A Semantic Approach to Sensor-Independent Vehicle Localization

Jan Oberländer1, Sebastian Klemm 2, Marc Essinger 2, Arne Roennau1, Thomas Schamm 2, J. Marius Zöllner 2, Rüdiger Dillmann 3

Abstract—As intelligent vehicles become more and more capable, they must learn to navigate and localize themselves in a variety of environments, including GPS-denied and only crudely mapped areas. We argue that since autonomous vehicles must be able to perceive, and semantically interpret, their immediate environment, they should be able to use abstract semantic information as their sole means of localization. This simplifies the level of detail and precision required from environment maps so that, for example, a rough floor plan of a parking garage will suffice to autonomously navigate it. We propose a concept for semantic localization which only requires a conceptual semantic map of the environment, and can be made to work with any kind of sensor data from which the required semantic information can be extracted. We present a localization algorithm which may be used as a base for semantic navigation, e.g. in context of automated driving, and some initial results of its application in a parking garage scenario.

I. INTRODUCTION

For autonomous driving, vehicle localization is of great importance. While a GPS-based system can be relied upon in many situations to determine an exact global pose, there are many areas in which alternative means of localization are required. This includes dense inner city areas with severe GPS signal fluctuations, and GPS-denied environments such as parking garages. To provide reference data for all such areas, in formats that can be used by existing metric or topological localization approaches with a wide variety of possible sensors (LIDAR, cameras, PMD sensors, etc.), would be a major logistical challenge, if not entirely infeasible. This implies that an autonomous vehicle must achieve a higher level of scene understanding in order to be able to use much simpler maps.

We consider that a semantic, qualitative approach to localization will be of interest for these challenges, for a variety of reasons:

• Semantic navigation can be considered a biologically inspired approach to navigation: Humans do not navigate using metric coordinates, they rely on a topological view of the environment and take semantic cues into account [1].
• Purely metric navigation puts fairly heavy requirements on an autonomous system: a highly precise, complete, and up-to-date map of the environment must be available. For autonomous cars, a high-precision Differential GPS and IMU are considered necessary in outdoor environments. However, most of this data is not even required, as human drivers prove every day: as long as the current location is roughly known, and the current traffic situation is observed and correctly interpreted, this is sufficient to turn right at the correct light or to safely back into a free parking space. Regardless of the precision and level of detail of available maps, local sensor data must be evaluated regardless in order to stay clear of obstacles.
• Especially with autonomous cars, a semantic understanding of the current traffic situation is necessary anyway: the vehicle must be capable of reasoning about the location of lanes, legal turns, road signage etc.
• Once the vehicle’s sensor observations are segmented and classified into specific objects and their meanings, a navigation system building on this information becomes completely independent of the underlying sensors. Regardless whether planar or 3-D laser scanners, monocular, stereo or PMD cameras or even RADAR or ultrasound sensors are used or combined, the semantic navigation only requires that enough semantic objects are detected with low misclassification rates.
• The maps required of the operating environment are greatly simplified: As long as they contain enough detectable objects, are topologically correct and metrically approximately correct, the location can be determined well enough for the autonomous system to perform its tasks. This can drastically reduce the effort required to install autonomous systems and keep them operating. Fine-grained local control, as required for example when backing into a parking space, must be handled reactively based on live sensor data in any case.
• Lastly, there is of course also a purely academic interest in taking this approach. To date, there are surprisingly few works aiming to construct purely semantic, “qualitative”, navigation methods.

In the following, we outline a concept for performing semantic localization using a Discrete Bayes Filter, which works on abstract semantic measurements we call egocentric semantic phrases. We illustrate the approach using a simple scenario for semantic navigation in a parking garage, with some initial results.
II. RELATED WORK

In robotics, the idea of modeling navigation and localization on the human cognitive map goes back to the work of Benjamin Kuipers’ *Spatial Semantic Hierarchy* [2], [1]. The robot determines for itself salient environment points and topological connections between them based on its own navigational abilities. The result is a topological graph connected by the actions the robot can perform. Various approaches have then been presented which perform topological mapping [3], [4].

