2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

A Region-based SLAM Algorithm Capturing Metric, Topological, and Semantic Properties

Jan Oberländer, Klaus Uhl, J. Marius Zöllner and Rüdiger Dillmann
FZI Forschungszentrum Informatik
Department of Interactive Diagnosis and Service Systems
76131 Karlsruhe, Germany
{oberlaender,uhl,zoellner}@fzi.de, dillmann@ira.uka.de

Abstract— **This paper proposes a SLAM algorithm based on FastSLAM 2.0 that maps features representing regions with a semantic type, topological properties, and an approximative geometric extent. The resulting maps enable spatial reasoning on a semantic level and provide abstract information allowing efficient semantic planning and a convenient interface for human-machine interaction. We present novel region features and an algorithm for estimating the feature parameters from uncertain measurements. In particular, we provide a means of estimating parameters even if the region feature is considerably larger than the robot's sensor range. Finally, we adapt the FastSLAM 2.0 algorithm to map the proposed features and show simulation-based results illustrating the capabilities of the proposed algorithm.**

I. INTRODUCTION

For a robot to achieve full autonomy, it needs to be able to learn its environment and store the acquired knowledge in an efficient manner. The resulting world model must enable the robot to efficiently plan and to reliably execute the tasks given to it. An abstract semantic environment representation in combination with a semantic navigation framework is beneficial in several ways. It allows the navigation framework to plan the robot's actions in an abstract manner, independent of the underlying controlling system. In addition, a multi-level representation of the robot's pose comprising metric and semantic information leads to more robust navigation.

In this work, we present a method for building maps which include metric as well as topological and semantic information. We propose new environmental features, called *regions*, which allow an abstract but rich representation of the environment. Along with a metric representation using simple geometric shapes, regions contain topological links to adjacent regions as well as subregions and further provide a semantic type. We extend the FastSLAM 2.0 algorithm [1] to map these features.

In recent years, *Simultaneous Localization and Mapping* (SLAM) has received much attention as one of the cornerstones of robot autonomy. Starting with Smith et al.'s seminal Kalman filter-based approach to SLAM [2], metric environment mapping has been an active area of research [3][4]. Purely metric maps, however, do not address all the challenges pointed out above, and with increasing environment size also tend to require a large amount of computational resources or create inconsistent maps [5][6]. To allow efficient planning, a more abstract environment representation is needed. Topological maps are a useful approach to obtain such a representation [7][8], but metric information cannot be discarded as it is required for local navigation, especially in more complex environments [9][10][11]. An efficient environment representation will in many respects be similar to the human cognitive map [12][13] which represents the environment at different levels of abstraction.

To allow a human to effectively communicate with a robot to supply it with tasks and retrieve information on its status, the robot should maintain semantic information. It is much more convenient for a human to be able to describe a task using human semantics, such as "Fetch the lunch from the cafeteria on third floor and deliver it to dining hall 3 on first floor" [14]. Similarly, a system reporting its status as "I am in the hallway on the second floor of building 2, next to the door leading to the kitchen" is much more useful to its user and is essential to the user's ability to interact with the robot without any detailed technical knowledge of the system. Some works have captured semantic information on the types of places and objects observed, proposing environment representations with semantic information and classifying elements of the environment using range and camera data [15][16][17].

The remainder of this paper is structured as follows: Section II describes the proposed region features and the measurement update for these features. In Section III we present the region-based FastSLAM algorithm. Section IV provides experimental validation of the measurement update process and the FastSLAM implementation.

II. REGION FEATURES

In 2-D space, feature-based SLAM algorithms mostly map low-level environmental features such as points or lines. This paper presents a new type of high-level feature providing much richer, abstract information of the environment. These *region features* define regions in the robot's environment carrying a specific semantic meaning (e. g., rooms, offices, hallways, doors, desks, cupboards, etc.) as well as a geometric layout and topological relations.

A. Design Criteria

The *metric description* of a region feature should be capable of modeling regions likely to be found in structured indoor environments such as those mentioned above. The number of required parameters should be as small as possible to limit complexity and ambiguities. Uncertainty needs to be captured by a probabilistic model. We favor regions with a geometric extent over simple topological maps of connected points due to their ability to express logical, spatial relations such as being "in front of," "next to," or "in the middle of" a particular region. As a special quality, region features should be able to handle *partial measurements*: due to its limited sensor range, a robot observing a long hallway may not be able to observe the entire hallway at any point.

