} \quad 0 \quad \cdots \quad 0 \quad H_{f_i,k} \quad 0 \quad \cdots) \quad (27)$$

of the single features, where

$$H_{rob,k} = \nabla_{s_{rob,k}} h_{rob}(s_k) \quad \text{and} \quad (28)$$

$$H_{f_i,k} = \nabla_{s_{f_i,k}} f_{rob}(s_k) \quad (29)$$

are the linearizations of the measurement equation $h_{f_i}(s_k)$ of the single observed feature.

B. Integration of new Features

Special for the SLAM-Problem is the need to incorporate new, not yet observed features, in the state vector. The parameters of the new feature s_{new} in the global reference frame are calculated as

$$s_{new} = g(s_{rob,k}, z), \quad (30)$$

where g is a function of the parameters of the measured feature z in the measurement space and the current robot state $s_{rob,k}$. To correctly expand the covariance matrix P_k to P_{new} this function is linearized and used to calculate the crosscovariances between the existing states and the new state. The resulting state vector and covariance matrix then become

$$s = \begin{pmatrix} s_{old} \\ s_{new} \end{pmatrix} \quad (31)$$

$$P = \begin{pmatrix} P_{old} & P_{old} G_x^T \\ G_x P_{alt} & G_x P_{alt} G_x^T + G_z R_{zz} G_z^T \end{pmatrix} \quad (32)$$

with

$$G_x = (\nabla_{s_{rob,k}} g \quad 0 \quad \cdots \quad 0)$$

$$G_z = \nabla_z g.$$

C. Explicit Formulation for the considered System

In the here considered case the system state vector

$$s = \begin{pmatrix} s_{rob} \\ s_{lin} \\ s_{cor} \end{pmatrix} \quad (33)$$

is composed by the parameters describing position and orientation of the robot, the parameters describing position and orientation of all known lines

$$s_{lin} = \begin{pmatrix} l_1 \\ \vdots \\ l_n \end{pmatrix} \text{ with } l_i = \begin{pmatrix} x_{l_i} \\ y_{l_i} \\ \varphi_{l_i} \end{pmatrix} \quad (34)$$

and the parameters describing the position of all known corners

$$s_{cor} = \begin{pmatrix} c_1 \\ \vdots \\ c_m \end{pmatrix} \text{ with } c_i = \begin{pmatrix} x_{c_i} \\ y_{c_i} \end{pmatrix}. \quad (35)$$

The control inputs are the velocity v_k and the rotation ω_k of the robot $u_k = (v_k, \omega_k)^T$. The dynamic of the robot states can be modeled as

$$f_{rob}(s_k, u_k) = \begin{pmatrix} x_{rob,k} + \cos(\varphi_{rob,k})v_k\Delta t \\ y_{rob,k} + \sin(\varphi_{rob,k})v_k\Delta t \\ \varphi_{rob,k} + \omega_k\Delta t \end{pmatrix}, \quad (36)$$

where Δt is the length of time between two discrete steps. The linearized dynamic becomes then:

$$A_{rob,k} = \begin{pmatrix} 1 & 0 & -\sin(\varphi_{rob,k})v_k\Delta t \\ 0 & 1 & \cos(\varphi_{rob,k})v_k\Delta t \\ 0 & 0 & 1 \end{pmatrix} \quad (37)$$

$$B_{rob,k} = \begin{pmatrix} \cos(\varphi_{rob,k})\Delta t & 0 \\ \sin(\varphi_{rob,k})\Delta t & 0 \\ 0 & \Delta t \end{pmatrix} \quad (38)$$

To emphasis the point-like representation of the lines the measurement step is performed in Cartesian space. The corresponding equation for the i -th known line segment l_i is

$$h_i(s_k) = \begin{pmatrix} \Delta x_{i,k} \cos(\varphi_{rob,k}) + \Delta y_{i,k} \sin(\varphi_{rob,k}) \\ -\Delta x_{i,k} \sin(\varphi_{rob,k}) + \Delta y_{i,k} \cos(\varphi_{rob,k}) \\ \Delta \varphi_{i,k} - k\pi \end{pmatrix} \quad (39)$$

with

$$\begin{aligned} \Delta x_{i,k} &= x_{l_i,k} - x_{rob,k}, \\ \Delta y_{i,k} &= y_{l_i,k} - y_{rob,k}, \\ \Delta \varphi_{i,k} &= \varphi_{l_i,k} - \varphi_{rob,k}. \end{aligned}$$