In other works, semantic information is used not to navigate, but to improve generated maps. E. g., [5] improve the performance of Iterative Closest Point point cloud registration algorithm by exploiting semantic information such as the location of points on the floor, the ceiling or the surface of objects.

Few works build an explicit link between the conceptual topological map and semantic environment representations using human-made environmental concepts (e. g., corridors, doors, intersections, traffic lights, etc.) for the purpose of navigation. [6] learn to distinguish rooms, corridors and doorways in an office environment from 2-D laser scans using an AdaBoost approach. [7] differentiate between buildings and nature in camera data collected outdoors. [8] define a set of actions for their robot and determine an information gain for each of those actions to perform active localization—the ideal action for the robot is to determine its position with high certainty. [9] present a technique to recognize door signs and perform data association about the semantic meanings of the signs. Semantic landmarks can either be generated manually or be inferred from a robot’s dynamic surroundings: [10] use changes of discrete states’ occupancy information as an indicator for activity and use this information for labelling semantic regions in an urban environment context. [11] use a Partially Observable Markov Decision Process (POMDP) generated out of topological information to track a robot’s position within an office environment.

The approaches listed above tend to use their semantic information only in order to improve their existing localization methods, and could be seen as building blocks for a full-blown semantic localization framework. If enough semantic information can be obtained, it is in principle possible to navigate using a purely semantic environment representation [12].

In our concept, we specify a generic semantic map structure and a method to localize against this map. The map structure contains certain classes of semantic objects specific to the general type of environment and to the vehicle’s, or robot’s, intended purpose of navigation. In contrast to other approaches, we propose a localization method independent of the concrete sensor used, so long as algorithms exist to observe the objects described in the map. By using a Dempster-Shafer belief model for the observed objects similar to [13], the classification uncertainty is modeled explicitly and taken into account in the localization procedure. Our approach uses a Discrete Bayes Filter (DBF) to estimate the unknown probability density function of the robot’s current position on a topological graph of discrete positions. We match the observations to the expected egocentric semantic phrases stored in the graph nodes, purely on an abstract semantic level, using minimal metric information. The localization concept is described in the following.

III. SEMANTIC LOCALIZATION

We propose a novel method for localization that relies on a simple conceptual semantic map with approximate metric relations. The overall architecture is shown in Fig. 1. Localization is performed by matching a semantic interpretation of the vehicle’s sensor observations against the expected semantic observations at a number of discrete semantic locations which are automatically generated from the conceptual map.

A. The Conceptual Semantic Map

The conceptual semantic map is typically generated by a human, which, due to the simplicity of the map can be accomplished easily with a graphical tool. The map contains traversable segments (such as corridors, lanes etc.) which are topologically with a leads-to edge connected at intersections. Objects of interest alongside these segments (e. g., obstacles, parking spaces, doors, signs, . . . ) are linked to the segments using left-of or right-of edges. Further semantic relations could be added if required. An example of such a conceptual map for an application scenario will be presented in Sec. IV.
B. The Semantic Location Graph

As the semantic locations represent a discrete set of possible states, we implement the localization using a Discrete Bayes Filter (DBF) [14]. Each DBF state corresponds to a semantic location and is a vertex in the Semantic Location Graph $\mathcal{S} = (\mathcal{V}_\mathcal{S}, \mathcal{E}_\mathcal{S})$. Two vertices $x_i, x_j \in \mathcal{V}_\mathcal{S}$ are connected by directed edges if $x_j$ is directly reachable from $x_i$, thus defining the topology of the state space. Each edge holds a transition probability; the probabilities can either be uniform across all outgoing edges of a vertex, or incorporate more detailed prior knowledge, e.g. U-turns at the end of a bidirectional lane that is not a dead end could be considered less likely.

Given knowledge about traversable corridors or lanes in the conceptual map, semantic locations can be generated automatically by projecting the positions of salient semantic objects next to the lanes, onto the lanes. The expected semantic observation for each semantic location can then be calculated from the objects surrounding the location.

