accepted for publication at
International IEEE Conference on Intelligent Transportation Systems, 2011

GIS-Based Topological Robot Localization through LIDAR Crossroad Detection

Andre Mueller, Michael Himmelsbach, Thorsten Luettel, Felix v. Hundelshausen and Hans-Joachim Wuensche

***Abstract*— While navigating in areas with weak or erroneous GPS signals such as forests or urban canyons, correct map localization is impeded by means of contradicting position hypotheses. Thus, instead of just utilizing GPS positions improved by the robot's ego-motion, this paper's approach tries to incorporate crossroad measurements given by the robots perception system and topological informations associated to crossroads within a pre-defined road network into the localization process. We thus propose a new algorithm for crossroad detection in LIDAR data, that examines the free space between obstacles in an occupancy grid in combination with a Kalman filter for data association and tracking. Hence rather than correcting a robot's position by just incorporating the robot's ego-motion in the absence of GPS signals, our method aims at data association and correspondence finding by means of detected real world structures and their counterparts in predefined, maybe even handcrafted, digital maps.**

I. INTRODUCTION

As stated in previous work [1], [2], [3], localizing a robot within its environment is considered to be a key prerequisite for autonomous robot navigation. Hence simultaneous localization and mapping (SLAM), which addresses the problem of building an environment map by associating sensor measurements to landmarks and localizing the robot within this environment, became quite popular. Localizing a robot within an unknown environment involves the correct mapping of the environment, by considering the noise prone ego-motion- and sensor measurements as well as finding correspondences between sensor measurements and already registered landmarks. Instead of generating a map from sensor measurements, this paper aims at robot localization by identifying comparable features in both LIDAR measurements and human readable (maybe even handcrafted) digital maps. By stemming on an already mapped environment, the approach's challenge demands the association of structures within LIDAR range measurements to features within the digital map. Especially the one to many association between detected structures within the sensor data and their correspondences within the map turns out to be demanding within our problem formulation.
By performing a topological analysis of the given map, comparable features to real world structures are identified. It is decomposed in crossroad acquisition by means of examining the map's road intersections, extracting their topology (number of branches and their angles) and evaluating the distance between adjacent crossroads. Second, crossroad hypotheses within the LIDAR range measurements are identified, that enable a meaningful comparison. Our method is based on a free-space evaluation between obstacles within an occupancy grid. The evaluation's result is a set of crossroad hypotheses composed of a center point and branches fanning out towards the intersecting lanes. In that fashion we get topological crossroad descriptions from LIDAR range measurements that match the map's extracted topological information and thus builds a means for robot localization.
Our approach aims at robot position estimation in areas that typically prevent correct GPS-localization due to error prone or even absent signals, e.g. while driving through a forest with continuously wrong or missing GPS measurements, or navigating in areas with huge obstacles, disturbing or reflecting GPS signals.

All authors are with the Institute for Autonomous Systems Technology (TAS), University of the Bundeswehr Munich, 85577 Neubiberg, Germany. Contact author email: a.mueller@unibw.de

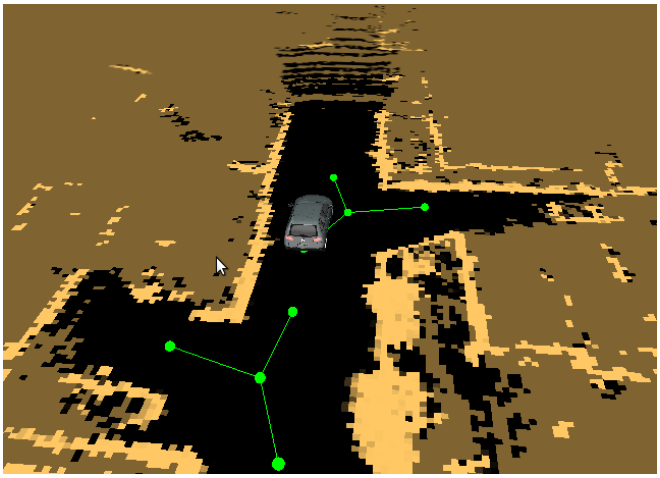


Fig. 1. A pane of the occupancy grid, provided by LIDAR range measurements, with our robot MuCAR-3 in it's center and two valid crossroads hypotheses (*green structures*). *All figures are best viewed in color.*

A. Related Work

Following the idea of mapping topological features, *Morris et al.* [4] built topological maps of abandoned mines by exploring mine corridors, their intersections and the distance between adjacent ones. Since external communication (such as GPS) was impossible, the robot's navigation and localization strongly depended on a correctly built map. Similar to our work, nodes represent the intersections (crossroads) and edges the paths between them. Intersections are extracted by triangulating a range scan according to Delaunay and finding triangles with all three sides longer than a given distance parameter. By matching hypotheses over multiple

accepted for publication at
International IEEE Conference on Intelligent Transportation Systems, 2011

range scans a robust intersection detection was established. Based on the intersections and the distance and direction between them a topological map was built up.

A more biologically inspired approach was presented by *Vasudevan et al.*[5]. Its topological information did not rely on some abstract sensor data, but on spatial relations between detected objects. Objects like doors, cups, tables, etc. are utilized to distinguish between different places. While moving within its environment, the robot examines detected objects for known features and relations. Detected doors cause the insertion of a new node into the topological graph, but only if the room associated to this door is unknown in its object relations. Otherwise, gained object information is used for rough self localization within the topological graph, instead of determining the exact position within a room. In that fashion a topological map was built up, that did not contain distances and directions between its nodes. But nodes were named after the place they represent (like kitchen, dining room, etc) while edges define the possible transitions between these places.

