

# Mobile Robot Self-Localization in Large-Scale Environments

Axel Lankenau, Thomas Röfer

Bremer Institut für Sichere Systeme, TZI, FB3, Universität Bremen,  
Postfach 330440, 28334 Bremen, Germany. E-Mail: alone@tzi.de, roefer@tzi.de

*Abstract*— This paper presents a new approach for the absolute self-localization of a mobile robot in structured large-scale environments. The requirements with regard to both, the necessary a-priori knowledge as well as the sensor equipment, are low. The algorithm scales up very well, due to a hybrid representation of the environment that augments a topological map with metric information. As a consequence, the method is especially suited for usage in large-scale service robotics applications. As an example for a future application, the so-called *Navigation Assistant* of the Bremen Autonomous Wheelchair “Rolland” is discussed. The self-localization results presented below stem from experiments with the wheelchair on a 2,176m long indoor and outdoor parcours on the campus of the Universität Bremen.

*Keywords*— Navigation, Localization, Mobile Robots, Service Robots, Nonholonomic Robots

## I. MOTIVATION

FUTURE generations of service robots are going to be *mobile* in the first place. Both, in classical application areas such as the cleaning of large buildings or property surveillance, but especially in the context of rehabilitation robots, such as intelligent wheelchairs, mobility will be a major characteristic of these devices. After having shown that it is technically feasible to build these robots, additional requirements will become more and more important. Examples of such demands are the operability in common and unchanged environments, adaptability to user needs, and low material costs. To satisfy these requirements, methods have to be developed that solve the fundamental problems of service robot navigation accordingly.



Fig. 1. Bremen Autonomous Wheelchair “Rolland”. It is based on the commercial power wheelchair *Genius 1.522* manufactured by the German company Meyra, and it has been extended by a PC and several proximity sensors.

Pursuing these considerations, we developed a new self-localization approach for the rehabilitation robot “Rolland” (see figure 1 and [1], [2]) within the framework of the project *Bremen Autonomous Wheelchair*. The algorithm requires only minimal sensor equipment (odometry and two sonar sensors), works in unchanged environments and provides a sufficient precision for a robust navigation in large building complexes.

## II. RELATED WORK

THERE are two basic principles for the self-localization of mobile robots [3]: *Relative* approaches need to know at least roughly where the robot started and are subsequently able to track its locomotion. At any point in time, they know the relative movement of the robot with respect to its initial position, and can calculate the robot’s current position in the environment. It has to be ensured that the localization does never lose track, because there is no way to recover from a failure for these approaches. Modern relative self-localization methods make often use of laser range finders. They determine the robot’s locomotion by matching consecutive laser-scans and deriving their mutual shift. Gutmann and Nebel [4], [5] use direct correlations in their *LineMatch* algorithm, Mojaev and Zell [6] employ a grid map as “short term memory”, and Röfer [7] accumulates histograms as basic data structure for the correlation process.

On the other hand, *absolute* self-localization approaches are able to find the robot in a given map without having any a-priori knowledge about its initial position. Even more difficult, they solve the “kidnapped robot problem” [8], where—during runtime—the robot is deported to a different place without being notified. From there, it has to (re-)localize itself. That means, the robot has to deliberately “unlearn” acquired knowledge.

The absolute approaches are more powerful than the relative ones and superior in terms of fault tolerance and robustness. They try to match the current situation of the robot—defined by its locomotion and the sensor impressions—with a given representation of the environment, e. g. a metric map. As this problem is intractable in general, probabilistic approaches have been proposed as a heuristics. The idea is to pose a hypothesis about the current position of the robot in a discrete model of the world

from which its location in the real world can be inferred. A distribution function that assigns a certain probability to every possible position of the robot, is adapted stepwise. The adaptation depends on the performed locomotion and the sensor impressions. Due to the lack of a closed expression for the distribution function, it has to be approximated. One appropriate model is provided by grid-based Markov-localization approaches that have been examined for some time: they either use sonar sensors [9] or laser range finders [10] to create a probability grid. As a result, a hypothesis about the current position of the robot can be inferred from that grid. Recently, so-called Monte-Carlo-localization approaches became very popular. They use particle filters to approximate the distribution function [11], [12]. As a consequence, the complexity of the localization task is significantly reduced. Nevertheless, it is not yet known how well these approaches scale up to larger environments.

