A Modular Hybrid Localization Approach for Mobile Robots combining Local Grid Maps and Natural Landmarks

Florian Mirus
Felipe Garcia Lopez
Frank Slomian
Matthias Gruhler
Stefan Dörr
Jürgen Pfadt

Fraunhofer Institute for Manufacturing Engineering and Automation IPA
Nobelstraße 12, 70569 Stuttgart, Germany
{frm, frm-fs, srd, flg, mig, cpc-jp}@ipa.fhg.de

ABSTRACT

This paper presents a hybrid localization approach for mobile robots combining local grid maps and natural landmarks. The approach at hand benefits from the advantages of both environment representations. While using memory-efficient geometric models describing natural landmarks as features for localization in structured regions, the proposed system clusters the remaining areas as raw local grid maps and incorporates those as pose features only for unstructured areas of the environment. To evaluate the functionality and performance of the approach at hand, extensive testing and benchmarking in an experimental setup has been conducted using an external sensor system for reference measurements.

CCS Concepts

•**Computer systems organization** → **Robotics; Robotic autonomy;** *Robotic control; Sensors and actuators;*

Keywords

Mobile robots; Localization; Local grid maps; Extended Kalman Filter

1. INTRODUCTION

Mobile robot localization is the problem of determing the pose, i.e. position and orientation, of a robot relative to a given map of the environment. The localization problem is of crucial importance for almost all tasks in mobile robot applications, especially autonomous navigation. There exists a large variety of solutions for mobile robot localization differing e.g. in the representation of the environment and the algorithm for pose estimation. Common approaches on environment representation are topological maps [9], discretized grid maps [9, 5] and maps based on abstract, geometric features [2]. Because of the high abstraction level of the represented environment, topological maps are rather inapplicable for robot localization, but mainly used for planning tasks. While localization approaches relying exclusively on geometric features show sound results in environments holding smooth structures, these methods tend to fail in highly unstructured environments, like e.g. factory floors, where almost no explicit features can be extracted from the sensor data. On the other hand, grid-based localization approaches, like e.g. Adaptive Monte Carlo Localization (AMCL), are able to tackle these issues without introducing artificial landmarks at the cost of high memory consumption, especially in spacious environments, for the storage of a global gridmap. Additionally, this approach can hardly integrate different types of sensors due to the occupancy grid based environment representation, which can only be updated with sensor data measuring the occupancy of a certain region. In turn, the use of abstract features usually requires the extraction of these features from raw sensor data and association with given map features. While omitting these additional steps, the use of grid maps requires matching of incoming raw sensor data with the stored grid cells, e.g. directly [3], using histogram matching [1] or probabilistic scan matching [6]. There already exist some localization methods [4], which combine the advantages of feature and grid maps, but most hybrid approaches combine topological maps and occupancy grids in terms of path-planning [8].

This paper presents a modular localization system for mobile robots combining local grid maps and natural landmarks using an Extended Kalman Filter (EKF) based on abstract features. While using memory-efficient geometric models describing natural landmarks as features for localization in structured regions, the proposed system clusters the remaining areas as raw local grid maps and incorporates those as pose features only for unstructured areas of the environment. The main contributions of this paper are considered to be the generic structure and implementation (see Sec. 2), which supports different types of features and a mix of sensors, and the method described in Sec. 3.1 to identify unstructured areas which require local grid maps.

SAC 2016, April 04-08, 2016, Pisa, Italy

ACM 978-1-4503-3739-7/16/04…$15.00

DOI: http://dx.doi.org/10.1145/2851613.2851929

2. LOCALIZATION SYSTEM

2.1 Sensor fusion system

The general approach to mobile robot localization used in the work at hand provides a probabilistic sensor fusion system based on abstract features, which are independent of the underlying sen-

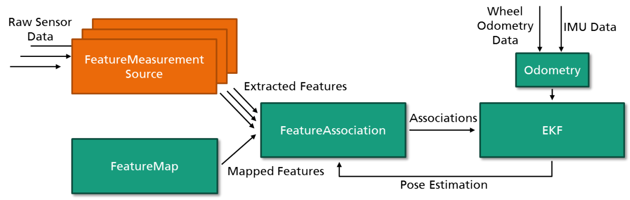


Figure 1: Schematic software architecture of the localization system.

sor raw data, using an Extended Kalman Filter (EKF) taking process and measurement uncertainties into account. The actual EKF-implementation follows closely the description given in [10] and provides a plugin-structure with standard interfaces for each feature-type. The whole system is implemented in C++ within the framework of the Robot Operating System (ROS) [7]. Fig. 1 shows a schematical description of the localization software architecture, illustrating the sensor specific modules *Feature Measurement Source* resp. *Feature Assumption Source* on the left side, which provide measured features resp. feature maps. The rest of the system is sensor independent.