The factor k in (39) equals 1 if the line lies between the robot and the origin of the global reference frame, 0 otherwise. The linearization $H_{rob,k}$ with respect to the robot states becomes

$$H_{rob,k} = \begin{pmatrix} -\cos(\varphi_{rob,k}) & -\sin(\varphi_{rob,k}) & h_{i,2}(s_k) \\ \sin(\varphi_{rob,k}) & -\cos(\varphi_{rob,k}) & -h_{i,1}(s_k) \\ 0 & 0 & -1 \end{pmatrix} \quad (40)$$

and the linearization $H_{l_i,k}$ with respect to the states of line l_i becomes

$$H_{l_i,k} = \begin{pmatrix} \cos(\varphi_{rob,k}) & \sin(\varphi_{rob,k}) & 0 \\ -\sin(\varphi_{rob,k}) & \cos(\varphi_{rob,k}) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (41)$$

The function $g(s_k, z)$ which transforms the measured features into the global reference frame becomes

$$g = \begin{pmatrix} x_{rob,k} + x_z \cos(\varphi_{rob,k}) + y_z \sin(\varphi_{rob,k}) \\ y_{rob,k} + x_z \sin(\varphi_{rob,k}) + y_z \cos(\varphi_{rob,k}) \\ \varphi_z + \varphi_{rob,k} + k\pi \end{pmatrix}. \quad (42)$$

The equations for the corner features are readily derived from those of the line features by omitting the last line, referring to the lines orientation.

IV. ESTIMATION OF LINE EXTENT AND LINE-CORNER RELATION

A. Recursive Estimation of Extent Information

The estimate for the position of the features and the corresponding uncertainty are calculated in the EKF. To represent the extent of a line, an additional covariance matrix approximating the dispersion of all measurements associated to that line, is calculated recursively as follows (Fig. 3).

After every incoming measurement first the estimation step of the EKF is performed. The covariances $C_{s_{l_i,k}}$ encoding the length of the line segments are adapted by rotating them

$$C'_{s_{l_i,k}} = \Phi_{l_i,k} C_{s_{l_i,k}} \Phi_{l_i,k}^T \quad (43)$$

$$\Phi_{l_i,k} = \begin{pmatrix} \cos(\Delta \varphi_{l_i,k}) & -\sin(\Delta \varphi_{l_i,k}) \\ \sin(\Delta \varphi_{l_i,k}) & \cos(\Delta \varphi_{l_i,k}) \end{pmatrix} \quad (44)$$

by the difference $\Delta \varphi_{l_i,k}$ in the lines orientations

$$\Delta \varphi_{l_i,k} = \varphi_{l_i,k}^+ - \varphi_{l_i,k}^- \quad (45)$$

before $\varphi_{l_i,k}^-$ and after $\varphi_{l_i,k}^+$ the estimation step.

Analogous, the measured points, represented by covariance $C_{z_{l_j}}$ and center of gravity $(x_{z_{l_j}}, y_{z_{l_j}})$ of the associated observed line z_{l_j} have to be projected on the already known line s_{l_i} . Therefore, the covariance matrix is rotated

$$C'_{z_{l_j}} = \Phi_{l_i l_j,k} C_{z_{l_j}} \Phi_{l_i l_j,k}^T \quad (46)$$

$$\Phi_{l_i l_j,k} = \begin{pmatrix} \cos(\Delta \varphi_{l_i l_j,k}) & -\sin(\Delta \varphi_{l_i l_j,k}) \\ \sin(\Delta \varphi_{l_i l_j,k}) & \cos(\Delta \varphi_{l_i l_j,k}) \end{pmatrix} \quad (47)$$

