To appear in **Proceedings IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI) 2001**

An Integrative Framework for Global Self-Localization

Joachim Weber, Lutz Franken, Klaus-Werner Jörg, Klaus Schmitt,
Ewald von Puttkamer

Computer Science Department, Robotics & Process Control Research Group
University of Kaiserslautern, D-67653 Kaiserslautern, Germany
http://ag-vp-www.informatik.uni-kl.de

***Abstract*--Concerning the robustness of mobile robot navigation, global self-localization is a key feature for many service applications. In this paper we describe an efficient Bayesian approach for hybrid topological/metric navigation, which is designed to exploit information from multiple sources of sensor data. Experiments with a combination of odometry/laserscans/computer vision show the system's ability to generate initial position hypotheses, to cope with environmental ambiguities and to recover from severe position errors.**

***Index Terms*--localization, Bayes rule, navigation, mobile robots, probabilistic reasoning**

I Introduction

In the development of new application fields for mobile service robots, *robustness* of self-localization is of vital importance. Operation conditions in the service domain differ decisively from those in industrial applications, where target environments typically have a predictable dynamic component and usually offer the opportunity to install guidance support (active beacons, induction lines, etc.). Instead, service robots have to deal with populated, unprepared, possibly mutable and generally unpredictable environments, requiring a higher degree of flexibility and error recovery capabilities.

Tracked self-localization techniques (e. g. [8], [9], [10]), which are used in state-of-the-art developments, provide high accuracy but insufficient error recovery properties in highly dynamic environments. Since they are designed to *optimize* position estimates within a limited tolerance, they are casually unable to recover from errors accumulated in areas which are - at least temporarily - unrecognizable. Especially, tracked self-localization techniques are inadequate to solve the so-called *bootstrap*-problem, i. e. the generation of initial position hypotheses.

In order to overcome these deficiencies, a new class of *global* localization techniques have been developed in recent years (e. g. [1], [2], [3], [4], [7]). By any means, these approaches permanently create and observe multiple position hypotheses and are thus able to a) solve the bootstrap-problem and b) recover from arbitrary position errors - as long as the target area is not completely symmetrical and unique environmental features can be detected to resolve ambiguities. However, the price for this increase in robustness is typically a loss of accuracy. Some of these techniques are purely topological, i. e. the target environment is discretized into few characteristic places.

The approach presented in this paper has been developed in the CAROL (**C**amera Based **A**daptive **Ro**bot Navigation and **L**earning) research project and is an extension of the global localization frame introduced in [7].

The rest of the paper is organized as follows: section 2 addresses the environment representation, while section 3 shortly introduces the data sources. The self-localization algorithm is treated in section 4, and experimental results are presented in section 5. Section 6 offers some concluding remarks.

II World Model

Basically, we distinguish between two different types of environment representations for navigation purposes: metric and topological world models [14].

Due to their easy implementation and efficient handling, metric representations (two- or three-dimensional maps of environment features with a unique coordinate system) are an obvious choice when ranging devices like laserscanners or ultrasonic sensors are used. Main problems are difficulties to integrate non-metric information and to guarantee consistent map building by the assembly of sensor data. The latter problem is topic of intensive research (e. g. [11], [12], [13]).

Topological world models represent the environment as graphs of nodes (distinctive places) and connecting edges (pathways). Advantages of topological world models compared to metric ones are:

- Only topological, not precise metric correctness has to be achieved during map building.
- Path planning is reduced to direct graph search.
- Sensor data is handled as attributes to nodes and/or edges, which simplifies the fusion of information from different sources and the integration of non-metric sensor data.

A basic disadvantage of topological world models is the poor environment resolution, which is insufficient for