2.2 Feature types

In principle, every geometric feature can be used as input for the localization system at hand. Common feature types are lines, points and poses. Especially in human related environments like office buildings or appartments, which are dominated by linear or rectangular structures, lines and corners extracted from two-dimensional laser range data are good examples for natural line and pose features respectively. In addition, artificial landmarks like reflective markers or RFID-tags (Radio-Frequency Identification), which are handled as point features, can be used to increase localization robustness in industrial environments holding fewer or not any natural landmarks. Even external measurements of the robot pose (not requiring any maps), for example coming from external tracking systems or from a SLAM-algorithm (Simultaneous Localization And Mapping) running in parallel, can be integrated as pose features into the localization system at hand.

2.3 Extraction and association

The task of the sensor specific extraction modules is to compute a set of feature measurements and corresponding covariance matrices taking into account the sensor specific uncertainty of the raw data as well as the estimated uncertainty of the extraction. A single *Feature Measurement Source* is able to output different feature types as, for instance, an extraction module for twodimensional laser range data is able to extract lines and poses for corner features at a time. A method for extracting lines and corners from 2D laser range date is described e.g. in [2]. Given the extracted features, the *Feature Association* module matches those features with its corresponding map features originating from the *Feature Assumption Source*. If the extracted feature has an unique identifier, like e.g. point features detected by RFID antennas, which are able to read the ID of the measured tag, this association is trivial. However, in most cases the association is a priori unknown and needs to be estimated. A common and effective approach is the Nearest Neigbor Search (NNS), which computes for each extracted feature its closest correspondence in the map given a specific distance measure, commonly the Mahalanobis distance. The result of the association step is a list of corresponding features, which is processed in the EKF for updating the predicted pose of the robot.

3. LOCAL GRID FEATURES

In general, it is possible to perform robust and reliable mobile robot localization using the proposed system described in Sect. 2 relying exclusively on abstract features. But there are environments and use cases with few or almost no usable natural landmarks, where the installation of artificial landmarks to increase localization robustness is not desired or simply impossible and the storage of a global grid map is impractical and thus not desired. Especially environments, where structured and unstructured areas alternate and major parts of the map can efficiently be described by abstract features, call for grid based localization only for local areas of the map.

3.1 Local map generation

To extract only relevant, i.e. unstructured, parts of the recorded global grid map, only occupied grid cells, which can not be associated to a feature should be regarded. To identify the relevant cells, the grid map is compared with a given feature map. All occupied grid cells G_i closer than a parameter d_l for the maximal euclidean distance to a feature f_j are not considered to be part of an unstructured area and thus discarded. The set of all occupied grid cells is denoted by $\mathscr{G} := \{G_i \mid i = 1,...,n\}$. The set of discarded cells $\mathscr{O}$ is denoted by

$$\mathscr{O} := \left\{G_i \,|\, d(G_i, f_j) < d_l, G_i \in \mathscr{G}, j \in \{1,...,m\}\right\}, \qquad (1)$$

where $d(G_i, f_j)$ is defined as the shortest distance from the center of a grid cell G_i to a feature f_j. The set $\mathscr{U} := \mathscr{G} \setminus \mathscr{O}$ contains only grid cells, which are not considered to represent a mapped feature. The next step is to split up $\mathscr{U}$ into smaller coherent sets of grid cells, which form the local maps. If the distance $\tilde{d}(G_i, G_j)$ between the center of two cells G_i, G_j is smaller than a parameter d_c, the cells are considered to belong to the same area. The set of grid cells which form a local grid map candidate B_j is calculated iteratively by

$$B_j := \left\{ G_i, G_j \in \mathscr{U} \setminus \bigcup_{k=1}^{j-1} B_k \,\middle|\, \tilde{d}(G_i, G_j) < d_c, i \neq j \right\}. \qquad (2)$$