The *topological description* of a region feature should model same-level links to neighboring regions as well as hierarchical links to subregions. To model uncertainties, these links should be associated with a confidence value.

The *semantic description* of a region should comprise a *semantic type* for the region, along with a confidence value to model uncertainty. We assume that if multiple hypotheses with regard to a region's semantic type exist, they will be modeled by distinct features with similar geometric shape but a different semantic type. The mapping algorithm should infer the correct type from a series of measurements.

B. Feature Parameters

Most regions found in a typical indoor environment can be approximated well by rectangles. We therefore model regions geometrically as shapes consisting of up to three rectangles lined up along the horizontal axis. The entire shape may be rotated arbitrarily. This is powerful enough to represent most commonly encountered shapes and only requires a small amount of parameters, leaving little ambiguity. More complex geometric models would significantly increase the computational complexity, without providing much benefit in terms of their representational capabilities. Fig. 1 shows all the parameters comprising a region feature. An example region consisting of three rectangles is shown in Fig. 2.

Topological information is stored via lists of links associating a neighboring region with a confidence value. We link individual rectangle edges to neighboring regions. The semantic type of a region is given as a simple text string, making the representation independent of any particular set of predefined types. Again, a confidence value is associated with the type field.

Fig. 1 shows that certain assumptions are made about the measurement uncertainty regarding the geometric properties of region features. In particular, we assume that

- the rotation angle φ,
- the main rectangle's midpoint position (x, y),
- the main rectangle's height (w, h),
- the left subrectangle's width and height (w_l, h_l),
- the left subrectangle's vertical displacement y_l,
- the right subrectangle's width and height (w_r, h_r), and
- the right subrectangle's vertical displacement y_r

are pairwise stochastically independent. While this is a limitation, it makes it possible to factor the error covariance into several smaller covariance matrices, which simplifies the update mechanism.

enum EdgeState {*open, closed*}
enum ExtEdgeState {*open, closed, subrect*}

class RegionFeature
  string *type* // *Semantics*
  double λ_{type}
  double x, y, w, h, φ // *Geometry*
  double[2×2] C_{xy}, C_{wh}
  double C_φ
  EdgeState t, b // *Topology*
  ExtEdgeState l, r
  map⟨RegionFeature, double⟩ *children*
  map⟨RegionFeature, double⟩ n_t, n_b
  if $l = subrect$: // *Left subrectangle*
    double w_l, h_l, y_l // *Geometry*
    double[2×2] $C_{w_l h_l}$
    double C_{y_l}
    EdgeState t_l, b_l, l_l // *Topology*
    map⟨RegionFeature, double⟩ n_{lt}, n_{lb}, n_{ll}
  else:
    map⟨RegionFeature, double⟩ n_l
  if $r = subrect$: // *Right subrectangle*
    double w_r, h_r, y_r // *Geometry*
    double[2×2] $C_{w_r h_r}$
    double C_{y_r}
    EdgeState t_r, b_r, r_r // *Topology*
    map⟨RegionFeature, double⟩ n_{rt}, n_{rb}, n_{rr}
  else:
    map⟨RegionFeature, double⟩ n_r

Fig. 1. The data structure representing a region feature. It comprises metric information (such as the parameters x, y, w, h along with their covariance matrices) as well as topological information (such as given by the maps n_t, n_b) and semantic information (given by the *type* string and the associated confidence λ_{type}).

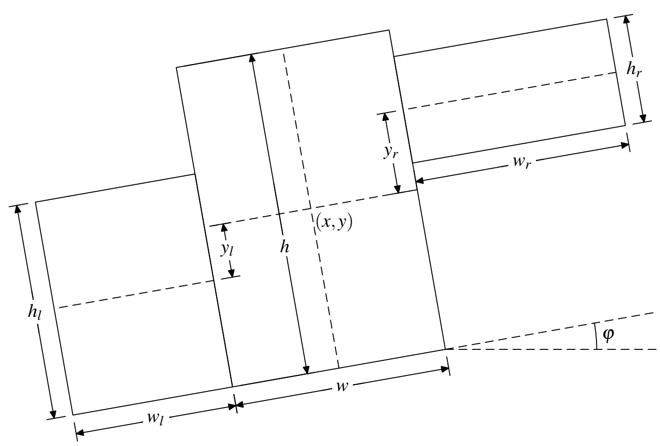


Fig. 2. Regions are comprised geometrically of up to three rectangles, capable of describing most rooms and objects in structured indoor environments.