C. Semantic Observations

We define a semantic observation $\mathbf{Z}_t$ as a radially ordered sequence of $m$ objects with a relative position and results from a semantic classifier:

$$\mathbf{Z}_t = (z_{t,1}, \ldots, z_{t,m}) , \text{ where } z_{t,i} = (x_{t,i}, \Sigma_{z,t,i}, m_{z,i}, \rho_{z,i}) .$$

The egocentric view is thus discretized into $m$ equal-size angular intervals, such that the observation with index $i$ lies in the direction of $i \cdot 2\pi/m$ with a field of view of $2\pi/m$ radians, relative to the vehicle coordinate frame. Each $z_{t,i}$ is a tuple containing the mean pose $x_{t,i}$ and pose covariance $\Sigma_{z,t,i}$ of the object, a Dempster-Shafer belief assignment $m_{z,i}$ for the semantic classification of the object and, optionally, raw sensor data $\rho_{z,i}$ associated with the object. By using a Dempster-Shafer belief assignment [15], [16], any classifier applied to the sensor data can describe the uncertainty, conflict and ignorance over the set of possible semantic object classes. In our approach, the set of classes is defined as

$$\Theta := \{ o_1, \ldots, o_n, \text{other}, \text{none} \}$$

where $o_1, \ldots, o_n$ are specific known object classes, other describes unknown objects, and none is used specifically for angular intervals in which nothing was observed. In typical semantic navigation scenarios, the number of possible classes for each object tends to be quite small. For example, in our parking garage scenario we can limit the object classes to parking spaces (empty or occupied), wall segments, lanes, and support structures.

D. DBF Process Model

The process model in the semantic Discrete Bayes Filter must be given special consideration, since the underlying state space is discrete, while the vehicle motion is continuous. The prediction step of the Discrete Bayes Filter is given by

$$\bar{p}_{k,t} = \sum_i p(X_t = x_k \mid u_t, X_{t-1} = x_i)p_{i,t-1}$$

following the notation given in [14]. To reduce complexity, one can calculate $p(X_t = x_k \mid u_t, X_{t-1} = x_i)$ only for those $x_i$ which are directly linked to $x_k$ in the Semantic Location Graph:

$$\bar{p}_{k,t} = \sum_{i \in (x_i, x_k) \in \mathcal{E}_\mathcal{S}} p(X_t = x_k \mid u_t, X_{t-1} = x_i)p_{i,t-1}$$

In our model, $u_t$ is continuous, represented by a Gaussian estimate of the vehicle’s odometry. Given that the previous state is $x_i$, we assume a uniform prior over its Voronoi region. The prediction step then convolves this prior with the Gaussian $p(u_t)$. The resulting probability for state $x_k$ is then equal to the area of the resulting distribution falling into $x_k$’s Voronoi region. In our implementation we simplify this model to a one-dimensional version, as shown in Fig. 2: We assume that the robot is moving along the straight line between $x_i$ and $x_k$. The Voronoi region of $x_i$ is bounded halfway between these two states, and we further assume it to be symmetric, which is valid as long as the states tend to be fairly evenly spaced. The prior probability density function at $x_i$ is then a box function

$$\Pi(x,w) = \begin{cases} 1/\ell_i, & -\ell_i/2 \leq x < \ell_i/2, \\ 0, & \text{otherwise,} \end{cases}$$

where $\ell_i$ is set to the distance between the states, i.e., $\ell_i := ||x_k - x_i||$. The Gaussian distribution obtained from vehicle odometry represents only the mean $\mu_i$ and covariance $\sigma^2_i$ of the distance driven (ignoring the change in orientation). The resulting posterior probability is then given by

$$p(X_t = x_k \mid u_t, X_{t-1} = x_i) = \int_{||x_k-x_i||+\ell_i/2}^{||x_k-x_i||-\ell_i/2} \Pi(\ell_i) \cdot \mathcal{N}(\mu_i, \sigma^2_i) \, dx .$$

If the filter update frequency is high enough to perform several updates while the vehicle is moving from one location to an adjacent one, this model is an adequate approximation. If several edges might be traversed between two sensor update steps, the update horizon can be extended recursively to include indirect predecessors. Algorithm 1 shows how: all possible paths of length up to $p$ are traversed. The prediction from (7) is calculated in line 2. Depending on the recursion depth $p$, the algorithm includes paths from all predecessor states in the sum (calculated in the loop in lines 5–11).