any other purpose than navigating from one region to another. Thus, real implementations often realize mixed architectures, like the metric world model in [10], which is extended by a topological graph to facilitate path planning or the topological model in [3], which has been augmented with metric information for localization purposes.
In contrast to their industrial counterparts, service robots usually don't require high accuracy in *absolute* coordinates of a unique coordinate system (e.g. for precise track following on defined pathways, movement coordination of multiple robots, etc.) but rather *local* precision (e.g. for room cleaning, docking, etc.). Consequently, we argue to benefit from the advantages of both, metric and topological models, by using a topological graph as the backbone of environment representation, and at the same time establishing local metric coordinate systems in the nodes' regions (Fig. 1). This preserves the advantages of local precision and simple data fusion, but eliminates the expensive and difficult necessity to establish global map consistency.
In our scheme, each node stores the approximate positions of its nearest topological neighbours with respect to its own local coordinate system. That reveals our world model as a true hybrid, since maximum accuracy of neighbour estimate is equivalent to the existence of a correct metric world model, while minimum precision stands for pure topological representation.
In practice, instead of managing many different coordinate systems we're using only *one* global coordinate system, where approximate correctness is only expected for immediate topological neighbours. Small errors between neighbours are allowed and the map may overlap - as long as topologically remote regions are affected, only. However, when closing big cycles during map building, large errors between the latest connected nodes are inevitable. In this case, corrective transformation parameters are stored in the corresponding graph edges.
In topological maps, sensor data usually is stored in the nodes' data structures. Due to the exclusive use of directed sensors in the CAROL project (laserscanner and camera), we chose to place sensor data *bidirectionally* in the *arcs*' ends, instead. Thus, the robot 'knows' a node's appearance for each registered arc in both directions of that particular pathway.

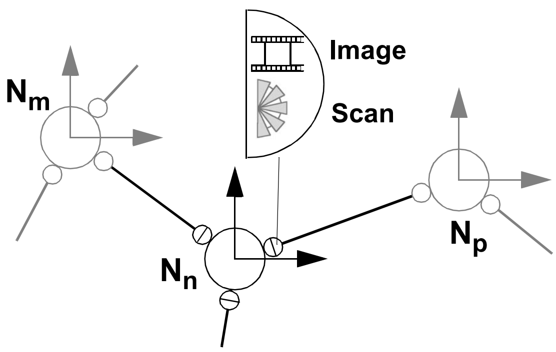

Figure 1. Topological nodes and their local coordinate systems

III Sensors

Besides odometry, two main sources of localization information were used to achieve the experimental results presented in section 5:

A. Laserscanner

In the automated map building phase, reference scans are inserted into the topological graph. During self-localization, the feature-based APR (**A**nchor **P**oint **R**elation Matching) algorithm [6] tries to find matches in this reference set for each new laserscan. Result of each matching cycle is a certain number of tripels $(id_i, \vec{v}_i, q_i)$, where id_i is the identifier of a similar reference scan, and $\vec{v}_i$ a vector (x_i, y_i, ϕ_i) of translational and rotational parameters to align reference scan id_i with the current laserscan s. A quality measure for the alignment of id_i and s is given by q_i. An APR matching cycle thus produces a *multitude* of *weighted* matching hypotheses for each new laserscan. Alignment precision lies within few centimeters.

B. Computer Vision

In order to recognize robot positions by means of computer vision, we apply a modified implementation of the selforganizing approach introduced in [5]. In this technique, two hierarchical neural nets are trained with typical feature-filtered colour images of the target environment. During this offline training, the first neural net creates classes of representative feature combinations, while the second net, referring to these pixel classes, separates the image input space into classes of similar scenes. Thus, the neural nets autonomously learn the specifics of the target environment. Classification results are generally stable concerning moderate rotation and translation. For visual self-localization, the current image scene classes are compared with those stored in the topological graph.

IV Self-Localization

In classic *tracked* localization techniques (e. g. [8], [9], [10]) only one robot position estimate is observed and new sensor data is used to proceed and optimize (e. g. by Kalman filters) the position belief in space. However, since being *unimodal*, these techniques are able to compensate limited position errors, only. As mentioned in section 1, unimodal tracked localization provides high precision but insufficient error recovery from severe position errors. However, these errors have to be expected in dynamic environments. Furthermore, tracked localization is only applicable for metric world models and the starting position has to be provided.

In contrast, *global* self-localization techniques have to be *multimodal*, i. e. the ability to track multiple hypotheses in parallel. A common approach (e. g. in [1], [3], [4], [7]) is to model position belief as probability distribution over the target environment. Bayes rule is used to fuse current sensor information and last position belief in update cycles.

A. Position Hypotheses

The decision how to discretize the position probability distribution is essentially influenced by the underlying world model. In topological representations (e. g. [3], [4]) probability values are usually computed for *transitions* between nodes, rather than for the nodes themselves. This allows the consideration of the robot's rough orientation. In the metric Markov approach in [1] the robot's three dimensional *configuration space* (with dimensions x, y and orientation ϕ with respect to the global coordinate system) is discretized into fine-grain 3D-grids. In the update cycles the probability value for each cube is computed depending on new sensor data. Due to its large computational expense, this method has been superseded by a more efficient sample-based method [2].