To accommodate partial observations of features, individual rectangle edges can be marked *open* to indicate missing information in that direction. This has a significant effect on how the measurement update works.

C. Measurement Update

Before the parameters of a region m can be updated using a corresponding new measurement z, a second correspondence problem needs to be solved associating the subrectangles of the two regions. This would be unnecessary if only complete region measurements were obtained. If partial measurements are possible, however, this additional association step becomes a necessity. For example, given a mapped region m consisting of three rectangles (shape "LMR"), a measurement z might only comprise two rectangles (shape "LM") with open edges on the left and right. The main rectangle of z may then correspond to either the main rectangle or the right subrectangle of m. The best correspondence is determined by finding the minimum distance between any two rectangle midpoints from the two features, constrained by the fact that the regions must be *compatible*: one region cannot have a closed edge where the other indicates the presence of a subrectangle, and the resulting region cannot consist of more than three rectangles.

1) Geometric Update: The geometric parameters for a region m, $(x^{[m]}, y^{[m]}, w^{[m]}, h^{[m]}, \varphi^{[m]})$ and for subrectangles $(w_l^{[m]}, h_l^{[m]}, y_l^{[m]})$ and $(w_r^{[m]}, h_r^{[m]}, y_r^{[m]})$ respectively, need to be treated differently depending on the state of rectangle edges in the map feature and in the new measurement. We call a parameter *complete* if none of the related rectangle edges are marked *open* and the feature is thus entirely visible. For instance, the height parameter for the main rectangle, $h^{[m]}$, is only complete if neither the top nor the bottom edge of the main rectangle are open. If some, but not all, related rectangle edges are marked *open*, we call the parameter *partially available*. If all related edges are marked *open*, we call the parameter *restricted*. This terminology applies to all parameters except the rotation angle $\varphi^{[m]}$, which is considered to always be available. The geometric update then proceeds in the following manner:

1) All partially available or restricted parameters of m are first *expanded* so that the region becomes large enough to explain both the existing data and the new measurement. If a subrectangle is first observed in z, it is added to m.
2) All partially available or restricted parameters of z are expanded in the same manner.
3) A Kalman filter-based update is performed on all parameters of m for which corresponding complete or partially available parameters exist in z.

In particular, note that even restricted parameters of the measurement provide information about m, provided that the corresponding parameters of m are not yet complete.

The initial expansion step uses the measurement z to enlarge m in the direction of open edges to make it consistent with z. Fig. 3 shows the principle. For the individual open edges of m, expansion terms can be calculated by projecting the measurement's corresponding edge midpoints onto the respective principal axes of m and deriving a length difference. Assuming correspondence of the main rectangles of m and z, we get

$$d_t^{[m]} = \begin{pmatrix} -\sin\varphi^{[m]} & \cos\varphi^{[m]} \end{pmatrix} \begin{pmatrix} x^{[z]} - x^{[m]} \\ y^{[z]} - y^{[m]} \end{pmatrix} + \tfrac{1}{2}\left(h^{[z]} \cos(\varphi^{[z]} - \varphi^{[m]}) - h^{[m]}\right) \quad (1)$$

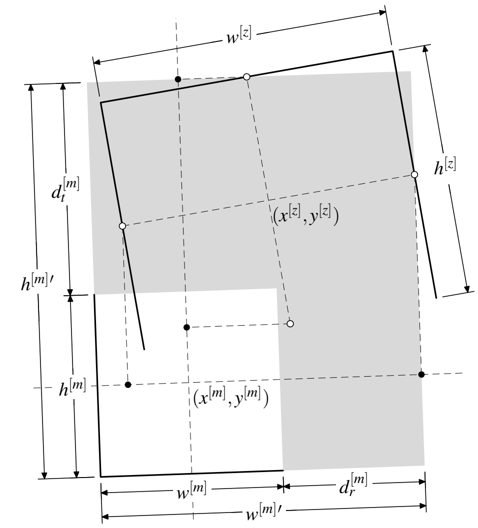


Fig. 3. Expansion of a rectangle with open edges. The map feature m (white square in the bottom left corner) can be expanded in the direction of its open edges, i. e. to the top and right, using projections of the measured region's edge midpoints, according to (1) and (2). The resulting expanded rectangle is shown in light gray.