Apart from these purely metric representations of the environment, Kuipers *et al.* propose the integration of metric and topological concepts with their “spatial semantic hierarchy” [13]. The idea is pursued by Simmons and Koenig [14] and Nourbakhsh *et al.* [15] by augmenting topological maps with metric information. The resulting self-localization methods also work probabilistically on the basis of the odometry and a local model of the environment perceived with the sensors. Thrun [16] compares grid-based and topological approaches. He finds that especially in large-scale environments, a combination of both paradigms gives the best results with respect to computational complexity and preciseness.

For about a decade, robot navigation research also deals with integrated approaches of simultaneous localization and mapping (so-called *SLAM* techniques). Current results in a hybrid grid-based and topological approach have been presented by Tomatis *et al.* [17]. They employ a 360° laser range finder and extract features such as corners and openings which are used to navigate in a global topological map. In addition, the laser-scans are searched for line structures (walls, cupboards, etc.) which build the basic data structure for several local metric maps (one for each node of the topological map). Choset and Nagatani [18] use generalized Voronoï graphs to represent the robot’s environment topologically. Low-level control laws enable the robot to explore unknown territory by generating and following the Voronoï graphs edges. Localization is done by matching (local) graph fragments with the previously constructed global graph.

### III. MODELING LOCOMOTION AND ENVIRONMENT

**I**N order to be able to determine the robot’s position by matching its situation (i. e. the current and past sensor impressions and its locomotion) with a representation of the environment, comparable models for both, the robot’s situation and the environment, must exist. We present an

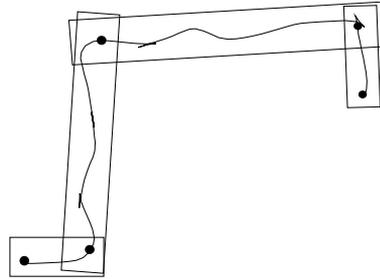


Fig. 2. Route generalization [19]. The figure shows the locomotion of the robot as recorded by its odometry, the detected corners as well as the acceptance areas for each route segment.

approach for the absolute self-localization of a mobile robot that uses as situation model the incremental generalization of traveled tracks introduced by Röfer [19]. The idea is to generalize the locomotion of the traveling robot during runtime to an abstract route description. This description represents the route as a sequence of straight segments that intersect under certain angles, and thus defines the situation of the robot.

Fig. 2 shows the locomotion of the robot as recorded by its odometry system as a solid curved line. The corners recognized by the generalization algorithm are depicted as circles. The rectangular boxes represent the so-called acceptance areas: As long as the robot remains within such a region, it is assumed that the robot is still located in the same corridor. The width of the rectangular boxes is determined with the help of a histogram-based approach from the measurements of two sonar sensors mounted on the wheelchair’s left- and righthand side chassis [19]. The generalization of the traveled track is carried out incrementally, i. e. while the robot moves. Therefore, the distance traveled so far in the current segment as well as the angle to the previous segment may change during runtime depending on the locomotion of the robot. This property of the route generalization approach influences the self-localization method proposed in the sequel, since the information about the current segment is volatile and is thus subject to change.

The abstraction resulting from this generalization method turns out to be very robust with regard to temporary obstacles and minor changes in the environment. Nevertheless, it is only helpful, if the routes are driven in a network of corridors or the like. Fortunately, almost all larger buildings such as hospitals, administration or office buildings consist of a network of hallways.