The set of all local grid candidate areas is denoted by $\mathscr{B} := \{B_i \mid i = 1,...,N\}$. Since during this process the creation of several small or overlapping areas is very likely, an additional merge step is performed. The rectangular local areas are represented by their bottom left point E_1 and the top right point E_3. Two areas are considered overlapping if at least one corner E_{ik} with $k \in \{1,\dots,4\}$ of area $B_i \in \mathscr{B}$ lies inside another area $B_j \in \mathscr{B}$. The set of overlapping areas $\tilde{\mathscr{B}}$ is therefore given by

$$\tilde{\mathscr{B}} := \{\, B_i \in \mathscr{B} \mid \exists B_j \in \mathscr{B} \text{ with } i \neq j \text{ and } \\ \exists k \in \{1,...,4\} : E_{ik} \in B_i \cap B_j \}. \qquad (3)$$

By unifying overlapping areas in $\tilde{\mathscr{B}}$, one gets a new set of local areas $\mathscr{B}$. Since it is possible to create new overlaps during the merge-process, this procedure is repeated iteratively until all overlaps are removed. Fig. 2 illustrates the process for local grid map generation.

3.2 Extraction and association of local unstructured areas

To extract only scan points belonging to unstructured areas, incoming scan measurements, which can be related to measured features,

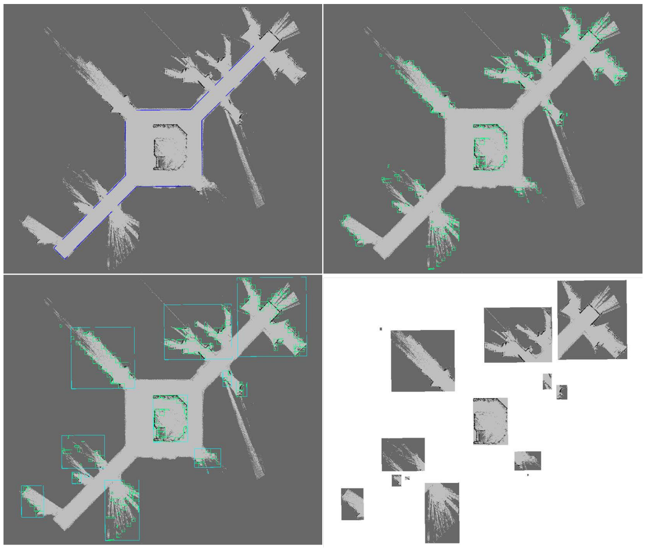


Figure 2: Local map generation procedure (exemplary illustration with the system considering lines as structured features).

have to be refrained. Therefore, the set of incoming scan points L_i, denoted by $\mathscr{L} := \{L_i \mid i = 1,...,r\}$, is compared with the set of measured lines features $\mathscr{F} := \{f_j \mid j = 1,\ldots,s\}$. Similar to the calculation in Eq. 1, the set $\mathscr{O}^L$ of laser scan points, whose distance $d(L_i, f_j)$ to a measured feature f_j is under a given threshold d_l, is calculated. The set $\mathscr{U}^L := \mathscr{L} \setminus \mathscr{O}^L$ contains only scan points, which are not considered to belong to a measured feature. Following the merge-procedure used for local map generation described in Sect. 3.1, the set of measured unstructured areas after removal of overlapping areas is denoted by $\mathscr{B}^L$.

A measured unstructured area is associated to a given local grid map if they overlap, i.e. one corner point E_{ik} with $k \in \{1,\ldots,4\}$ of an unstructured area $B_i \in \mathscr{B}^L$ lies within a given map $m_j \in \mathscr{M} := \{m_j \mid j = 1,...,p\}$. The set $\mathscr{A}^L$ of associated areas and the set $\mathscr{A}^m$ of associated local maps is therefore denoted by

$$\mathscr{A}^L := \{\, B_i \in \mathscr{B}^L \mid \exists m_j \in \mathscr{M} \text{ and } \exists k \in \{1,...,4\} : E_{ik} \in B_i \cap m_j\}, \tag{4a}$$

$$\mathscr{A}^m := \{\, m_i \in \mathscr{M} \mid \exists B_j \in \mathscr{A}^L \text{ and } \exists k \in \{1,...,4\} : E_{jk} \in B_j \cap m_i\}. \tag{4b}$$