by the difference in the orientation

$$\Delta \varphi_{l_i l_j,k} = \varphi_{l_i,k}^+ - \varphi_{z_{l_j}} \quad (48)$$

of the observed and the known line. The transformation is then completed by projecting the center of gravity $(x_{z_{l_j}}, y_{z_{l_j}})$ of the observed line onto the known line. The transformed center of gravity $(x'_{z_{l_j}}, y'_{z_{l_j}})$ hence becomes

$$\begin{pmatrix} x'_{z_{l_j}} \\ y'_{z_{l_j}} \end{pmatrix} = \begin{pmatrix} x_{z_{l_j}} + \Delta d \sin(\varphi_{l_i,k}^+) \\ y_{z_{l_j}} + \Delta d \cos(\varphi_{l_i,k}^+) \end{pmatrix} \quad (49)$$

where Δd is the distance

$$\Delta d = d_{s_{l_i},k} - d_{z_{l_j}} \quad (50)$$

between the known and the observed line. Now, the extent information encoded in the two covariances $C'_{s_{l_i},k}$ and $C'_{z_{l_j}}$ is fused in the covariance $C_{s_{l_i},k+1}$. Therefore, first the temporary center of gravity $(x_{s_{l_i}}, y_{s_{l_i}})_t$ for the whole set of measured points is calculated by the weighted average of the former centers of gravity

$$\begin{pmatrix} x_{s_{l_i}} \\ y_{s_{l_i}} \end{pmatrix}_t = \begin{pmatrix} \frac{n_{s_{l_i},k}}{n_{s_{l_i},k+1}}x_{s_{l_i},k}^+ + \frac{n_{z_{l_j}}}{n_{s_{l_i},k+1}}x'_{z_{l_j}} \\ \frac{n_{s_{l_i},k}}{n_{s_{l_i},k+1}}y_{s_{l_i},k}^+ + \frac{n_{z_{l_j}}}{n_{s_{l_i},k+1}}y'_{z_{l_j}} \end{pmatrix} \quad (51)$$

where $n_{s_{l_i},k}$ is the number of measurements associated to the known line, $n_{z_{l_j}}$ the number of those associated to the observed line and $n_{s_{l_i},k+1}$ the sum of both

$$n_{s_{l_i},k+1} = n_{s_{l_i},k} + n_{z_{l_j}}, \quad (52)$$

the updated number of measurements associated to the known line. The new covariance matrix can now be calculated. The terms characterising the autocorrelation with respect to the x -coordinate become

$$\begin{aligned} C_{xx} &= \frac{n_{s_{l_i},k-1}}{n_{s_{l_i},k+1}-1} \left(C'_{xx,s_{l_i}} + \frac{n_{s_{l_i},k}}{n_{s_{l_i},k-1}}(x_{s_{l_i},t} - x_{s_{l_i},k}^+)^2 \right) \\ &+ \frac{n_{z_{l_j}-1}}{n_{s_{l_i},k+1}-1} \left(C'_{xx,z_{l_j}} + \frac{n_{z_{l_j}}}{n_{z_{l_j}-1}}(x_{s_{l_i},t} - x'_{z_{l_j}})^2 \right) \end{aligned} \quad (53)$$

and analogous for the autocorrelation $C_{yy,k+1}$ with respect to the y -component. The covariances characterising the cross-correlations become

$$\begin{aligned} C_{xy} &= \frac{n_{s_{l_i},k-1}}{n_{s_{l_i},k+1}-1} \left(C'_{xy,s_{l_i}} \right. \\ &+ \left. \frac{n_{s_{l_i},k}}{n_{s_{l_i},k-1}}(x_{s_{l_i},t} - x_{s_{l_i},k}^+)(y_{s_{l_i},t} - y_{s_{l_i},k}^+) \right) \\ &+ \frac{n_{z_{l_j}-1}}{n_{s_{l_i},k+1}-1} \left(C'_{xy,z_{l_j}} \right. \\ &+ \left. \frac{n_{z_{l_j}}}{n_{z_{l_j}-1}}(x_{s_{l_i},t} - x'_{z_{l_j}})(y_{s_{l_i},t} - y'_{z_{l_j}}) \right) \end{aligned} \quad (54)$$

while, as a result of the symmetry of covariance matrices, $C_{yx,k+1}$ is identical. The updated covariance, representing the lines extent then becomes

$$C_{s_{l_i},k+1} = \begin{pmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{pmatrix}. \quad (55)$$

The coordinates of the temporary center of gravity $(x_{s_{l_i}}, y_{s_{l_i}})_t$ are discarded, as the information of place of the line was already incorporated into the system state during the EKF's estimation step.