Thus, the approach presented here confines itself to “corridor-rich” environments. Since we did not yet integrate a mapping component in our algorithm, an a-priori known map in form of a so-called *route graph* [20] must exist. The nodes of a route graph (see figure 3) correspond to decision points in the real world: hallway corners, junctions or crossings. The edges of the graph represent straight corridors that connect the decision points. In addi-

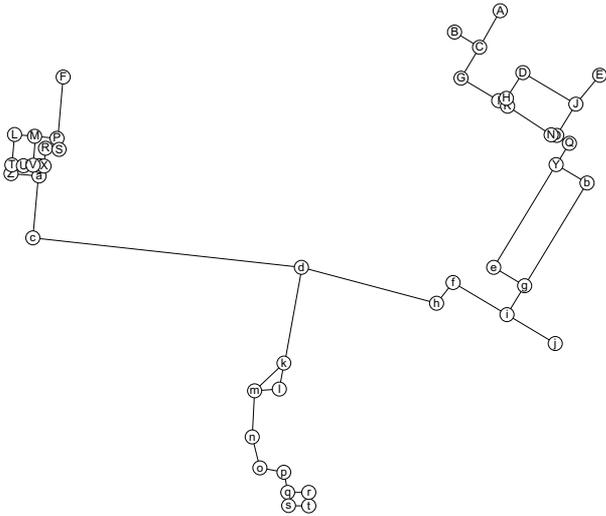


Fig. 3. Route graph. The depicted route graph represents the part of the campus of the Universität Bremen that is relevant for the experiments shown in the results section of this paper (see section V). It consists of 46 graph nodes and 144 junctions. The represented corridors range in length from 4.3m to 179m.

tion to the topological information, the route graph contains (geo-)metric data about the length of the corridors as well as about the included angles. In contrast to grid-based representations, such a data structure is much easier to handle with respect to the required amount of computing time and memory. For example, the campus environment depicted in figure 4 is represented as a list of only 144 so-called *junctions* (see figure 3). A junction is defined by its incoming corridor, the outgoing corridor, the angle included, and the length of the outgoing corridor.

While the representation of the environment as a route graph is formally very similar to Voronoï diagrams as recently used, e.g., by Thrun [16], Zwynsvoorde *et al.* [21], [22], and Choset [18], our localization approach is not only applicable in sensory-rich (indoor) environments but also in pure outdoor or hybrid scenarios such as the campus example presented below. This is because we use the generalization of the robot’s locomotion as reference information for the localization. Thus, we do not have to rely on input from proximity sensors as it is necessary for the Voronoï diagram based approaches (a Voronoï diagram is defined on the basis of the sensor-perceived distance of the robot to objects in its environment).

#### IV. SELF-LOCALIZATION IN ROUTE GRAPHS

WE use a probabilistic approach to ongoingly determine the hallway  $H$  (represented by an edge in the route graph), in which the robot is most likely located at that very moment in time. Since the distance already traveled in  $H$  is also known, an additional offset can be derived. As a result, the position of the robot within the hallway is

found precise enough for most global navigation tasks. The precision is limited by about half of the width of the corridor the robot is located in. This approach turns out to be rather insensitive to odometry errors (see figure 5), because the offsets normally represent only short distances that result from accumulating straight movements, and almost no rotational motion which often causes dead reckoning errors.

##### A. Basic Idea

The basic idea of the self-localization approach is to match the incremental generalization  $R$  of the currently traveled route with the route graph.  $R$  is defined as a sequence of corners as follows:

$$R = \langle C_i \rangle, \text{ where } C_i = (\alpha, l), i \in \{1 \dots n\} \quad (1)$$

In (1),  $\alpha$  is the angle between the incoming and the outgoing segment of a corner, and  $l$  is the length of the outgoing segment. Note that  $\alpha_{C_1}$  is a “don’t care” value, i. e. only the outgoing segment of the first corner is considered, the angle is ignored. Furthermore, the shape and thus the defining parameters of the final corner  $C_n$  are subject to change until a new corner  $C_{n+1}$  is detected (see above).