$$d_r^{[m]} = \begin{pmatrix} \cos\varphi^{[m]} & \sin\varphi^{[m]} \end{pmatrix} \begin{pmatrix} x^{[z]} - x^{[m]} \\ y^{[z]} - y^{[m]} \end{pmatrix} + \tfrac{1}{2}\left(w^{[z]} \cos(\varphi^{[z]} - \varphi^{[m]}) - w^{[m]}\right), \quad (2)$$

and analogous terms for $d_b^{[m]}$ and $d_l^{[m]}$. According to these terms, the region m is expanded

- upwards by $d_t^{[m]}$ if $t^{[m]} = open$ and $d_t^{[m]} > 0$,
- downwards by $-d_b^{[m]}$ if $b^{[m]} = open$ and $d_b^{[m]} < 0$,
- to the right by $d_r^{[m]}$ if $r^{[m]} = open$ and $d_r^{[m]} > 0$, and
- to the left by $-d_l^{[m]}$ if $l^{[m]} = open$ and $d_l^{[m]} < 0$.

Similar terms can be derived for subrectangles and other rectangle correspondences [18].

The measurement z is expanded in the same manner, except in one particular case. If the corresponding parameters in m are *complete*, then we assume that they are correct except for the length differences given by those edges that are closed in both m and z. This is illustrated in Fig. 4.

After expansion, the Kalman filter update is applied to complete or partially available measurement parameters. As an example, the update step is illustrated using the main rectangle's width and height. The estimated state is thus $x = (w^{[m]}, h^{[m]})^T$. Given a direct observation model

$$z_t = x + \delta_t, \qquad \delta_t \sim \mathcal{N}(0, Q_t), \quad (3)$$

the state is estimated as

$$K_t = \Sigma_{t-1}(\Sigma_{t-1} + Q_t)^{-1} \quad (4)$$
$$\mu_t = \mu_{t-1} + K_t(z_t - \mu_{t-1}) \quad (5)$$
$$\Sigma_t = (I - K_t)\Sigma_{t-1} \quad (6)$$

with K_t being the Kalman gain, μ_t the state estimate and Σ_t the estimation error covariance.

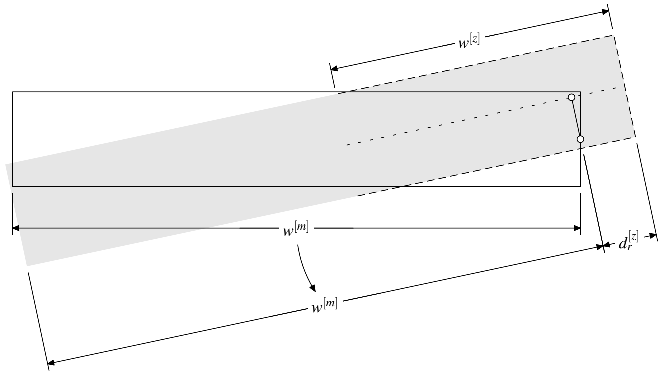


Fig. 4. The partially available width of the measurement z (dashed rectangle) is expanded according to the *complete* width of m (solid rectangle). The difference $d_r^{[z]}$, given by comparing the available edges of m and z on the right, is the only valid piece of information regarding the feature width. z is expanded to the same width as m, differing only by $d_r^{[z]}$. The resulting expanded version of z is shown in light gray.

2) Semantic Update: A measurement z bears a specific semantic type along with a confidence $p(z) \in [0,1]$. Multiple measurements of the same object with different types may be present. For the type confidence $\lambda^{[m]}_{\text{type},t}$ of a mapped region feature m, we use a binary Bayes filter in log-odds form to perform the update:

$$\lambda^{[m]}_{type,t} = \lambda^{[m]}_{type,t-1} + \log \frac{p(m\,|\,z_t)}{1-p(m\,|\,z_t)} - \log \frac{p(m)}{1-p(m)}. \tag{7}$$

Here, $p(m)$ represents the a-priori probability for a feature with the type and dimensions of m. This can simply be chosen as $p(m) = 0.5$ or incorporate more sophisticated prior knowledge. $p(m\,|\,z_t)$ is the inverse measurement model describing the probability of m existing, given a set of *relevant measurements*

$$z_t = \{z_t^{[1]}, \dots, z_t^{[L_t]}, z_t^{[L_t+1]}, \dots, z_t^{[L_t+M_t]}\}, \tag{8}$$

where the first L_t measurements *support* the hypothesis m, i.e. have the same semantic type, and the remaining M_t measurements *oppose* it, i.e. have a different type. All supporting and opposing evidence is then summed as