This paper's approach shall combine portions off the aforementioned work to overcome some of their disadvantages. In the fashion of *Vasudevan et al.*[5] we also try to find objects in the sensor data for topological mapping. Unlike *Morris et al.* [4] intersections are not represented by means of a simple center point, but as a center point with branches that define possible driving directions. This enables robot localization even in case of missed crossroads, since the topology of each crossroad is incorporated into the localization process in addition to the distances and directions between them. Furthermore to localize the "place" we are in, we do not have to learn objects beforehand like in *Vasudevan et al.*[5], but rather match the object descriptions given by the map to the object descriptions provided by the perception system. Our approach even enables a correct re-localization relative to a given object, which was not covered in the work presented so far, e.g. 5 meters behind crossroad X.

The paper is composed as follows. In section II a short overview of the algorithms processing the raw LIDAR data is given, followed by a detailed explanation of the crossroad detection mechanisms in section III. Mapping the detected crossroads to the given topological map is discussed in section IV, closing in section V with a short analysis of the achieved results and an outlook to future extensions and challenges.

II. PERCEPTION

Equal to [6] we use a $2\frac{1}{2}$-D ego-centered occupancy grid of dimension $200\text{m}\times200\text{m}$, each cell covering a small ground patch of $0.25\text{m}\times0.25\text{m}$. Each cell stores a single value expressing the degree of how occupied that cell is by an obstacle given the current scan only. For calculating the occupancy values, we first inertially correct the LIDAR scan, taking the vehicle's motion into account (exploiting IMU and information from odometry). This is done by simultaneously moving the coordinate system of the vehicle while transforming the local LIDAR measurements to global 3D space. After a frame is completed, all points are transformed back into the last local coordinate system of the vehicle, simulating a scan as if all measurements were taken at a single point of time instead of the 0.1s time period of one LIDAR revolution.

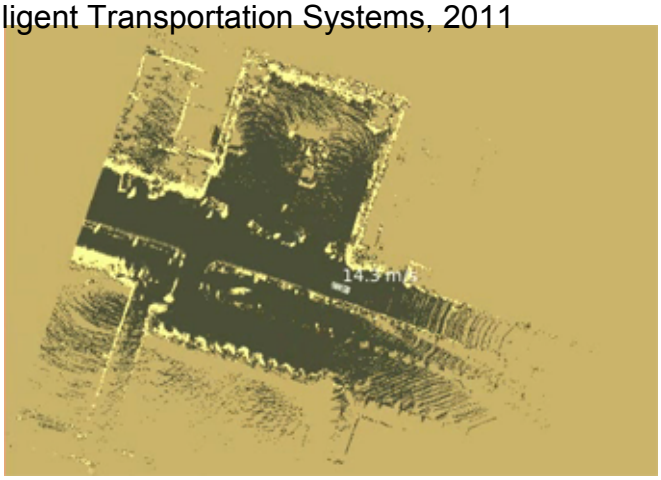


Fig. 2. MuCAR-3 freely moving over the occupancy grid and accumulating obstacles. Black indicates low obstacle probability, light yellow indicating high obstacle probability. Note the large ochre area not hit by any beam, thus having obstacle probability of 0.5.

Similar to Thrun *et. al.* [7], each cell's occupancy value c_{occ} is then calculated to be the maximum absolute difference in z-coordinates of all points falling into the respective grid cell, $c_{occ} = |c_{z_{max}} - c_{z_{min}}|$.

A. Accumulation of Obstacles

To accumulate sensor information from multiple scans in the occupancy grid, we simply let the vehicle freely move over the grid while keeping the grid at a fixed position. Unfortunately, it turns out that keeping records of extreme z-coordinate measurements over multiple scans introduces many spurious obstacle cells, i.e. ones with large absolute z-coordinate differences, into the grid. The reason is that accumulating raw 3D point cloud data requires very high precision in ego-motion estimation which the inertial navigation system of the robot can not provide, especially in harsh environments causing violent sensor pitch motions. Resorting to ICP-style algorithms [8] is also not a viable solution due to the additional computational load involved.

A solution to this problem is to probabilistically accumulate the obstacles detected at a cell instead of the raw point measurements. We thus introduce an occupancy probability $c_{P_{occ}}$ with $0 \leq c_{P_{occ}} \leq 1$, where $c_{P_{occ}} = 0$ denotes a free space cell and $c_{P_{occ}} = 1$ a cell being occupied. Whenever we measure an obstacle at a cell, we increment its probability of being occupied, while in contrary decrementing $c_{P_{occ}}$ if the measurements do not indicate an obstacle. If the measurements do not tell us anything about there being an obstacle at the cell or not, we keep the occupancy probability as it is. Hence, apart from the occupancy of a cell based on multiple scans, we can now tell the probability of a cell being occupied. Fig.2 shows a snapshot of the vehicle moving over the occupancy grid while accumulating obstacles.

accepted for publication at
International IEEE Conference on Intelligent Transportation Systems, 2011

III. CROSSROAD DETECTION AND TRACKING

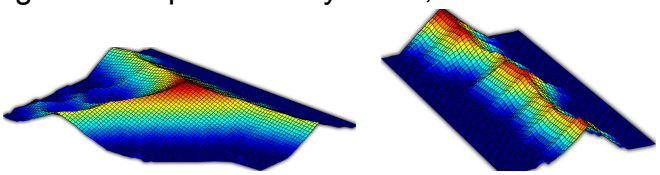
Even before the introduction of GPS, humanity was able to determine the current position on a (maybe even handcrafted) map by exploring the environment and matching detected objects to features within a map. Following this principle, we thought of matching intuitive real world objects to data provided by a global map, since they depict severe decisions regarding the steering direction: crossroads.