B. Line-Corner Relations

Lines and corners are decomposed into separate features. For reference, their relationship can be investigated, as corresponding measurements are available. Thus, during feature-extraction the relations between corners and lines are extracted and stored.

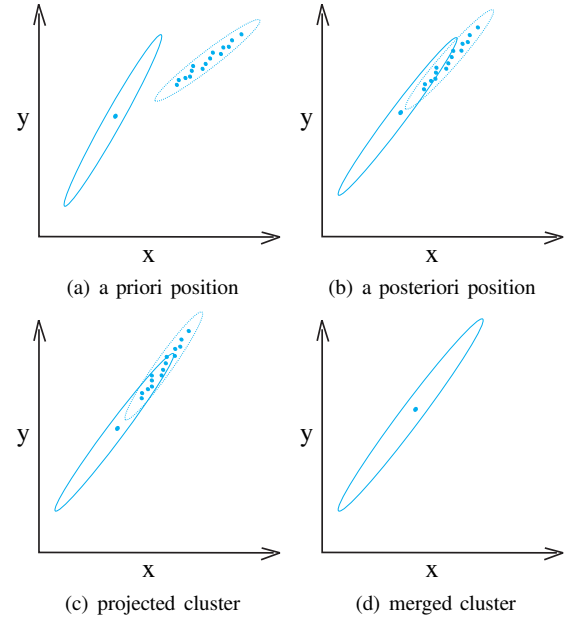


Fig. 3. Exemplified incorporation of a new measurement into the covariance of a known feature. 3(a) shows relative position of existing and new feature directly after measurement. 3(b) shows relative position after robot pose has been corrected in the estimation step. 3(c) shows projection of new feature onto the existing one. 3(d) shows new covariance after merging

If a known feature is observed, it is checked whether a relationship to another observed feature has been detected during the extraction process. If so, an identifier indicating the relationships between the known features is stored in a list associated to the known feature. To improve robustness this procedure is restricted to already known features. If a new feature is inserted into the EKF it is initialized without any relationship information.

The obtained relationship information is used to stop the statistical growing of the line segments. Apparently, as soon as a corner is associated to a line, this line has grown large enough in the direction of this corner. If the center of gravity of an observed line lies between the center of gravity of a known line and a corner known to belong to that known line, the new set of measurements is not incorporated into the covariance matrix encoding the extant information.

V. DATA-ASSOCIATION

Before incoming data can be processed in the EKF the observed features have to be associated to the already known ones. A common way to do this is to calculate the Mahalanobis Distance (D_{MH})

$$D_{MH}(i, j) = (s_i - z_j)^T (C_{s_i} + C_{z_j})^{-1} (s_i - z_j) \quad (56)$$

between the observed (z) and the known (s) features and associate always those features z_j and s_i , which have the smallest Mahalanobis Distance from each other. More robust are methods that minimize D_{MH} over a whole set of possible associations. In [13] a comprehensive description of these methods can be found.