Each junction  $J$  in the route graph is assigned a parameter  $p$  which indicates the probability that  $R$  finishes in  $J$ . Initially,  $p$  is uniformly distributed over all junctions. The matching quality  $p$  is inductively defined: A route generalization comprising only one corner  $C_1$  directly matches a junction  $J$ , if the corresponding angles and outgoing segment lengths match with regard to a certain metric. A route consisting of the corners  $C_1 \dots C_n$  ( $n > 1$ ) matches  $J$  if its final corner  $C_n$  directly matches  $J$ , and if the initial part of  $R$  (i. e.  $C_1 \dots C_{n-1}$ ) matches the route graph such that the incoming edge of  $J$  is reached.

Then, the matching quality  $p$  is defined as follows:

$$p = h \cdot s_{\Delta d} \cdot s_{\Delta \alpha} \quad (2)$$

Here,  $\Delta d = |d - l|/l$ , where  $d$  is the distance already traveled in a corridor, and  $l$  is the length of the corridor.  $\Delta \alpha$  is the difference between the angle included by the final corner of the route and the one included by the junction of the route graph under consideration. The sigmoid function  $s$  ensures that small deviations with respect to the corridors’ lengths or the angles, respectively, are tolerated whereas large deviations only result in a very weak matching quality. The “history” factor  $h$  is explained in section IV-B.

If the robot stayed in a distinct hallway forever, this method would suffice to quickly determine which junction in the route graph represents the corresponding corridor. In order to be also able to model transitions to adjacent hallways, the junctions have to be detected as corners by the route generalization algorithm, and the relevant junction probabilities have to be propagated through the route graph.

## B. Propagation

If a corner is detected in the trajectory traveled so far, one of the following cases applies: the generalized corner also exists in reality (*correct detection*), the corner does *not* exist in reality (*phantom detection*), or the robot did turn around in a hallway (see below).

Furthermore, it is possible that a junction existing in reality had been passed and had not (yet) been detected by the generalization algorithm (*junction missed*). Usually, this cannot be blamed on the generalization but on the fact that—based only on locomotion data—one cannot distinguish traveling in a straight hallway with no junctions or crossings from traveling in a straight hallway passing several T-junctions. Therefore, the self-localization algorithm has to solve this problem.

The three different cases of a detected corner as well as the case of a missed junction are considered in parallel by processing a temporary probability value  $p'_{ij}$  for each of these cases  $i$  for every junction  $j$  in the route graph. The ultimate decision about the definitive shape of the so far final route corner  $C_n$  is not made until another corner  $C_{n+1}$  is detected. Only then, the characteristics of  $C_n$  are irrevocably known, as mentioned above. The maximum of the temporary probability values  $p'_{ij}$  determines which case is “believed in”.

During the propagation process, the “history”  $h$  of a junction  $J$  is assigned the maximum probability of those junctions that comprise an outgoing segment that leads to  $J$  (see (2)). Updating and propagating the probabilities is of linear complexity with respect to the number of junctions representing the environment. Since the number of junctions is usually related sublinearly (or linearly at most) to the size of the environment, the approach scales very well.

## C. Turning Around Within a Corridor

As the Bremen Autonomous Wheelchair is a nonholonomic vehicle, the orientation of the robot within a corridor has to be modeled as well. Since turning on the spot is generally not possible for the wheelchair, each real corridor is represented by at least two unidirectional edges in the route graph (forward and backwards orientation in the corridor). Handling transitions between these edges, i. e. dealing with turns of the wheelchair in a hallway, requires some special effort. This is because a turning maneuver can be carried out at any position within the hallway. In contrast to that, leaving the corridor is only possible at junctions. To be able to also cover turns, the set of junctions that initially form the route graph is extended with so-called “*turn-junctions*” at program start. As an example, consider the route graph depicted in figure 3 that is used for the experiments presented in section V. The 144 junctions of this route graph require an additional set of 102 turn-junctions. The upper bound of the number of required turn-junctions for a route graph with  $n$  “real” junctions is  $2n$ . However, in typical