A. CROSSROAD DETECTION

By utilizing the accumulated grid's data we perform an evaluation of the obstacle-free area, with obstacles defined by an obstacle probability $c_{P_{occ}} \geq 0.5$ for the following two reasons. First, negative obstacles are considered. As discussed in section II one cell's occupancy probability is determined by examining the probabilities of being free or occupied, resp. However, negative obstacles, like ditches or slopes, are not hit by any laser beam, information about their occupancy state can not be gained. But instead of explicitly detecting negative obstacles, the phenomenon of not knowing a cell's occupancy state is utilized. By definition, occupancy probability $c_{P_{occ}} = 0.5$ is assigned to each cell not covered by any laser beam, which involves grid cells in the far distance as well as cells that lie within a negative obstacle region. I.e. each cell with probability $c_{P_{occ}} = 0.5$ is declared to be an obstacle, which covers negative obstacles as well. Especially in terrain characterized by negative obstacles, such as off road pathways with ditches to the left and to the right this obstacle definition gets useful. Second, grid cells that do not offer information regarding their occupancy state (especially cells far from the center) are declared to be occupied, which reduces the computational demands with respect to the free-space evaluation. As shown in Fig. 2, the ochre area is declared to be an obstacle whereby the free space evaluation is reduced to a smaller patch close to the vehicle, which has proved to be sufficient for crossroad detection.

According to each cell's occupancy state the grid is now binarized to enable the application of the Distance Transform, as proposed by *Borgefors* [9]. We thus get the approximate distance from every free-space cell to the closest obstacle cell. Since the distance transform's outcome builds the foundation of our crossroad detection algorithm and is thus subject to a manifold of operations, we store it in a grid like structure which is further referred to as distance grid G with each grid cell's value $v(c)$, $c \in G$ representing the distance to the closest obstacle.

1) Center Extraction: To determine a crossroad's center alongside it's branches our method exploits the distance grid's property of monotonically increasing values towards the cell with the largest distance to all nearby obstacles. Interestingly, crossroads show salient characteristics in this data structure as shown in Fig. 3, with (a) representing a crossroad's and (b) a road's distance distribution, each cornered by obstacles. The occurrence of a crossroad forces the appearance of one salient maxima within its center, while roads are characterized by an approximately uniform



(a) (b)

Fig. 3. The heights of two samples of the distance grid. Left: the sample of a crossing with a salient maximum in its center. Right: the sample of a straight road with its multiple maxima distributed equally along the center of the road.

distribution. To extract these maxima we introduce $P(Occ \mid \Omega_{D_{c_i}})$, the probability of a grid cell $c_i \in G$ being surrounded by grid cells that share the same distance towards an obstacle. For each obstacle distance v within G we first build up the set:

$$\Omega_{V=v} = \{c_i : v(c_i) = v\} \tag{1}$$

,with c_i denoting the grid's i-th cell. For each element within a set we then determine distance $d_{.,.}$ to all other cells within the same set by:

$$d_{c_i,c_j} = \max\{\mid g_{x_i} - g_{x_j} \mid, \mid g_{y_i} - g_{y_j} \mid\} \tag{2}$$

with (g_{x_i}, g_{y_i}) being the i-th and (g_{x_j}, g_{y_j}) the j-th cell's grid coordinates and $|\cdot|$ the absolute value, to yield the set:

$$\Omega_{D_{c_i}} = \{d_{c_i,c_j} : c_i, c_j \in \Omega_{V=v}, 1 \leq j \leq |\Omega_{V=v}|, j \neq i\}. \tag{3}$$

Each distance in $\Omega_{D_{c_i}}$ then serves as evidence of how unique cell c_i is with respect to it's location. However, the same distance may occur multiple times within $\Omega_{D_{c_i}}$, e.g. if three cells with equal distances share the same value $v(c)$. To incorporate multiple appearances into the computation of $P(Occ \mid \Omega_{D_{c_i}})$, we first define

$$P(Occ \mid D = d) = \frac{1}{((2d+1)^2 - (2d-1)^2)} = \frac{1}{8d} \tag{4}$$

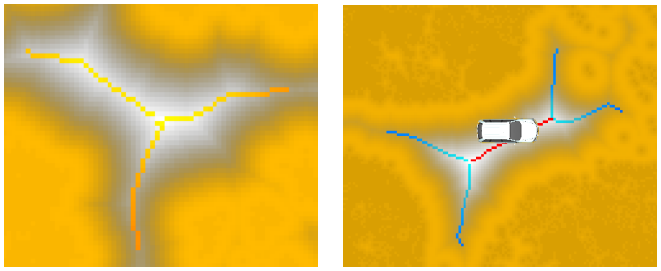
,which represents the probability that one cell is occupied at a distance d. Finally we define $P(Occ \mid \Omega_{D_{c_i}})$ to be the probability that a grid cell in the vicinity of c_i is occupied by the same value:

$$P(Occ \mid \Omega_{D_{c_i}}) = \frac{1}{d_m} \sum_{j=1}^{|\Omega_{D_{c_i}}|} P(Occ \mid D = d_j), d_j \in \Omega_{D_{c_i}} \tag{5}$$

,with d_m representing the maximal achievable distance, in our case the occupancy grid's halfway width.