E. DBF Sensor Model

After calculating the prediction $\bar{p}_{k,t}$ using (4) or the variants thereof discussed in the previous Section, the measurement update is given by

$$p_{k,t} = \eta p(\mathbf{Z}_t \mid X_t = x_k)\bar{p}_{k,t} ,$$

again following the notation of [14], with $\eta$ being a normalization factor. We determine the conditional probability of the semantic observation by calculating an expected
Fig. 2. Prediction step for the Discrete Bayes Filter, from a predecessor \( x_i \) of the target state \( x_t \). Given the current state \( X_{t-1} = x_i \), we assume a bounded uniform prior distribution (1), delimited by the edges of the approximate Voronoi region around \( x_i \). We convolve the prior with a Gaussian obtained from the vehicle motion \( u_t \) (2) to obtain the posterior distribution (3). The area under the curve coinciding with \( x_t \)’s Voronoi region is the posterior probability (4).

**Algorithm 1 DBF-Predict**

**Input:**
- \( x_i \rightarrow x_k \) - a given path from state \( x_i \) to state \( x_k \), with length \( \|\psi\| \) and probability \( p(\psi) \)
- \( \mu \) - mean of vehicle motion
- \( \sigma^2 \) - variance of vehicle motion
- \( \ell_i \) - size estimate of Voronoi region at \( x_i \)
- \( \ell_k \) - size estimate of Voronoi region at \( x_k \)
- \( p \) - recursive depth for predecessor vertices of \( x_t \) to take into account

**Output:**
- \( p' \) - The probability of reaching \( x_k \) from \( x_i \) or any of \( x_t \)'s predecessors, up to \( p \) levels of recursion, via the path \( \psi \).

1. Calculate this path’s probability:
   \[ p' \leftarrow p_{1-1} \cdot p(\psi) \cdot \| \psi \|^{-1/2} \cdot \prod_{t} \left( \frac{1}{\mu} \right) \cdot \left( \frac{\sigma^2}{\pi} \right) \]

2. Recurse for paths from predecessor states:
   \[ \text{for all } j: (x_j \rightarrow x_i) \in E, \text{ do} \]
   \[ p'' \leftarrow p(\psi) \cdot p(x_j \rightarrow x_i) \]
   \[ \text{extended path probability} \]
   \[ p'' \leftarrow p'' + \text{DBF-Predict}(x_j \rightarrow \psi' \cdot x_k; \mu, \sigma^2, \ell_j, \ell_k, p - 1) \]
   \[ \text{end for} \]

3. if \( p > 0 \) then

4. Recurse for paths from predecessor states:

5. for all \( j: (x_j \rightarrow x_i) \in E, \text{ do} \)

6. \[ \| \psi' \| \leftarrow \| \psi' \| + \| x_j \rightarrow x_i \| \]
   \[ \text{extended path length} \]

7. \[ p'' \leftarrow p'' + \text{DBF-Predict}(x_j \rightarrow \psi' \cdot x_k; \mu, \sigma^2, \ell_j, \ell_k, p - 1) \]

8. if \( p > 0 \) then

9. return \( p \)

end if

Measurement \( \hat{Z}_k \) for the semantic location \( x_k \) in the form shown in (1), where the belief assignment assigns all mass to the expected object type\(^1\).