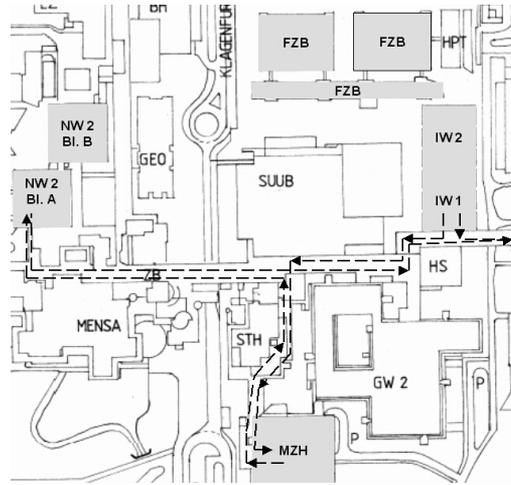


Fig. 4. Sketch of the campus of the Universität Bremen. The area shown covers about  $380\text{m} \times 322\text{m}$ . The buildings visited by the wheelchair are grey shaded. The route traveled is depicted as a dashed line. Its length amounts to 2,176m.

environments, it often happens that two or more junctions share one turn-junction, e. g. junctions  $cdh$  and  $kdh$  in figure 3 both need the turn-junction  $dhd$ . The incoming and the outgoing segment of these turn-junctions represent the same hallway (forwards and backwards direction) and include an angle of  $180^\circ$ . After having generated the turn-junctions at program start, they are dealt with as if they were “normal” junctions in the sequel. The only exception is that the deviation of the length is ignored when calculating the matching quality of a generalized route corner with such a turn-junction (undershooting is accepted for turn-junctions).

## V. RESULTS

EXPERIMENTS with the Bremen Autonomous Wheelchair “Rolland” have been carried out on the campus of the Universität Bremen. The wheelchair is driven indoors and outdoors, visits seven different buildings and passes the boulevard which connects the buildings. The traveled distance amounts to 2,176m. Traveling along this route with a maximum speed of 84cm/s takes about 75min. While traveling, the wheelchair generates a log file which records one state vector every 32ms. Such a state vector contains all the information available for the wheelchair: current speed and steering angle, joystick position, current sonar measurements, and complete laser scans. As mentioned, only locomotion data and the measurements of two sonar sensors are used for the localization approach presented here. Feeding the log file (192MB) into our simulator SimRobot [23], we are able to test the self-localization approach with *real* data on a simulated robot. Note that the simulator works in real-time, i. e. it also delivers the recorded data in 32ms intervals to the connected software modules, one of which is the self-localization module.