By applying equation 5 to all the distance grid's cells we get a local similarity distribution over all cells. We now use this distribution to identify grid cells that are very likely to be unique with respect to their location. I.e. cells having a low similarity value are considered to be crossroad center points.

accepted for publication at
International IEEE Conference on Intelligent Transportation Systems, 2011



(a) (b)

Fig. 4. Results of successful branch extractions within the grid structure. While (a) shows a processed 3-way road junction, (b) presents the algorithms result with regard to a 4-way crossroad. The 4-way junction's characteristic is recognizable with a center line (*red colored cells*) instead of a single center point.

2) Branch Extraction: Extracted crossroad center hypotheses are now utilized as a seed for branch extraction within an expansion algorithm. The method explores the distance grid, starting at detected center hypotheses, along the level sets of maxima in a forward value iteration-style algorithm. I.e. it tries to find multiple paths with length d on the distance transform, while certain paths may collapse in the process. For each path the procedure tries to choose action $u_i \in U(c_i)$ which minimizes:

$$C_d^*(c_i) = \min_{u_1,\ldots,u_{d-1}} \left\{ l_I(c_i) + \sum_{j=1}^{d-1} l(c_j, u_j) \right\} \quad (6)$$

,with $l_I(c_i) = 0$ for all center cells, $l_I(c_i) = \infty$ for any other cell and $l(x_j, u_j)$ defined by:

$$l(c_j, u_j) = 1 + \delta v(c_j, u_j) + d(c_j, u_j) + m(c_j, u_j). \quad (7)$$

In this connection action u describes a move towards a cell's 1-neighborhood, i.e. to one of a cell's edge neighbors. With $\delta v(c_j, u_j)$, the difference between the two cell values $v(c_j)$ and $v((c_j, u_j))$ is expressed, which forces the expansion towards the neighboring cell that shares a similar value and thus along a level set. The second term $d(c_j, u_j)$ is able to take two values either $d(c_j, u_j) = 0$ if the distance, as defined in equation 3, to the center cell is kept or increases, or $d(c_j, u_j) = \infty$ otherwise, which causes the expansion away from the center. With $m(c_j, u_j)$ the difference between $v((c_j, u_j))$ and the cell having the largest value in the 1-neighborhood of c_j is expressed. It aims to ascent towards a level set by preferring cells with large values. The path expansion procedure stops if either distance d is reached or only two paths remain throughout the expansion process. Latter even causes the algorithm to reject cell c_i as crossroad hypotheses, since two resulting paths indicate the occurrence of a road instead of a junction. Distance d is chosen to be $d = v(c_i) * s$ with s weighting the value of a crossroad's center point and thus expresses how deep a crossroad shall be explored depending on the crossroads dimension.

While a path is extended towards a new cell, the respective cell may already be a part of another path, i.e. two paths may expand into the same direction. To omit multiple hypotheses directing towards the same branch, the occurrence of two paths occupying the same cell, forces the path with the higher cost to terminate it's expansion. In short: The expansion procedure forces the paths to run along the level sets of maxima in the distance transform, while paths having low costs are chosen over expensive paths if they meet at a certain grid cell. A successful 3-way crossroad extraction is shown in 4(a).

While applying the described procedures to four-way junctions we learned that the characteristics of the center point change. Instead of a single center point a center line, comprising a set of similar valued cells, occurs as shown in 4(b). Interestingly this line's two terminal points still satisfy the prerequisite's to be crossroad center points. Hence, after all the grid's crossroads were extracted a further process revises the grid cells associated to branches of different crossroads, while overlapping branches cause the crossroads to merge. The merged crossroads center is defined by the center line's mean cell.

The expansion algorithm's result, a set of paths $\{p_1, p_2, \ldots, p_n\}$, with n being the number of extracted branches, each containing an ordered set of m grid cells $p_i = \{c_1, c_2, \ldots, c_m\}$, is then evaluated to yield Ω_B, a set of angles expressing the different branch directions with regard to one crossroad. To compute Ω_B we switch back from grid coordinates (g_x, g_y) to the cartesian x, y by determining the position of each cell c_i in the robot's coordinate frame with $(x_i, y_i) = f(g_{x_i}, g_{y_i})$. For each path p_i, with $1 \leq i \leq n$, we then determine mean angle:

$$\bar{b}_i = \frac{1}{m} \sum_{j=1}^{m} atan2(y_j - y_c, x_j - x_c) \quad (8)$$

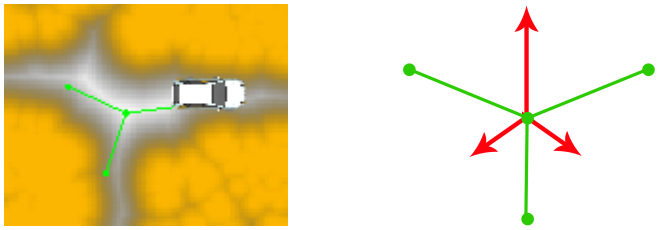
,utilizing the vector between crossroad center cell coordinate (x_c, y_c) and the j-th cell's cartesian coordinate (x_j, y_j), to yield $\Omega_B = \{\bar{b}_1, \ldots, \bar{b}_n\}$ with $\bar{b}_1 \leq \bar{b}_2 \leq \cdots \leq \bar{b}_n$. Additionally each branch's standard deviation s_i is determined, see equation 9, to yield $\Omega_S = \{s_1, \ldots, s_n\}$, which gets especially meaningful in the context of crossroad comparison.

$$s_i = \frac{1}{m} \sum_{j=1}^{m} \left(atan2(y_j - y_c, x_j - x_c) - \bar{b}_i \right)^2 \quad (9)$$