$$p^{m,z_t}_{\text{supp}} = \begin{cases} L_t^{-1} \sum_{i=1}^{L_t} p(z_t^{[i]})\,\text{overlap}(z_t^{[i]}, m), & L_t > 0 \\ 0.5, & L_t = 0 \end{cases} \tag{9}$$

$$p^{m,z_t}_{\text{opp}} = \begin{cases} M_t^{-1} \sum_{i=1}^{M_t} p(z_t^{[L_t+i]})\,\text{overlap}(z_t^{[L_t+i]}, m), & M_t > 0 \\ 0.5, & M_t = 0 \end{cases} \tag{10}$$

where

$$\text{overlap}(z,m) = a_{z\cap m}/a_z \in [0,1] \tag{11}$$

is the ratio of z's area intersecting with m. This is then used to define the inverse measurement model as

$$p(m\,|\,z_t) = (1 + \exp\{p^{m,z_t}_{\text{opp}} - p^{m,z_t}_{\text{supp}}\})^{-1}. \tag{12}$$

For the update step of the binary Bayes filter, we thus obtain the simple formula

$$\lambda^{[m]}_{type,t} = \lambda^{[m]}_{type,t-1} + p^{m,z_t}_{\text{supp}} - p^{m,z_t}_{\text{opp}} - \log \frac{p(m)}{1-p(m)}. \tag{13}$$

3) Topological Update: Given two mapped region features m,n and a set of measurements z_t with $z_{m,t}, z_{n,t} \in z_t$ being measurements of the two features, we assume that we receive a measured confidence $p(z_{m,t} \to z_{n,t})$ for a topological link between the measurements. The inverse measurement model for the topological link $m \to n$ is then calculated as

$$p(m \to n\,|\,z_t) = \begin{cases} p_\varnothing, & z_{m,t} \not\to z_{n,t} \\ p(z_{m,t} \to z_{n,t}), & z_{m,t} \to z_{n,t} \end{cases} \tag{14}$$

if both m and n are measured, and

$$p(m \to n\,|\,z_t) = 0.5 \tag{15}$$

otherwise. $p_\varnothing$ is a predefined probability that a link exists even if both regions were observed, but not the link between them. The log-odds confidence of the topological link $m \to n$ is then updated using the binary Bayes filter

$$\lambda_t(m \to n) = \lambda_{t-1}(m \to n) + \log \frac{p(m \to n\,|\,z_t)}{1 - p(m \to n\,|\,z_t)} - \log \frac{p(m \to n)}{1 - p(m \to n)}\ , \tag{16}$$

where $p(m \to n)$ provides an a-priori probability for a topological link between regions of the given types.

Hierarchical relations are described by log-odds confidence values $\lambda_t(m \prec n)$ representing the belief that region m is a subregion of region n. Similar to (16), we get

$$\lambda_t(m \prec n) = \lambda_{t-1}(m \prec n) + \log \frac{p(m \prec n\,|\,z_t)}{1 - p(m \prec n\,|\,z_t)} - \log \frac{p(m \prec n)}{1 - p(m \prec n)}\ . \tag{17}$$

The a-priori probability $p(m \prec n)$ could, for instance, indicate that tables are likely to be found inside rooms, but not vice versa. We define the inverse measurement model by

$$p(m \prec n \mid z_t) = \text{overlap}(z_{m,t}, z_{n,t}) \min\{a^2_{n,t}/a^2_{m,t}, 1\}\ , \tag{18}$$

taking into account the relative sizes of the two measurements.

Using the three described update methods for geometrical, semantic, and topological properties of regions, we can update the map of all regions with a set of measurements.

III. Region-based SLAM

In this section we present a semantic SLAM algorithm using the features defined above, based on the FastSLAM 2.0 algorithm [1], a SLAM approach based on a Rao-Blackwellized particle filter, with unknown data association [19]. The Kalman filter update step is replaced by the update method described in Section II-C. Several steps of the FastSLAM 2.0 algorithm require the calculation of an innovation term $(z - \hat{z})$ expressing the difference between actual and expected measurements. For a pose estimate $\hat{x}$ and the expected measurement $\hat{z} = h(m, \hat{x})$, a naïve way of calculating the difference would use the midpoint and rotational difference to obtain

$$z - \hat{z} = (x^{[z]} - x^{[\hat{z}]}, y^{[z]} - y^{[\hat{z}]}, \varphi^{[z]} - \varphi^{[\hat{z}]})^T. \tag{19}$$