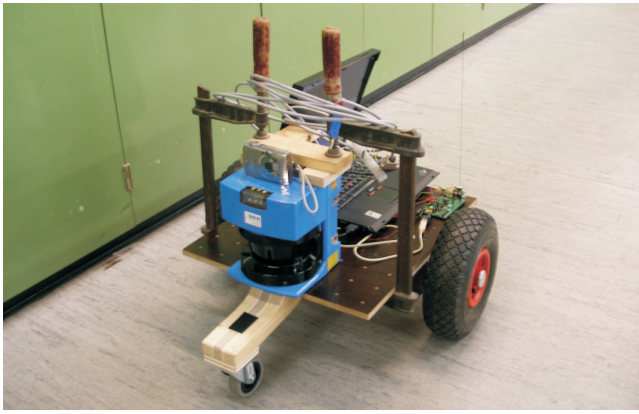


Fig. 4. ITE's experiment platform *ITrike* equipped with an SICK LMS200

The Mahalanobis Distance is a measure for the Euclidean distance between two points in a space of arbitrary dimension, weighted by the inverse of the uncertainty of these points. If the Mahalanobis Distance between expanded objects like lines shall be calculated additional measures have to be taken. With respect to the here chosen point-like representation of line features the Mahalanobis Distance between lines can readily be calculated by simply adding the covariance matrix $C_{s_{l_i,k}}$ encoding extension and the covariance matrix $P_{s_{l_i,k}}$ encoding the uncertainty of the i -th line's local parameters $(x_{l_i,k}, y_{l_i,k}, \varphi_{l_i,k})$

$$C_{s_i} = P_{s_{l_i,k}} + C_{s_{l_i,k}} \quad (57)$$

VI. EXPERIMENTAL RESULTS

The experiments were performed on the mobile platform *ITrike* (Fig. 4), currently under development at the ITE. During the tests the system was operated by a remote controle. To acquire range measurements, a Sick LMS200, with an aperture angle of 180° , was used with the resolution of 0.5° . The odometry had an resolution of 1.25° related to the wheel rotation.

The algorithm was tested by partially mapping the third floor of the ITE's office building (Fig. 5). Here, the system had to close a loop of approximately 14m length and 6m width while traveling through a modest cluttered environment (room in the lower left part).

Fig. 6(a) shows the results obtained for the robot trajectory if simply the data delivered by the odometry was integrated. In comparison Fig. 6(b) shows the estimated robot-trajectory obtained if the range measurements were incorporated in form of corner and statistical grown line features and used to correct the pose estimates.

Fig. 7 shows the obtained map if the range-measurements are naively integrated based on the trajectory calculated from the odometry data (Fig. 7(a)) and the map obtained when line and corner features are extracted (Fig. 7(b)) to correct the robots pose.

A sound quantitative evaluation of the results is rather difficult, as only relative position estimates are obtained by the algorithm. However, in Table I the dimensions of

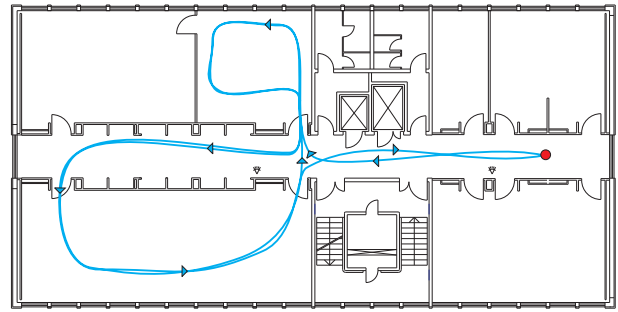


Fig. 5. Map of 3rd floor at ITE and modeled driven trajectory

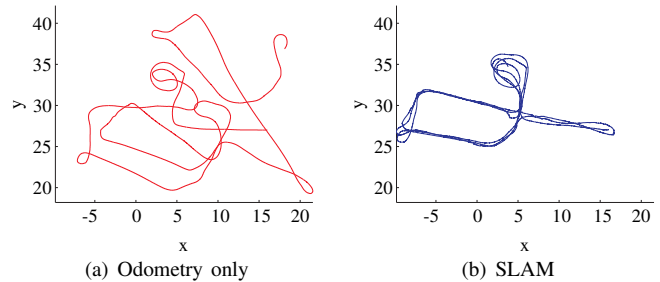


Fig. 6. Estimated robot trajectory with and without incorporation of range measurements

the rooms, as denoted in the plans, are compared with the estimated distance of features situated on opposite walls. As expected, dimensions which can be acquired within a single range scan, e.g. width of corridor and rooms, are mapped much more accurately, than those which cannot be seen at once, e.g. the length of the corridor. The algorithm seems to tend to underestimate dimensions, as both the length of the corridor and the length of the bigger room are estimated too small. The difference between experimental result and ground truth is here at the edge of the 2σ -region of the according covariance. This tendency is probably a result of emphasising the point-like character of the proposed line representation by performing the measurement step in Cartesian space.