Comprising crossroad center point $p_c = (x_c, y_c)$, the set of branches Ω_B and the set of standard deviations Ω_S our branch description is composed by the tuple: $C = (p_c, \Omega_B, \Omega_S)$

3) Topology Optimization for 3-way-junctions: Fig. 5(a) shows the result of a 3-way crossroad extraction. Unfortunately, the detected center point does not represent the true road intersection point, which causes discrepancies within the topology extraction process, i.e. the extracted set Ω_B does not represent the true topology. E.g. in the example

accepted for publication at
International IEEE Conference on Intelligent Transportation Systems, 2011



(a) (b)

Fig. 5. Within (a) an extracted crossroad topology is shown. The crossroad's center (*green dot in the crossroad's center*) resides on the distance grid's largest value with respect to the location, while its branches (*green lines terminating at a specified distance, denoted by green dots*) fan out towards the level set of maxima. (b) presents the forces (*red arrows*) that are used to optimize the shape of a 3-way road junction.

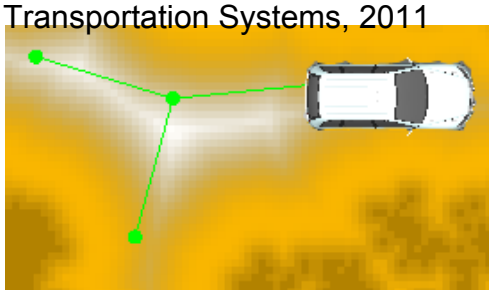


Fig. 6. presents the corrected topology from figure 5(a). Notably the crossroad's center point shifted slightly away from the distance grid's largest value to enable a more realistic crossroad topology.

of figure 5(a) the T-junction's topology resembles more a Y than a T. To get a more accurate crossroad topology a Levenberg-Marquardt optimizer tries to minimize:

$$g(p_c, \epsilon) = \left(\tilde{d} - v\left(f^{-1}(p_c + \epsilon\overrightarrow{a_1} + \epsilon\overrightarrow{a_2} + \epsilon\overrightarrow{a_3})\right)\right)^2 \quad (10)$$

,with $f^{-1}(\cdot)$ transforming cartesian coordinates into grid coordinates, $v(\cdot)$ providing the value at a given grid coordinate, and $\tilde{d}$ being the median of a crossroad's set of grid cell values. The set is built up by incorporating all the grid cells that define the branches of a crossroad. Vectors $\overrightarrow{a_1}, \overrightarrow{a_2}, \overrightarrow{a_3}$ represent the angle bisectors of neighboring crossroad branches with their magnitude equal to the angle difference as shown in Fig. 5(b). By applying a gradient descent method, the crossroad's center is now translated in a way that minimizes equation 10. While the center point is moved, one sample point for each branch is retained that represents the entry of this respective branch as shown in Fig. 6(a).

B. CROSSROAD RECOGNITION AND TRACKING

To be declared valid, a crossroad needs to be detected in at least two successive measurements. Hence, to associate crossroad measurements over several time steps, all detected hypothesis are moved according to the vehicles ego-motion in the robot's coordinate frame, utilizing an Extended Kalman Filter. State $x_{cr} = (x_c, y_c, \Psi_c)^T$ is defined by the crossroad's center position (x_c, y_c) in the robot's coordinate frame and yaw-angle Ψ_c, which defines the crossroad's rotation relative to the robot. Crossroads are moved within the Kalman Filter's prediction step according to the robot's ego-motion in a constant velocity model [10]. System noise covariance Q_k and measurement noise covariance R_k are built up and optimized by filter tuning.

To enable the comparison between generated hypotheses and current measurements, crossroad descriptions are evaluated by means of position and topology.

1) MEASURE OF CORRESPONDENCE: To compare two crossroads we define a measure of correspondence which is composed as follows:

$$S(C_1 \hat{=} C_2) = \left(\frac{1}{\|(c_1, c_2)\|+1} \cdot S((\Omega_{B_1}, \Omega_{S_1}) \hat{=} (\Omega_{B_2}, \Omega_{S_2}))\right). \quad (11)$$

The first term relates to the euclidean distance between the crossroads C_1 and C_2, i.e. the more the two crossroads are spatially separated the less their correspondence. Without referring to their location, $S((\Omega_{B_1}, \Omega_{S_1}) \hat{=} (\Omega_{B_2}, \Omega_{S_2}))$ evaluates the topology of both crossroads by utilizing their branch descriptions. The function basically tries to adapt parameter β that maximizes:

$$\max_\beta \Big(\frac{1}{n}\big(h\left(\mathcal{N}(\bar{b}_{1_1} + \beta, s_{1_1}), \mathcal{N}(\bar{b}_{1_2}, s_{1_2})\right) + \ldots + h\left(\mathcal{N}(\bar{b}_{n_1} + \beta, s_{n_1}), \mathcal{N}(\bar{b}_{n_2}, s_{n_2})\right)\big)\Big) \quad (12)$$

with β being a rotation offset and $h(\cdot,\cdot)$ a function that computes the convolution between two normal distributions in the fashion of [11]. Input to $h(\cdot,\cdot)$ are the crossroads mean branch angles $(\bar{b}_{1_1}, \ldots, \bar{b}_{n_1} \in \Omega_{B_1}, \bar{b}_{1_2}, \ldots, \bar{b}_{n_2} \in \Omega_{B_2})$ alongside their variances $(s_{1_1}, \ldots, s_{n_1} \in \Omega_{S_1}, s_{1_2}, \ldots, s_{n_2} \in \Omega_{S_2})$. In case $|\Omega_{B_1}| \neq |\Omega_{B_2}|$, similarity function $S((\Omega_{B_1}, \Omega_{S_1}) \hat{=} (\Omega_{B_2}, \Omega_{S_2}))$ returns zero.