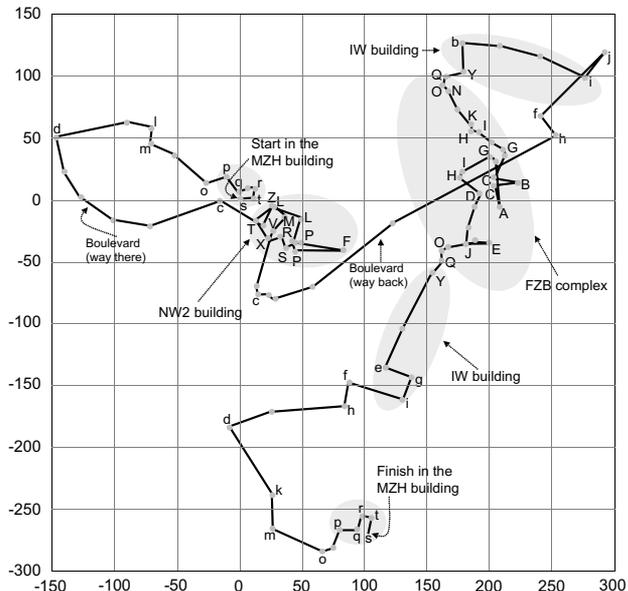


Fig. 5. Annotated route generalization data used for the experiments described in the results section. Axes’ unit is meter. Obviously, the data logged by the wheelchair’s odometry is not that precise. While distances are measured relatively precise, angles between segments are often completely wrong: e. g., compare the different generalizations of the “Boulevard” on the way there and back from the NW2 building. The circles along the route depict the corners found by the generalization algorithm. Those labeled are correctly detected, the others are “phantom” corners. The labels correspond to the junction labels used in figure 3. The route starts at (0/0) in the MZH building. The indoor parts of the route are indicated by the grey-shaded areas labeled with the buildings’ names. Even though start and end point (bottom center, labeled “Finish”) are identical in reality, they differ by 290m in the recorded data.

For the evaluation of the approach, a laser-scan map of the whole route was generated, using the scan matching method presented in [7]. For such a large scene, the laser map deviates from the original layout of the environment in that the relative locations of the buildings are not 100% correct. Therefore, the route-graph was embedded into the laser scan map making it possible to compare both localization results on a metric basis while traveling through the route with simultaneously active scan matching and route localization modules<sup>1</sup>.

The deviations between the metric positions determined by the reference locator and the locations calculated by the route localization approach presented in this paper are depicted in figure 6. Note that the horizontal axis corresponds to *time* along the route and not to *distance*, i. e. the wheelchair stopped several times and also had to shunt sometimes, so that distances along this axis do not directly correspond to metric distances along the route.

The new method represents the environment as edges of a graph that should be—but in fact *are* not always—centered in the corridors. As a consequence, the metric pre-

<sup>1</sup>That is the reason why the layout of the route graph depicted in figure 3 differs from the map shown in figure 4.

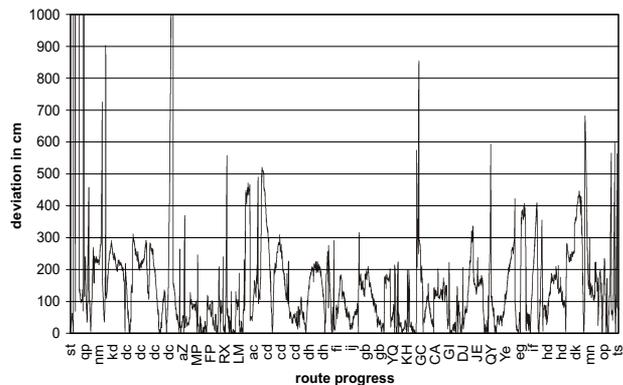


Fig. 6. Deviations of the route localization from the metric positions determined by the laser scan based localization. The letters correspond to segments between the junction labels used in figure 3, but due to the lack of space, some are missing.

cision of the approach is limited: deviations perpendicular to a corridor can reach its width, which can be more than 10m outdoors (e. g. corridor *dc*). Deviations along a corridor may be caused by one of three reasons: first, they can result from the location at which the current corridor was entered. The bandwidth of possibilities depends on the width of the previous corridor. Second, deviations can be due to odometry errors, because the wheelchair can only correct its position when it drives around a corner. In case of the boulevard (corridor *cdh*), the wheelchair has covered approximately 300m without the chance of re-localization. Third, deviations can also result from a certain delay before a turn is detected (e.g. the peak behind *JE* in Fig. 6).

Even though the odometry data turned out to be very bad (see Fig. 5), the approach presented here is able to robustly localize the wheelchair. It takes a while before the initial uniform distribution adapts in such a way that there is sufficient confidence to pose a reliable hypothesis about the current position of the robot. But if this confidence is once established, the position is correctly tracked.

## VI. APPLICATION SCENARIO

THE self-localization approach presented here will be used to extend the *Route Assistant* of the Bremen Autonomous Wheelchair [1] that has been developed in cooperation with the neurological clinic of a Bremen hospital.