Qualitatively, the algorithm renders the environment reasonably well. Fig. 8 shows a projection of the obtained map on the plan of the 3rd floor. Corners are only placed where doorways or obstacles are. The conglomeration of corners at the outside walls of the rooms are due to the edges of radiators situated there. Oversegmentation of the walls occurred only in the case of the very large corridor wall and for the larger of the two rooms, which was modest cluttered.

TABLE I
REAL DIMENSIONS VS. ESTIMATES

	Dimensions of 3rd floor at ITE [m x m]					
	ground truth			estimated		
3rd floor	16.16	x	32.14			
Corridor	2.39	x	32.32	2.33	x	31.91
Room 1	6.58	x	6.27	6.58	x	6.31
Room 2	6.58	x	15.83	6.57	x	15.63

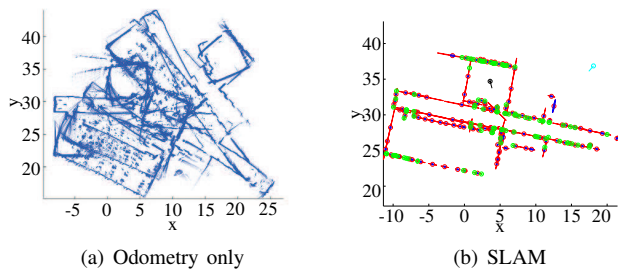


Fig. 7. Obtained map; Fig. 7(a) shows all assembled range data (blue); Fig. 7(b) shows the center of gravity of the lines (red) and corners (green) together with their covariances tracked in the EKF (blue/green) and the covariances encoding extent information (red)

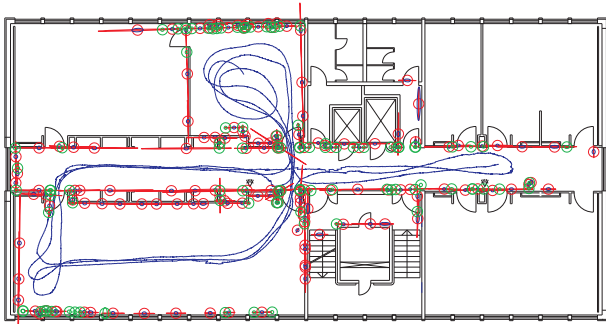


Fig. 8. Map of decomposed line and corner features projected into plan of 3rd floor (angular offset corrected manually). Covariances encoding the line length are represented as red ellipses (2σ -Region).

VII. CONCLUSION AND FUTURE WORKS

In this paper a method for the description of linear features based on decomposition into local lines and corners was presented. A method to statistically represent extent information has been proposed. Thus, the necessity to track temporary line endpoints was remedied. A way to recursively incorporate new measurements in the extent estimate was shown. Knowledge about line corner relations was used to stop the statistical growing process of the lines.

Based on the proposed parameterization of the environment the simultaneous localization and mapping problem was tackled. Therefore, an Extended Kalman Filter was used to estimate the parameters of the lines and corners characterising position and orientation. The measurement step was performed in Cartesian space to emphasis the point-like characteristics of the line features. The local parameterization and statistical encoding of line extent was exploited to simplify the calculation of the Mahalanobis Distance for data-association of observed lines significantly.

With the implemented algorithm the test platform *ITrike* was able to map the surrounding, localize in modest cluttered environments and close small loops. The corner features made it possible to resolve wall end points, e.g. edges of rooms and door openings.

Future work will aim on generating environmental representations based on the obtained feature map, which are more apt for tasks like path planning. E.g. the statistical description promises to be readily transformed into potential fields.

Also, as erroneous data-association is the main threat to the robustness of the system, more sophisticated algorithms for data-association will be examined.