For each hypothesis from time-step $t-1$, processed by the prediction step of the Kalman filter, the measurement at time-step t that maximizes equation 11 is searched. Valid associations are input to the Kalman innovation step to update the filter's prediction. Also, within each time-step the crossroad's probability of being valid is updated by evaluating the ratio between the number of time-steps a measurement was associated and the number of time-steps no measurement could be assigned since the first occurrence.

IV. TOPOLOGICAL MAPPING

Topological maps represent the environment as a combinatorial structure that minors the importance of geometric information, such as distances, directions or scale, but keeps the relationship between distinct points. In that manner information is simplified with unnecessary details removed, while the vital information required for navigation still remains. These maps are represented by means of graphs that consist of vertices, denoting different places or landmarks and edges connecting adjacent nodes. Topological approaches in

accepted for publication at
International IEEE Conference on Intelligent Transportation Systems, 2011

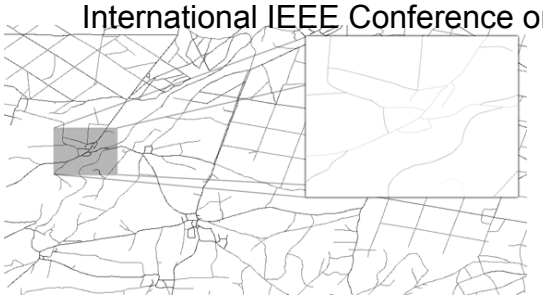


Fig. 7. Raw GIS data, represented as a set of point sequences. In the top right corner a pane of the complete network is cut out.

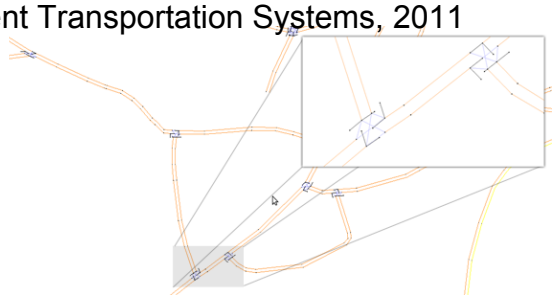


Fig. 8. The resulting road network after extracting road and crossroad features from raw GIS data. In the top right corner crossroads and their topological information is visualized. (*black lines represent the topology of the associated crossroad*).

general determine the position of the robot relative to the model. These models are primarily based on landmarks or distinct, momentary sensor features. Since the resolution of topological maps corresponds directly to the complexity of the environment their compactness can be distinguished to be a key advantage. Due to their compactness they permit fast planning, facilitate interfacing to symbolic planners and problem-solvers and provide more natural interfaces for human instructions (such as: go to place X).
With respect to our localization method, they provide an appropriate environment description, since crossroad descriptions, including their number of branches, relative branch angles and their sequence of occurrence is maintained.

A. GIS DATA

Within our system, GIS (Geo Information System) data provides a means to build the necessary predefined map, since a rich set of environment features, including informations with regard to roads is given. It is composed by vector data including positions (highly precise GPS or UTM coordinates, provided by land surveying offices) and their description. The vector data can define any kind of continuous entity, such as roads, contour lines, borders, etc. and is divided into different layers. A layer is composed of features (line sequences, stand alone points, etc.) and refers to a specific information, such as “highways” or “dirt road”. Figure 7 presents an example of raw GIS data.

B. MAP FEATURE EXTRACTION

Feature extraction is carried out by identifying crossroads, their topology and the distances between adjacent ones in the GIS data by combining GIS-features within the “road”-network layer. A feature in the context of a “road” layer is defined to be a point sequence that connects either two crossroads or one crossroad and a dead end. Hence, features sharing equal endpoints have to be identified to detect possible crossroads within the network. These crossroad hypotheses are then analyzed with respect to their topology. Each crossroad’s center point alongside its outgoing branches is extracted while distances to adjacent crossroads are determined by moving along the outgoing paths. Since GIS data provides unidirectional network data only, the extracted data has to be duplicated to yield a bidirectional road network that allows movement in both road directions. A pane of an extracted road network is shown in figure 8

C. POSITION ESTIMATION

Crossroad hypotheses from LIDAR data are now utilized within a particle filter to estimate the robot’s position within the extracted road network. The particle state $\mathbf{x_p} = (x, y, \Psi)$ is composed by the particle’s global position (x, y) and its rotation Ψ within the map. The first occurrence of a valid crossroad hypothesis $C_H = (c_H, \Omega_{B_H}, \Omega_{S_H})$ causes the particle filter’s initialization. With Ω_N being the set of all the network’s crossroads, the initialization routine tries to find all crossroads $C_N = (c_N, \Omega_{B_N}, \Omega_{S_N}) \in \Omega_N$ that satisfy:

$$S((\Omega_{B_H}, \Omega_{S_H}) \hat{=} (\Omega_{B_N}, \Omega_{S_N})) > \alpha. \tag{13}$$

I.e. with respect to their topology, all the road network’s crossroads are compared to the LIDAR crossroad hypothesis, while their degree of similarity has to exceed a predefined parameter α. Furthermore the number of initialized particles N_{C_N} with regard to a certain crossroad is determined by utilizing the value of $S((\Omega_{B_H}, \Omega_{S_H}) \hat{=} (\Omega_{B_N}, \Omega_{S_N}))$ in the fashion of equation 14.