This, however, is incorrect in the presence of partial measurements. Instead, after determining the rectangle association between z and $\hat{z}$, the difference is calculated from the individual differences in the position of edges closed in both z and $\hat{z}$. The rotational difference is given by

$$\Delta\varphi = \varphi^{[z]} - \varphi^{[\hat{z}]}. \tag{20}$$

$\hat{z}$ is rotated accordingly to yield

$$\begin{pmatrix} x^{[\hat{z}]\prime} \\ y^{[\hat{z}]\prime} \\ \varphi^{[\hat{z}]\prime} \end{pmatrix} = \begin{pmatrix} \cos\Delta\varphi & -\sin\Delta\varphi & 0 \\ \sin\Delta\varphi & cos\Delta\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x^{[\hat{z}]} \\ y^{[\hat{z}]} \\ \varphi^{[\hat{z}]} + \Delta\varphi \end{pmatrix} . \tag{21}$$

Writing the difference of the feature midpoints in z's coordinate system as

$$d_c = \begin{pmatrix} \cos\varphi^{[z]} & \sin\varphi^{[z]} \\ -\sin\varphi^{[z]} & \cos\varphi^{[z]} \end{pmatrix} \begin{pmatrix} x^{[z]} - x^{[\hat{z}]} \\ y^{[z]} - y^{[\hat{z}]} \end{pmatrix} , \tag{22}$$

we can now calculate horizontal and vertical differences between pairs of closed edges. Assuming again correspondence of the main rectangles, we obtain for the main rectangle,

- for $l^{[z]} \neq open \wedge l^{[\hat{z}]} \neq open$: $d_x = d_{c,x} - (w^{[z]} - w^{[\hat{z}]})/2$
- for $r^{[z]} \neq open \wedge r^{[\hat{z}]} \neq open$: $d_x = d_{c,x} + (w^{[z]} - w^{[\hat{z}]})/2$
- for $b^{[z]} \neq open \wedge t^{[\hat{z}]} \neq open$: $d_y = d_{c,y} - (h^{[z]} - h^{[\hat{z}]})/2$
- for $t^{[z]} \neq open \wedge b^{[\hat{z}]} \neq open$: $d_y = d_{c,y} + (h^{[z]} - h^{[\hat{z}]})/2$

Averaging these individual differences gives a position difference of $(\bar{d}_x, \bar{d}_y)^T$ in the coordinate system of z, from which we obtain the final difference

$$(z - \hat{z}) := \begin{pmatrix} \cos\varphi^{[z]} & -\sin\varphi^{[z]} & 0 \\ \sin\varphi^{[z]} & \cos\varphi^{[z]} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \bar{d}_x \\ \bar{d}_y \\ \Delta\varphi \end{pmatrix} \tag{23}$$

to be used in lieu of $(z - \hat{z})$ in the FastSLAM implementation. Further details can be found in [18].

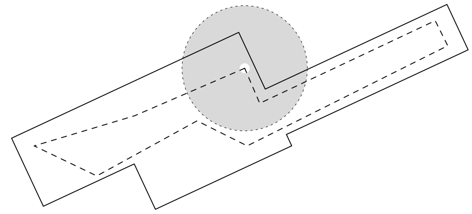
IV. Experiments

A. Update of Region Features

Due to the added complexity of partial measurements, the update of region features needs to be experimentally verified. Fig. 5 shows a single region consisting of three rectangles. In a simulation run, a region estimate is updated using only partial measurements perturbed by Gaussian noise. The squared estimate error is plotted, showing an accurate estimate despite only partial observations.

B. Region-based SLAM

The semantic region-based SLAM algorithm has so far only been tested in a simulation environment. This environment, based on a blueprint of our lab, was manually annotated with region features. Measurements were then simulated with a limited sensor range of 8 m and added Gaussian noise for the region parameters. The used measurement noise covariance matrices were known to the SLAM algorithm, but increased by 20 % for the simulated error. The entire course is about 300 m long. Fig. 6 shows the true robot position compared to the SLAM result, along with part of the resulting region map.



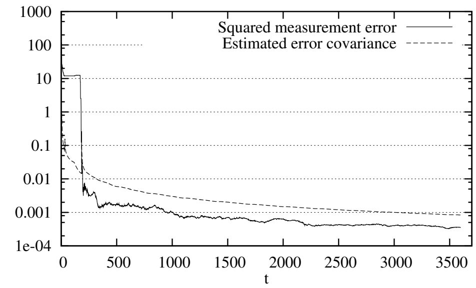


Fig. 5. *Top*: A region (solid line) of 18 m length left-to-right, measured from 3 600 different positions along a path (dashed line), with a sensor range of 5 m (indicated by gray circle). *Bottom*: Development of the squared measurement error and the trace of the estimated error covariance matrix. Around $t = 150$, all edges are closed and the mapped feature is complete.