For the hybrid world model described in section 2, the first discretization scheme doesn't provide the desired resolution, while the second is applicable to consistent metric representations, only. The discretization scheme we propose, analogous to the existence of local coordinate systems, presumes local *configuration spaces* which are united to a non-consistent super space.

Position beliefs in the topological graph at time t are represented as set of *explicit position hypotheses* PH_t. A position hypothesis discretizes a particular position estimate within a certain tolerance, moves in super space according to the robot's relative manoeuvres, changes its probability and shape due to new sensor data, and possibly 'dies' when never confirmed. Formally, the set of explicit position hypotheses is defined as: $PH_t = \{h_{t,i} | 1 \le i \le N\}$ with $h_{t,i} = (\vec{p}_i, \vec{u}_i, n_i, \lambda_i)$. A position hypothesis $h_{t,i}$ is valid in the context of node n_i, referring to the fact that sensor data from this node has been used to create or confirm $h_{t,i}$. In n_i's geometrical context, the hypothesis' center of gravity is given by vector $\vec{p}_i = (x, y, \phi)$, while $\vec{u}_i$ is the description of an uncertainty perimeter around $\vec{p}_i$. The probability of the robot position actually being inside the uncertainty region is reflected by value λ_i. For matters of simplicity and with respect to the perimeters used in the experiments of section 5, we assume $\vec{u}_i$ as a vector (r, Δ), r be-

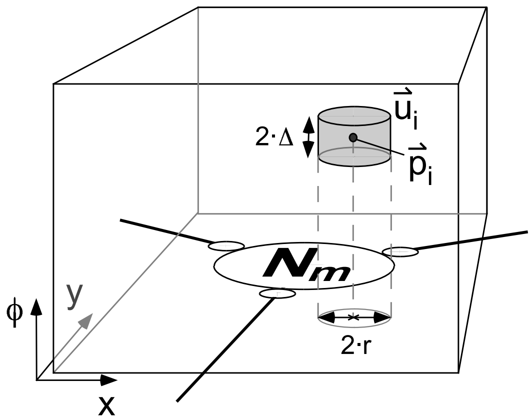

Figure 2. Local configuration space of node N_m and position hypothesis with uncertainty region

ing a radius of translational tolerance and Δ a tolerance angle (Fig. 2). Consequently, the uncertainty region around $\vec{p}_i$ is modelled as a cylinder in the local configuration space of node n_i=N_m.

Provided that the uncertainty regions in PH_t don't intersect in the super space, the probability values of the position hypotheses, together with the likelihood of 'rest' hypothesis $h_{t,0}$ have to sum up to 1. This special hypothesis represents the probability that the actual robot position is *not* in one of the explicit hypotheses.

B. Update

We distinguish between three types of sources for localization information: *tracking*, *primary* and *validation* data sources or *sensors*. ('Sensor' is understood in the meaning of a logical rather than a physical sensor.) New sensor data of any kind is processed on availability.

Tracking data sources only provide relative movement information, e. g. from odometry, inertial systems, tracked self-localization, or a combination of these. According to the indicated relative movement, the $\vec{p}_i$-vectors of all $h_{t,i}$, i>0 are modified to reflect the estimated position change. Since tracking sources don't provide any useful information to solve the global localization problem, the probability values of the $h_{t,i}$ are not affected. However, because tracking data is generally erroneous, the growing position uncertainty is reflected by *expanding* the uncertainty regions $\vec{u}_i$.

Primary sources provide information which is necessary to generate new or to verify old position hypotheses. We assume that the processing of a new sensor observation o results in a set of weighted position *alternatives* $PA_t = \{a_j | j \ge 0\}$, $a_j = (\vec{s}_j, n_j, \kappa_j)$ where $\vec{s}_j$ is a possible robot position vector in the local coordinate system of node n_j. The value κ_j is a measure for the conditional probability $P(o|\vec{s}_j)$ of observing o when being at position $\vec{s}_j$.

A first step checks if alternative a_j confirms an existing position hypothesis, i. e. if $\vec{s}_j$ lies in one of the $h_{t,i}$'s uncertainty regions. Since the robot might currently be heading out of one node context into another, $h_{t,i}$ and the confirming a_j don't nessecarily need to be in the same geometrical node context. Consequently, the comparison of these two positions has to take place in the super space and the $h_{t,i}$'s uncertainty perimeter $\vec{u}_i$ needs to cover both, the a_j's sensor errors *and* the expected map building error for two neighbouring nodes.