$$N_{C_N} = \frac{S((\Omega_{B_H}, \Omega_{S_H}) \hat{=} (\Omega_{B_N}, \Omega_{S_N}))}{\sum^{C_N \in \Omega_N} S((\Omega_{B_H}, \Omega_{S_H}) \hat{=} (\Omega_{B_N}, \Omega_{S_N}))} \cdot n \tag{14}$$

In this connection n denotes the maximum number of particles that is available for initialization. Each crossroad that satisfies equation 13 is further processed to yield the robot’s position that explains the initial crossroad measurement the best. At distance $\sqrt{x_{c_H}^2 + y_{c_H}^2}$, which denotes the distance to the measured crossroad’s center, a robot pose hypothesis p_i is placed at every branch of a crossroad. Each hypothesis’s local “view” of the crossroad C_{p_i} is then compared to the measured LIDAR hypothesis C_H to maximize:

$$\underset{C_{p_i}}{\operatorname{argmax}} \left(S(C_H \hat{=} C_{p_i}) \cdot \left(1 - \left|\frac{\beta}{\pi}\right|\right) \right) \tag{15}$$

By exploiting rotation offset β provided by equation 12, equation 15 basically chooses C_{p_i} and thus pose hypothesis p_i that provides the smallest value of β. Finally, at pose hypothesis p_i the number N_{C_N} of particles is initialized.

accepted for publication at
International IEEE Conference on Intelligent Transportation Systems, 2011

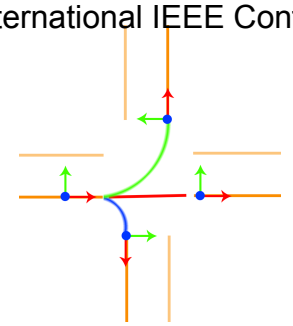


Fig. 9. Possibilites of a particle (*2D-coordinate systems, with the red arrow denoting the forward direction*) to move within a 4-way junction. With each choice a different movement behavior and thus a different progress of the yaw rate is measurable.

Once initialized, a particle's movement is coupled to the direction of the closest lane within the road-network. Position (x_t, y_t) at time-step t is updated according to the constant velocity model [10], while the particle's rotation Ψ is defined to be the closest lane's tangent angle Ψ_L. I.e. each particle moves in parallel to the closest lane with its rotation aligned to the lane's direction.

With each new crossroad measurement every particle's weight w_i is recomputed by:

$$w(i) = S(C_H \stackrel{\wedge}{=} C_{p_i}) \cdot \left(1 - \left|\frac{\beta}{\pi}\right|\right) \cdot P(x_p \mid Z_{\dot{\Psi}}) \qquad (16)$$

With C_H being the crossroad measurement and C_{p_i} the particle's closest crossroad, equation 16 exploits correspondence of equation 11 in combination with rotation offset β provided by equation 12. Hence particles that declare the current crossroad measurement the best are weighted the highest. Furthermore $P(x_p \mid Z_{\dot{\Psi}})$, the probability of observing state x_p given the robot's past yaw rates $Z_{\dot{\Psi}}$, provided by robot's INS (Inertial Navigation System), is incorporated to sustain particles that carry out a movement similar to the robot's. As mentioned, a particle's movement is coupled to its closest lane. Basically, a particle's yaw rate is close to zero, while moving on a lane. But within junctions each particle moves towards another branch, which results in different yaw rates as shown in figure 9. Thus each particle possesses a queue Q_Ψ that contains the former yaw rates of the particle. With regard to the i-th of m particle's queue $Q_{\dot{\Psi}_i}$ we thus get:

$$P(x_{p_i} \mid Z_{\dot{\Psi}}) = \frac{\sum_{j=\left|Q_{\dot{\Psi}_i}\right|}^{0} (q_j - z_j)^2}{\sum_{i=1}^{m} \sum_{j=\left|Q_{\dot{\Psi}_i}\right|}^{0} (q_j - z_j)^2} \qquad (17)$$

, with $q_j \in Q_{\dot{\Psi}_i}$, $z \in Z_{\dot{\Psi}}$, and the queues size $\left|Q_{\dot{\Psi}_i}\right|$ restricted to a defined length. I.e. equation 17 supports particles that have a similar progress of the the yaw rate to the yaw rate of the robot. Especially in the vicinity of 4-way crossroads, the equation gets useful, since the robot's motion provides a hint towards the driven branch.

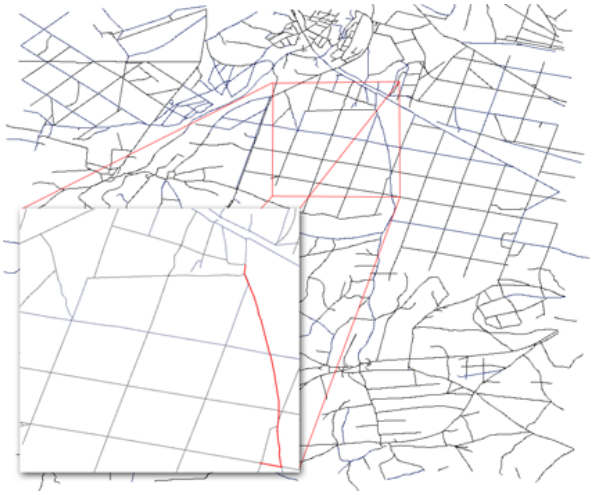
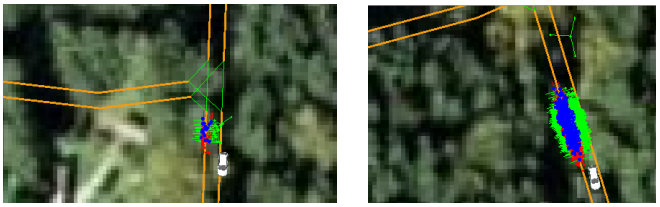
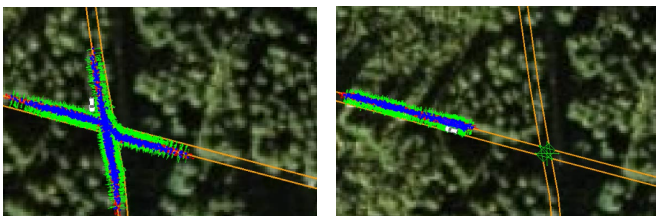