V. Conclusions and Future Work

A semantic SLAM algorithm based on region features was presented. Given a virtual sensor measuring region features, it is capable of successfully building a region-based map including metric, topological, and semantic information. Geometrical properties of regions can be estimated even when, due to limited sensor range, only partial measurements are available. Simulation-based tests show that region parameter estimation works well and that the region-based FastSLAM algorithm provides good results. The map built in this semantic mapping process provides a basis for semantic navigation and planning. This way, a mission control system can be implemented at an abstract, semantic level, providing a convenient human-computer interface and allowing semantic mission planning independent of the underlying control system.

With a definition for region features in place and a SLAM algorithm using these features written and tested, further research will involve creating virtual sensors using high-level feature extraction methods to obtain region features. Features may be extracted using a variety of sensors such as laser scanners or stereo camera systems. We plan to couple our region-based SLAM algorithm with an underlying standard SLAM approach used to build local metric maps. Regions may be used to find points at which the start of a new local map is triggered. Conversely, region features may be extracted from laser scans aligned using local mapping. We are also investigating the extraction and mapping of elliptical and trapezoidal features. The final goal is to use the semantic mapper proposed here as part of a semantic mapping and navigation framework [20].

References

[1] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proc. 16th Int. Joint Conf. on Artificial Intelligence*, 2003, pp. 1151–1156.