The current version of the Route Assistant provides the following functionality: During a teaching phase, the system explores the routes and places pertinent for the future user(s). If, e. g., the wheelchair is used in a rehabilitation center for amnesic patients, the routes to all relevant places in the building could be learned and stored for later replay with the help of the generalization algorithm mentioned in section III. In the replay mode, a nurse chooses a certain target for the patient in the wheelchair. Similar to a GPS-based navigation system, the large-scale navigation is done by the Route Assistant by giving instructions where to go

at decision points, enabling the patient to travel around on his or her own. The patient is independently responsible for controlling the vehicle with respect to local maneuvers such as obstacle avoidance. At the current state of development, the Route Assistant is restricted to scenarios in which it initially knows its position in a specific route. Then, it is able to direct the user to the goal of this single route. The major new contribution of the self-localization approach presented here is the robustness it provides the Route Assistant with. As networks of routes are represented in the route graph, an erroneous maneuver of the user can easily be dealt with: The self-localization method continuously calculates the current position of the wheelchair in the route graph which enables a simple planning module to find a path to the goal.

## VII. CONCLUSION AND FUTURE WORK

**S**ELF-LOCALIZATION of mobile robots in large-scale environments can be efficiently realized if a hybrid representation of the environment is used. The probabilistic approach presented here matches an incremental generalization of the traveled route with an integrated topological-metric map, the *route graph*. Real-world experiments at the Universität Bremen showed the robustness and efficiency of the algorithm. Nevertheless, it should be regarded as a basic method for absolute self-localization that can be extended on demand. In the first place, a disambiguation of situations and the resulting reduced time for the initial localization can be obtained if the route generalization and the route graph were augmented by feature vectors.

The algorithm will be extended such that self-localizing becomes possible even in a-priori unknown environments, i. e. it will tackle the SLAM problem. For this purpose, the robot has to build the route graph from scratch during runtime and, subsequently, it has to solve the problem of *place integration*. That means, it has to find out whether its current position is already represented in the route graph, or whether it is located in a corridor that is so far unknown.

## ACKNOWLEDGEMENTS

**T**HE Deutsche Forschungsgemeinschaft supports this work through the priority program “Spatial Cognition” and another research program.