We then expect this observed location to the actual observation \( Z_t \) as follows. Both measurements are given as *semantic phrases* of equal length \( m \) following Eq. (1). We align the phrases using a variant of the optimal sequence alignment algorithm presented in [17]. Two effects must be considered in this alignment step: the semantic phrases are not rotationally invariant (thus requiring radial alignment), and radial orderings may differ slightly due to changes in perspective. To handle radial alignment, we compare \( Z_t \) to a duplicated version of the expected observation, \( \langle \hat{z}_{k,1}, ..., \hat{z}_{k,m}, \hat{z}_{k+1,1}, ..., \hat{z}_{k+1,m-1} \rangle \). An example cost matrix is shown in Fig. 3. We define the following cost functions to fill the cost matrix (illustrated in Fig. 4):

- **Deletion** represents an expected object that does not correspond to any measured object. This is only allowed in the first and last row of the cost matrix, to test for different rotational alignments. The cost for deleting \( \hat{z}_{k,j} \) after processing \( z_{t,j} \) is given by

\[ c_{\text{del}}(i,j) = \begin{cases} 0, & j = 0 \lor j = m, \\ \infty, & \text{otherwise.} \end{cases} \] (9)

- **Insertion** represents a measured object that does not correspond to any expected object. We do not allow any insertions since both phrases are by definition of the same length:

\[ c_{\text{ins}}(i,j) = \infty. \] (10)

- **Substitution** represents a direct correspondence between an expected and a measured object. We can directly measure how well this correspondence fits, by combining their basic belief assignments and using the resulting conflict between the classifications of \( \hat{z}_{k,j} \) and \( z_{t,j} \) in the substitution cost. Additionally, the difference between the expected and actual metric object distance \( \delta(\hat{z}_{k,i}, z_{t,j}) \) influences the cost:

\[ c_{\text{sub}}(i,j) = 1 - e^{-\frac{\delta(\hat{z}_{k,i}, z_{t,j})^2}{2}} \cdot \left( 1 - \sum_{A,B \in \Theta} \hat{m}_{t,j}(A) m_{t,j}(B) \right). \] (11)

- **Transposed substitution** represents a correspondence between two expected and two measured objects, where the expected objects’ radial ordering has been transposed. This incurs a fixed transposition cost \( c_x \) plus the respective substitution costs:

\[ c_{\text{xsub}}(i,j) = c_x + c_{\text{sub}}(i-1,j) + c_{\text{sub}}(i,j-1). \] (12)
Along with the optimal alignment, the cost matrix also provides the optimal alignment cost $c_{opt}(\hat{Z}_k, Z_t)$ as the sum of costs along the optimal alignment path. If we calculate the optimal alignments with respect to all expected measurement phrases $\hat{Z}_k$, we can expect a lower alignment cost for those semantic locations near the true location. Therefore, we calculate the probability (8) simply as

$$p(Z_t | X_t = x_k) = 1 - \frac{c_{opt}(\hat{Z}_k, Z_t)}{c_{max}},$$

where

$$c_{max} = m$$

is the maximum (worst-case) alignment cost incurred by $m$ direct substitutions at full conflict.

Given the optimal alignments of $\phi_j$ with respect to all expected measurement phrases $\hat{\phi}_i$ and the associated minimum costs, $c(\phi_j, \hat{\phi}_i)$ the most likely semantic location could simply be determined as

$$\hat{\psi}_t = \arg\max_j c(\phi_j, \hat{\phi}_i).$$

IV. APPLICATION:
SEMANTIC NAVIGATION IN PARKING GARAGES

Our application scenario targets autonomous navigation inside parking garages. The goal is a system which only requires a comparatively crude map of the parking garage environment, yet suffices to, for example, find a specific free parking space and navigate back out of the garage without a high-precision on-board localization system.