Fig. 10. Complete road network for robot localization. The small cutout shows the area in which the robot has operated. The results of one run in a timbered area (driving direction from top to the bottom) is shown with the red line denoting the path of the best particles.



(a) (b)

Fig. 11. Figure (a) presents the particle filter's initialization at the first detected crossroad in the vicinity of the robot's GPS position (*white vehicle model*). At the initialization, the occurrence of a crossroad forces the distribution of the whole particle set over the complete map. Hence only few particles are initialized close to the GPS position. Figure (b) shows the particle distribution after the occurrence of the second crossroad. Since a crossroad's topology is integrated into the localization process, the majority of particles is located at the true position, even though only two crossroads appeared.



(a) (b)

Fig. 12. Figure (a) presents the particle distribution after a long period without crossroad measurements, before a 4-way-junction, while (b) shows the distribution after passing the crossroad. Notably the robot's motion supported the choice of the correct particles.

accepted for publication at
International IEEE Conference on Intelligent Transportation Systems, 2011

V. CONCLUSIONS AND FUTURE WORKS

A. Results

Throughout a test on a forest track of about 4.2 km length, see figure 10, 8 out of 10 real world crossroads were detected correctly. The forest track itself contained more crossroads than represented in the GIS-data. Possibly the GIS-data was not up to date or some junctions just not significant enough to represent them in the GIS-data. At 6 crossroads in the road network a correct re-localization was established. To align the particle weight onto the robot's motion greatly improved the resampling. In the vicinity of complicated crossroads, like e.g. 4-way road junctions with neighboring branches that were arranged orthogonally, a hint towards the driven branch was provided when the junction was passed. As figures 11(a) and 11(b) show, only a small subset of particles is required to localize the robot already after the second crossroad that occurred. Even after a long period without crossroad measurements, as presented in figure 12(a), with the particles wide spread, the chosen particle weighting function supported these particles the most that carried out the same motion as the robot, see figure 12(b).

B. Conclusions

In this paper we presented work in the area of crossroad detection and topological mapping. The proposed crossroad detection algorithm has proven to be sufficient to detect crossroads in obstacle (positive and negative) rich environments. The particle filter approach to localize the robot worked really well, especially the incorporation of the robot's motion increased its reliability. Furthermore, the interesting fact of describing pathways verbally arose from the topological approach. This offers a new perspective, since road descriptions do not have to be provided by means of coordinates and angles, but by verbal descriptions such as "take the next crossroad's right branch, then after 500 meters take the next crossroad's left branch" and thus may make navigation more human like.

C. Future Works

While the particle filter, to localize the robot within the given GIS data, worked very well, improvements have to be made with regard to crossroad detection. The approach restricts the robot's movement to obstacle rich areas, like forests or inner city environments, while flat areas, like grasslands or fields, are not covered. We thus want to extend the current approach with visual features, that describe obstacles, e.g. green surfaces (trees, grassland, etc.). Furthermore, the output and application of human readable plans is desired, that may lead to a more human like navigation.

VI. ACKNOWLEDGEMENTS

The authors gratefully acknowledge funding by German cluster of excellence COTESYS, Cognition for Technical Systems (also see *http://www.cotesys.org*).

REFERENCES

[1] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller, "An atlas framework for scalable mapping," in *IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA '03.*, vol. 2, Sept. 2003, pp. 1899–1906 vol.2.

[2] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fastslam: A factored solution to the simultaneous localization and mapping problem," in *National Conference on Artificial Intelligence*, September 2002.

[3] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using fastslam," in *IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA '03.*, vol. 2, Sept. 2003, pp. 1985–1991.

[4] A. Morris, D. Silver, D. Ferguson, and S. Thayer, "Towards topological exploration of abandoned mines," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005.*, April 2005, pp. 2117–2123.

[5] S. Vasudevan, S. Gaechter, V. Nguyen, and R. Siegwart, "Cognitive maps for mobile robots–an object based approach," *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 359 – 371, 2007, from Sensors to Human Spatial Concepts.

[6] M. Himmelsbach, T. Luettel, F. Hecker, F. von Hundelshausen, and H.-J. Wuensche, "Autonomous Off-Road Navigation for MuCAR-3 – Improving the Tentacles Approach: Integral Structures for Sensing and Motion," *Kuenstliche Intelligenz*, vol. Special Issue on Off-Road-Robotics, 2011, accepted for publication.

[7] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley: The robot that won the darpa grand challenge: Research articles," *J. Robot. Syst.*, vol. 23, no. 9, pp. 661–692, 2006.

[8] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[9] G. Borgefors, "Distance transformations in digital images," in *Computer Vision, Graphics and Image Processing*, vol. 34, 1986, pp. 344–371.

[10] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley and Sons, Ltd., 2002.

[11] R. Bracewell, *"Convolution" and "Two-Dimensional Convolution."*. McGraw-Hill, 1965.