## REFERENCES

- [1] A. Lankenau and T. Röfer, “The Bremen Autonomous Wheelchair – a versatile and safe mobility assistant,” *IEEE Robotics and Automation Magazine*, “Reinventing the Wheelchair”, vol. 7, no. 1, pp. 29–37, 3 2001.
- [2] T. Röfer and A. Lankenau, “Ensuring safe obstacle avoidance in a shared-control system,” in *Proc. of the 7th International Conference on Emergent Technologies and Factory Automation*, J. M. Fuertes, Ed., 1999, pp. 1405–1414.
- [3] J. Borenstein, H. R. Everett, and L. Feng, *Navigating Mobile Robots – Systems and Techniques*, A. K. Peters, Ltd., USA, 1996.
- [4] J.-S. Gutmann and B. Nebel, “Navigation mobiler Roboter mit Laserscans,” in *Autonome Mobile Systeme*, P. Levi, Th. Bräunl, and N. Oswald, Eds., Berlin, Heidelberg New York, 1997, Informatik aktuell, pp. 36–47, Springer.
- [5] J.-S. Gutmann, T. Weigel, and B. Nebel, “A fast, accurate, and robust method for self-localization in polygonal environments using laser-range-finders,” *Advanced Robotics*, vol. 14, no. 8, pp. 651 – 668, 2001.
- [6] A. Mojaev and A. Zell, “Online-Positionskorrektur für mobile Roboter durch Korrelation lokaler Gitterkarten,” in *Autonome Mobile Systeme*, H. Wörn, R. Dillmann, and D. Henrich, Eds., Berlin, Heidelberg, New York, 1998, Informatik aktuell, pp. 93–99, Springer.
- [7] T. Röfer, “Building consistent laser scan maps,” in *Proc. of the 4th European Workshop on Advanced Mobile Robots (Eurobot 2001)*, 2001, vol. 86 of *Lund University Cognitive Studies*, pp. 83 – 90.
- [8] S. P. Engelson and D. V. McDermott, “Error correction in mobile robot map learning,” in *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, May 1992, pp. 2555–2560.
- [9] A. Elfes, “Occupancy grids: A stochastic spatial representation for active robot perception,” in *Autonomous Mobile Robots*, S. S. Iyengar and A. Elfes, Eds., Los Alamitos, California, 1991, vol. 1, pp. 60–70, IEEE Computer Society Press.
- [10] W. Burgard, D. Fox, and D. Henning, “Fast grid-based position tracking for mobile robots,” in *KI-97: Advances in Artificial Intelligence*, G. Brewka, Ch. Habel, and B. Nebel, Eds., Berlin, Heidelberg, New York, 1997, Lecture Notes in Artificial Intelligence, pp. 289–300, Springer.
- [11] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, “Monte Carlo localization: Efficient position estimation for mobile robots,” in *Proc. of the National Conference on Artificial Intelligence*, 1999.
- [12] S. Thrun, W. Burgard, and D. Fox, “A Real-Time Algorithm for Mobile Robot Mapping With Applications to Multi-Robot and 3D Mapping,” in *Proc. of the IEEE Int. Conf. on Robotics & Automation*, 2000.
- [13] B. Kuipers, R. Froom, Y. W. Lee, and D. Pierce, “The semantic hierarchy in robot learning,” in *Robot Learning*, J. Connell and S. Mahadevan, Eds. 1993, pp. 141–170, Kluwer Academic Publishers.
- [14] R. Simmons and S. Koenig, “Probabilistic robot navigation in partially observable environments,” in *Proc. of the Int. Joint Conf. on Artificial Intelligence, IJCAI-95*, 1995, pp. 1080–1087.
- [15] I. Nourbakhsh, R. Powers, and S. Birchfield, “Dervish: An office-navigating robot,” *AI Magazine*, vol. 16, pp. 53–60, 1995.
- [16] S. Thrun, “Learning maps for indoor mobile robot navigation,” *Artificial Intelligence*, vol. 99, pp. 21 – 71, 1998.
- [17] N. Tomatis, I. Nourbakhsh, and R. Siegwart, “Simultaneous localization and map building: A global topological model with local metric maps,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2001)*, Maui, Hawaii, October/November 2001.
- [18] 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 – 136, April 2001.
- [19] T. Röfer, “Route navigation using motion analysis,” in *Proc. Conf. on Spatial Information Theory '99*, Berlin, Heidelberg, New York, 1999, vol. 1661 of *Lecture Notes in Artificial Intelligence*, pp. 21–36, Springer.
- [20] S. Werner, B. Krieg-Brückner, and Th. Herrmann, *Modelling Navigational Knowledge by Route Graphs*, vol. 1849 of *Lecture Notes in Artificial Intelligence*, pp. 295–316, Springer, Berlin, Heidelberg, New York, 2000.
- [21] D. van Zwynsvoorde, T. Simeon, and R. Alami, “Incremental topological modeling using local Voronoi-like graphs,” in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and System (IROS 2000)*, Takamatsu, Japan, October 2000, vol. 2, pp. 897 – 902.
- [22] D. van Zwynsvoorde, T. Simeon, and R. Alami, “Building topological models for navigation in large scale environments,” in *Proc. of IEEE Int. Conf. on Robotics and Automation ICRA 2001*, Seoul, Korea, May 2001, pp. 4256 – 4261.
- [23] T. Röfer, “Strategies for using a simulation in the development of the Bremen Autonomous Wheelchair,” in *Simulation-Past, Present and Future*, R. Zobel and D. Moeller, Eds. 1998, pp. 460–464, Society for Computer Simulation International.