REFERENCES

- [1] R. Chatila and J. Laumond, "Position referencing and consistent world modeling for mobile robots", *IEEE International Conference on Robotics and Automation*, 1985, pp 138-145.
- [2] H.P. Morovac and A. Elfes, "High resolution maps from wide angle sonar", *IEEE International Conference on Robotics and Automation*, 1985, pp 116-121.
- [3] S. Thrun, "Learning Occupancy Grid Maps with forward sensor models", *Autonomous Robots*, Springer, Vol. 15, No. 2, September 2003, pp 111-127.
- [4] W. Burgard, D. Fox, H. Jans, C. Matenar, S. Thrun, "Sonar-Based mapping with mobile robots using EM", *International Conference on Machine Learning*, 1999, pp 67-76.
- [5] R.C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty", *The International Journal of Robotics Research*, Vol. 5, No. 4, 1986, pp 56-68.
- [6] J.K. Uhlmann, S. Julier and M. Csorba, "Nondivergent simultaneous map building and localization using covariance intersection", *SPIE AeroSense, Navigation and Control Technologies for Unmanned Systems II*, Vol. 3087, April 1997, pp 2-11.
- [7] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem", *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 3, June 2001, pp 229-241.
- [8] M. Montemerlo and S. Thrun, *FastSLAM A scalable method for the simultaneous localization and mapping problem in robotics*, Springer, STAR Vol. 27, 2007.
- [9] K.R. Beevers, W.H. Huang, "SLAM with sparse sensing", *IEEE International Conference on Robotics and Automation*, May 2006, pp 2285-2290.
- [10] A. Garulli, A. Giannitrapani, A. Rossi, A. Vicino, "Mobile robot SLAM for line-based environment representation", *IEEE Conference on Decision and Control, and the European Control Conference*, December 2005, pp 2041-2046.
- [11] S.T. Pfister, S.I. Roumeliotis, J.W. Burdick, "Weighted line fitting algorithms for mobile robot map building and efficient data representation", *IEEE International Conference on Robotics and Automation*, Vol. 1, September 2003, pp 1304-1311.
- [12] S.T. Pfister, J.W. Burdick, "Multi-scale point and line range data algorithms for mapping and localization", *IEEE International Conference on Robotics and Automation*, May 2006, pp 1159-1166.
- [13] T. Bailey, *Mobile robot localisation and mapping in extensive outdoor environments*, Ph.D. dissertation, Australian Centre for Field Robotics, University of Sydney, 2002
- [14] M. Altermatt, A. Martinelli, N. Tomatis, R. Siegwart, "SLAM with corner features based on a relative map", *IEEE International Conference on Intelligent Robots and Systems*, Vol. 2, September 2004, pp 1053-1058.
- [15] W.Y. Jeong, K.M. Lee, "Visual SLAM with line and corner features", *IEEE International Conference on Intelligent Robots and Systems*, October 2006, pp 2570-2575.
- [16] O. Wulf, K.O. Arras, H.I. Christensen, B. Wagner, "2D mapping of cluttered environments by means of 3D perception", *IEEE International Conference on Robotics and Automation*, April 2004, pp 4204-4209.
- [17] E. Brunskill, N. Roy, "SLAM using Incremental Probabilistic PCA and Dimensionality Reduction", *IEEE International Conference on Robotics and Automation*, April 2005, pp 342-347.
- [18] G.A. Borges, M.J. Aldon, "Line extraction in 2d range images for mobile robotics", *Journal of Intelligent and Robotic Systems*, Springer, Vol. 40, No. 3, July 2004, pp 267-297.
- [19] V. Nguyen, A. Martinelli, N. Tomatis, R. Siegwart, "A comparison of line extraction algorithms using 2d laser rangefinder for indoor mobile robotics", *IEEE International Conference on Intelligent Robots and Systems*, August 2005, pp 1929-1934.
- [20] R.M. Haralick, "Propagating covariance in computer vision", *IAPR International Conference on Pattern Recognition*, Vol.1, October 1994, pp 493-498.
- [21] R.E. Kalman, "A new approach to linear filtering and prediction problems", *Journal of Basic Engineering*, ASME, 1960, pp 35-45.