Our test environment is the parking garage in the basement of our institution, a single-storey installation with 132 parking spaces. For reference purposes, a map (see Fig. 6, top) was built using a graph-based 2-D SLAM algorithm developed at our institute and manually annotated. Our localization algorithm, on the other hand, uses only a crude map sketch created manually by walking through the parking garage with pen and paper. This map sketch is represented as a graph whose vertices represent different environment entities such as parking spaces, obstacles such as columns or walls, and the general lane layout. The graph edges represent simple relations between the entities such as left-of or right-of (between lane segments and objects next to them, or between parallel lanes), and leads-to (establishing topological connections between lane segments). A new lane segment starts at every lane intersection, and whenever a new group of parking spaces starts next to the lane. A simple example is shown in Fig. 6 (bottom).

To convert this map into a Semantic Location Graph, we...
iterate over all lane segments and, for each segment, over the objects marked as left-of or right-of the segment. The points of origin of each object’s coordinate system are projected onto the lane segment. Semantic locations are established at the location of the projected point. If several semantic locations fall close to each other (up to a certain distance threshold), they are merged into one location. If longer stretches of a lane segment (over a certain length threshold) do not contain any semantic locations, additional locations are inserted.

For each semantic location $\psi_j$, an expected observation $\hat{\theta}_j$ is calculated incorporating all objects in the map sketch within a certain radius of the location. The locations are then connected to form a directed graph. Edges are inserted between consecutive locations along a lane segment. For lane segments connected by a leads-to edge in the map sketch, edges are inserted between the last semantic location along the source lane segment and the first semantic location along the target lane segment. Transition probabilities are always distributed uniformly among the outgoing edges of a semantic location, although more sophisticated models could be implemented as well.

V. INITIAL RESULTS

An illustrative example of the localization scheme uses a sensor dataset recorded with our research vehicle, CoCar (see Fig. 7), within our institution’s underground parking garage. For this dataset we use only 2-D point cloud data extracted from CoCar’s three IBEO laserscanners, a simple object classifier which recognizes obstacles, walls and occupied parking spaces, and a minimalistic motion model accounting only for the vehicle’s forward motion. Figure 8 illustrates the vehicle’s localized pose while moving along a corridor, passing by multiple parking spots and columns, driving onto an intersection and turning right into a different corridor. Since

2Typically the object coordinate systems lie at their geometric center.

3For most objects, this chooses the closest point on the lane segment; for parking spaces, the projected point lies at the intersection of the lane segment with the parking space’s main axis and thus represents a good approximation of the point at which to turn into the parking space.

we do not use any information about the odometry except the accumulated driven length, the intersection introduces a large amount of ambiguity as the path taken is unknown. However, we can observe that the sequence of the semantic states the car passes through matches the actually driven trajectory when the sensor data was recorded – the car turns right at the intersection. The used sensor model is capable of supplying enough information gain to infer which direction the car took when crossing intersections. This example merely illustrates the capabilities of the approach; further experiments are currently being performed.

VI. CONCLUSIONS AND OUTLOOK

In this paper we presented a semantic localization approach that is solely based on limited vehicle odometry and the observed semantic view of the vehicle’s surroundings. We have illustrated how this approach can be used to perform qualitative, rather than precise metric, localization in a parking garage, with simple means, provided a semantic classifier is available for situation interpretation. Although our preliminary object classification method does not provide high certainty when assigning masses to the different belief classes in our Dempster-Shafer model, the sensor model nevertheless has the potential to compensate and make use of the contained information and match the observation onto the semantic map. The semantic objects observed in our example scenario could equally be extracted using, e.g., cameras instead of laserscanners. The simple, abstract nature of our map means that the logistic effort required to, for example, provide autonomous vehicles with usable maps of all parking garages of a city, can be reduced significantly.

Our current work focuses on the detailed evaluation of the concept, and on the improvement of our simple classifier. By

![Fig. 7. CoCar – the Cognitive Car [18] at Research Centre for Information Technology used for research in cognitive assistive systems and automated driving [19].](image1)

![Fig. 8. A sequence of semantic poses at which the vehicle is localized. At the intersection shown in the lower right corner, any path could be taken; evaluating the sensor model however finds the correct path taken.](image2)
building small local maps (e.g. with the approach presented in [20]), we can collect more local information to improve classification, which will aid localization performance.

REFERENCES