3.3 Matching and position correction

The sets $\mathscr{A}^L$ and $\mathscr{A}^m$ defined in Sect. 3.2 give the association of measured unstructured areas and local grid maps. The next step is to calculate the mismatch between measurement and map, i.e. the transformation $(R,\vec{t})$ consisting of a rotation matrix R and a translation vector $\vec{t}$, which maps the points $\mathscr{P} := \{p \in B_i\}$ of a measured unstructured area $B_i \in \mathscr{A}^L$ to the cells $\mathscr{Q} := \{q \in m_j\}$ of a local grid map $m_j \in \mathscr{A}^m$ in optimal fashion. Therefore, the iterative closest point algorithm (ICP), which finds the optimal function

$$\phi_{R,\vec{t}} : \mathscr{P} \to \mathscr{P}(R,\vec{t});\; p \mapsto \tilde{p} := R \cdot p + \vec{t} \tag{5}$$

depending on R and $\vec{t}$, is applied. This information is subsequently passed to the localization system to estimate the current robot pose after processing all available feature sources.

4. EXPERIMENTAL EVALUATION

4.1 Reference sensor system

The external tracking systems used in the work at hand are the Optitrack V120:Duo and Optitrack V120:Trio. These systems use two resp. three infrared cameras and are able to track the movements of the six degrees of freedom of a rigid body equipped with at least three reflective markers down to sub-millimeter accuracy[1].

4.2 Setup and calibration

To make the robot pose estimated by the localization system and the tracked pose from the Optitrack systems comparable, the tracking system needs to be calibrated with reference to the given map, which is also used for localization. Since the Optitrack systems provide precise tracking results in sub-millimeter range but only in a limited area in front of the cameras, the systems are placed at two reference areas, where localization precision is measured. During an evaluation run, the robot navigated alternately between these positions recording sensor data from the two laser sensors, odometry information as well as the tracked poses from the Optitrack systems to provide comparable and identical input data for the localization algorithms to be evaluated. These algorithms run offline and their localization precision is evaluated by comparing the recorded pose provided by the Optitrack systems, which is considered as *ground truth*, with the currently estimated localization pose when the robot enters the reference areas. The robot used for evaluation in the work at hand is Fraunhofer IPA's rob@work3[2].