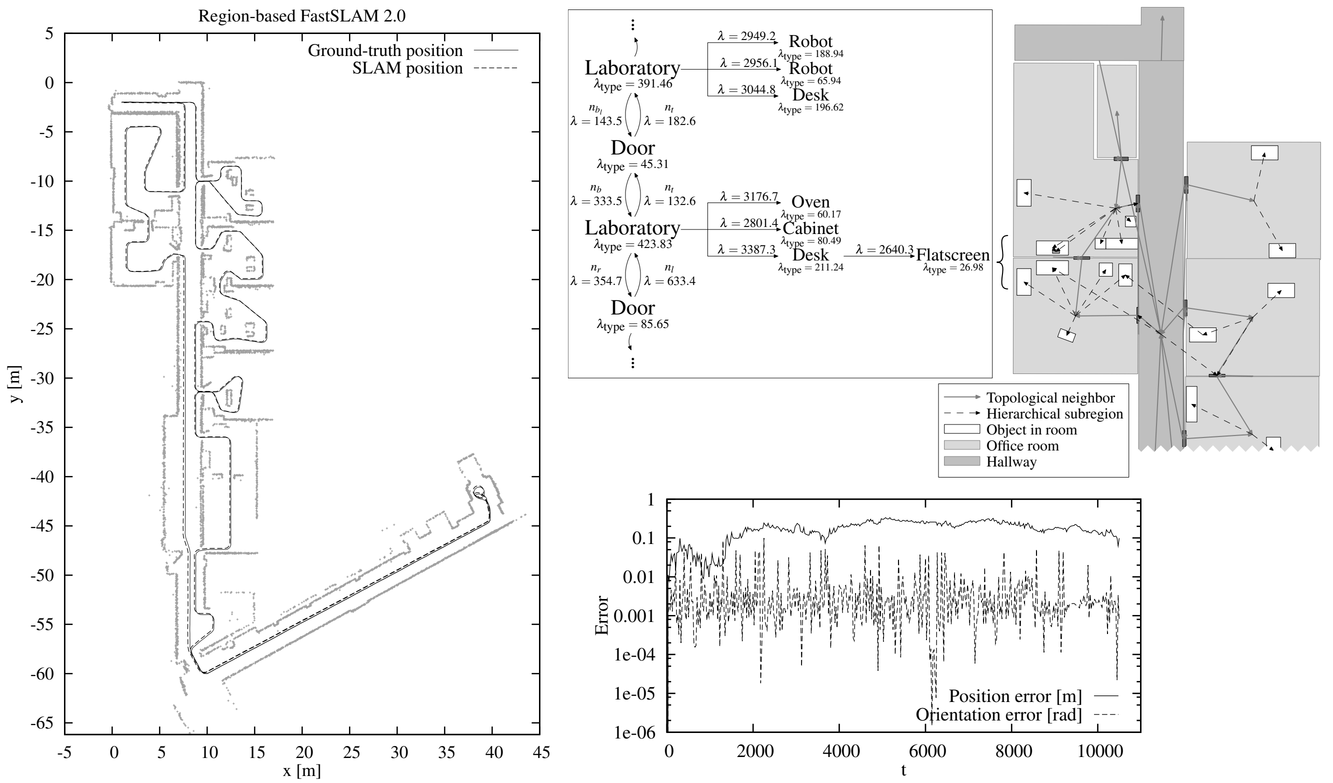


Fig. 6. Region-based FastSLAM test results using four particles. *Left*: The estimated and ground-truth vehicle positions. The given test run accumulates some drift in x direction. *Bottom right*: The position and orientation error over time. The position error stays below 1 m at all times and amounts to about 10 cm at the end of the run. *Top right*: Detail of the created region map. Different semantic types are marked in different shades of gray: hallways are marked darker than office rooms. Dashed arrows indicate a subregion relationship, gray arrows indicate topological adjacency.

[2] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *Int. J. of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.

[3] 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, pp. 229–241, 2001.

[4] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. 18th AAAI Nat. Conf. on Artificial Intelligence*, 2002, pp. 593–598.

[5] J. E. Guivant and E. M. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 242–257, 2001.

[6] T. Bailey, J. Nieto, and E. M. Nebot, "Consistency of the FastSLAM algorithm," in *Proc. 2006 IEEE Int. Conf. on Robotics and Automation*, May 2006, pp. 424–429.

[7] D. Kortenkamp and T. Weymouth, "Topological mapping for mobile robots using a combination of sonar and vision sensing," in *Proc. 12th AAAI Nat. Conf. on Artificial Intelligence*, 1994, pp. 979–984.

[8] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 2, pp. 125–137, 2001.

[9] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller, "An Atlas framework for scalable mapping," in *Proc. 2003 IEEE Int. Conf. on Robotics and Automation*, 2003, pp. 1899–1906.

[10] N. Tomatis, I. Nourbakhsh, and R. Y. Siegwart, "Hybrid simultaneous localization and map building: A natural integration of topological and metric," *Robotics and Autonomous Systems*, vol. 44, no. 1, pp. 3–14, 2003.

[11] B. J. Kuipers, J. Modayil, P. Beeson, M. MacMahon, and F. Savelli, "Local metrical and global topological maps in the hybrid spatial semantic hierarchy," in *Proc. 2004 IEEE Int. Conf. on Robotics and Automation*, 2004, pp. 4845–4851.

[12] B. J. Kuipers, "The cognitive map: Could it have been any other way?" in *Spatial Orientation: Theory, Research, and Application*. Plenum Press, 1983, pp. 345–359.

[13] ——, *An Intellectual History of the Spatial Semantic Hierarchy*, ser. Springer Tracts in Advanced Robotics. Springer Berlin/Heidelberg, 2008, vol. 38, pp. 243–264.

[14] F. Dellaert and D. Bruemmer, "Semantic SLAM for collaborative cognitive workspaces," in *AAAI Fall Symposium Series 2004: From Interfaces to Intelligence*, 2004.

[15] C. Stachniss, Óscar Martínez-Mozos, A. Rottmann, and W. Burgard, "Semantic labeling of places," in *Proc. 12th Int. Symposium of Robotics Research*, 2005.

[16] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J.-A. Fernández-Madrigal, and J. G. Jiménez, "Multi-hierarchical semantic maps for mobile robotics," in *Proc. 2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2005, pp. 2278–2283.

[17] S. Vasudevan, V. Nguyen, and R. Y. Siegwart, "Towards a cognitive probabilistic representation of space for mobile robots," in *Proc. 2006 IEEE Int. Conf. on Information Acquisition*, 2006, pp. 353–359.

[18] J. Oberländer, "Semantisches SLAM mit komplexen metrisch-topologischen Regionsmerkmalen," Diploma thesis, FZI Forschungszentrum Informatik, Karlsruhe, 2007, in German.

[19] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using FastSLAM," in *Proc. 2003 IEEE Int. Conf. on Robotics and Automation*, vol. 2, 2003, pp. 1985–1991.

[20] K. Uhl, M. Ziegenmeyer, B. Gaßmann, J. M. Zöllner, and R. Dillmann, "Entwurf einer semantischen Missionssteuerung für autonome Serviceroboter," in *20. Fachgespräch Autonome Mobile Systeme (AMS 2007)*, Oct. 2007, in German.