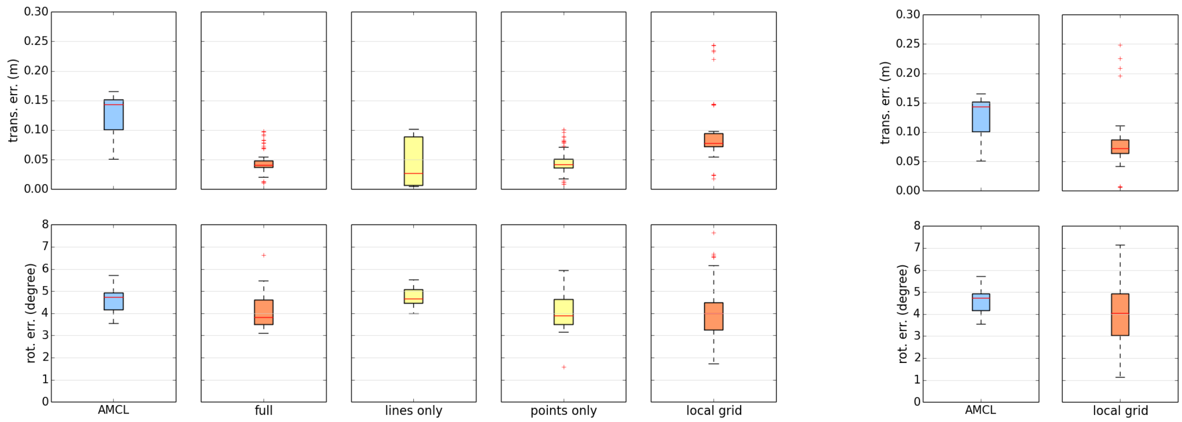
4.3 Test scenarios

To revise the functionality and performance of the proposed localization approach, the system was tested in two different environments using the setup described in Sect. 4.2. These environments differ in the amount of geometric features and are therefore entitled *structured* and *unstructured*. To give a comprehensive overview over the system's capabilities, several configurations incorporating different feature sources have been evaluated. The features used in this evaluation are line features extracted from raw laser sensor data, reflective markers extracted from the intensities of the laser scan considered as point features and the local grid features described in Sect. 3, which are considered as pose features. The actual evaluated configurations are *full* (all available features), *lines only*, *points only* (only reflective markers) and *local grids* (using local grid features mainly and line features where available). To give an idea how the system at hand performs in comparison with a state of the art localization system, the open-source implementation of the AMCL[3] algorithm, which closely follows the description in [10], has been evaluated for reference.

4.4 Results

Fig. 3 illustrates the results of the experimental evaluation described in the preceding sections in both *structured* (Fig. 3a) and *unstructured* (Fig. 3b) environments. The proposed localization system outperforms AMCL in the *structured* environment in every

[1] http://www.optitrack.com/products/v120-trio/indepth.html
[2] http://www.care-o-bot.de/en/rob-work.html
[3] http://wiki.ros.org/amcl



(a) Structured environment.

(b) Unstructured environment.

Figure 3: Evaluation results: the upper resp. lower section depicts the translational resp. rotational localization error. Configurations with and without local grids are illustrated by *orange* resp. *yellow* boxes while the reference system AMCL is depicted by *blue* boxes.

configuration, where the best results (5 cm and 4° median translational and rotational error) are achieved, as expected, when using all available feature sources with configuration *full* (see Fig. 3a). In unstructured environments, the localization relying exclusively on abstract features fails due to the lack of adequate input data, but by incorporating local grids, the performance of the proposed localization system is in the same range as AMCL or slightly better (see Fig. 3b).

5. CONCLUSIONS

This paper presented a modular localization system for mobile robots combining local gridmaps and geometric models, which shows promising results. The actual implementation provides a flexible, sensor-independent plugin-structure and therefore allows the incorporation and combination of various sensors. The precision of several configurations of the proposed localization system has been evaluated and compared with the state of the art localization system AMCL. The proposed system outperforms AMCL in both, structured and unstrcutured environments in the test setup at hand. The translational error in structured environments with sufficient availablable features can even be expected to be in the range of 5 cm, which corresponds to the measurement accuracy of the laser sensor.

Possible directions for future research are the implementation of a SLAM-algorithm, which is able to generate local grid maps online, instead of creating them offline from a prerecorded global grid map either manually or using the method decribed in Sect. 3.1 or to analyse more sophisticated matching approaches than ICP, which could improve the matching of the local grid maps. The modular structure of the localization system at hand also allows to replace the current EKF-implementation by a particle filter to incorporate highly non-linear measurement and motion models.

6. REFERENCES

[1] M. Bosse and J. Roberts. Histogram matching and global initialization for laser-only slam in large unstructured environments. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 4820–4826. IEEE, 2007.

[2] C. P. Connette, O. Meister, M. Hägele, and G. F. Trommer. Decomposition of line segments into corner and statistical grown line features in an EKF-SLAM framework. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3884–3891. IEEE, 2007.

[3] A. Diosi and L. Kleeman. Laser scan matching in polar coordinates with application to slam. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3317–3322. IEEE, 2005.

[4] J. Guivant, E. Nebot, J. Nieto, and F. Masson. Navigation and mapping in large unstructured environments. *The International Journal of Robotics Research*, 23(4-5):449–472, 2004.

[5] E. Ivanjko et al. Extended Kalman Filter based mobile robot pose tracking using occupancy grid maps. In *Electrotechnical Conference, 2004. MELECON 2004. Proceedings of the 12th IEEE Mediterranean*, volume 1, pages 311–314. IEEE, 2004.

[6] L. Montesano, J. Minguez, and L. Montano. Probabilistic scan matching for motion estimation in unstructured environments. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3499–3504. IEEE, 2005.

[7] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, 2009.

[8] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.

[9] S. Thrun and A. Bücken. Integrating grid-based and topological maps for mobile robot navigation. In *Proceedings of the National Conference on Artificial Intelligence*, pages 944–951, 1996.

[10] S. Thrun, W. Burgard, and D. Fox. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents), 2005.