A confirmed position hypothesis $h_{t,i}$ takes over node context n_j from the validating alternative a_j, merges its center position $\vec{p}_i$ with $\vec{s}_j$ and resets uncertainty perimeter $\vec{u}_i$ to an appropriate size. If position alternative a_j doesn't confirm an existing position hypothesis, a new hypothesis $h_{t,m}$ is created from the values of a_j, initializing uncertainty region $\vec{u}_m$ and probability λ_m with default values.

The new probability distribution following the observation of o is computed by updating all λ_i using Bayes rule:

$$\lambda_i^{new} = P(h_{t,i}|o) = \frac{P(o|h_{t,i}) \cdot P(h_{t,i})}{\sum_{k=0}^{N} P(o|h_{t,k}) \cdot P(h_{t,k})}$$

The probability $P(h_{t,i})$ prior to observing o is given by λ_i^{old}, and $P(o|h_{t,i})$ is - for new and confirmed position hypotheses - provided by the a_j's κ-value. For unconfirmed $h_{t,i}$ we assume a value for $P(o|h_{t,i})$ which is smaller than the lowest κ_j-probability, instead. This guarantees the reward of all confirmed position hypotheses in relation to the unverified.

Since the denominator sum is obviously equal for all $P(h_{t,x}|o)$, it is replaced by a normalization factor σ which ensures that the whole distribution sums up to 1. Thus, the formula can be simplified to:

$$\lambda_i^{new} = \frac{\kappa \cdot \lambda_i^{old}}{\sigma}$$

At this point, the question is still open how to modify the probability of 'rest' hypothesis $h_{t,0}$, which should reflect the system's „scepticism“ concerning the current explicit position hypotheses. In the ideal scenario, one single $h_{t,i}$ is permanently confirmed and consequently dominates the probabilities of other position hypotheses by some orders of magnitude. Due to ambiguities in the environment, it might occur that more than one position hypothesis achieves high probability values until the detection of unique features. In both cases the real position should be represented in one of the explicit hypotheses and $P(h_{t,0})$ should be low. Worst case is the robot operating in an unrecognizable environment part, either causing the sensors to stop producing localization data or causing them to 'hallucinate', i. e. to permanently create new, invalid hypotheses. Consequently, we propose to periodically check for the occurrence of hypothesis *confirmations*. If no hypothesis has been confirmed by primary or validation sources in the last period, $h_{t,0}$ is rewarded, or penalized in the opposite case.

Validation sources provide useful information to confirm *existing* position hypotheses, but are inadequate to create new ones by themselves. This is typically due to three main reasons:

- Sensor resolution is too low. (E. g. a logical 'door' sensor might be useful to confirm existing position hypotheses. However, it seems unfavourable to apply the sensor for hypothesis creation in environments with many doors.)
- Mapping data for a particular sensor hasn't been collected area-wide and large parts of the environment are not represented in the graph's data set.
- Data processing for a logical sensor is too slow to initiate a global correspondence search. Thus, search space has to be reduced to the hypotheses' uncertainty regions (e. g. for iterative algorithms like [15], which require good starting points).

Confirmations from validation sources are processed exactly like those of primary sources.

V Experimental Results

In the first experimental setup, odometry serves as tracking sensor and the APR algorithm as sole primary source. In Fig. 3 the environment structure (size approx. 50x70 meters) is indicated by the stored laserscans. Please note, that no scans are available in the middle of two corridors. In these regions all doors had been closed and since no APR-features were detectable, no reference scans have been inserted into the topological graph.

Each triangle indicates the position of the currently best rated $h_{t,i}$ during a test course. Fig. 4(a) shows the probability trend of the best rated hypothesis together with the likelihood of $h_{t,0}$ on a track section between starting position and milestone 4. The reward of $h_{t,0}$ in the absence of confirmations is chosen *unpragmatically* high, i. e. the system is quickly forced back into uncertainty after only few periods of missing validations.

Starting with the first laserscan, the correct position hypothesis achieves the highest probability by successive confirmations, leading to the peak at milestone 1. Afterwards, the robot heads into a long, featureless corridor, causing a dramatic probability decrease to near-zero. Since *none* of the existing position hypotheses is confirmed in the corridor, the probability *proportions* are sustained and the correct hypothesis remains *best rated*. At milestone 2, the probability values have decreased to such a low level, that even some confirmations in this region can't produce a visible peak. Thus, after covering the same distance back through the corridor to milestone 3, but starting with a much smaller λ-value than on the way forth, the hypothesis' probability is virtually equal

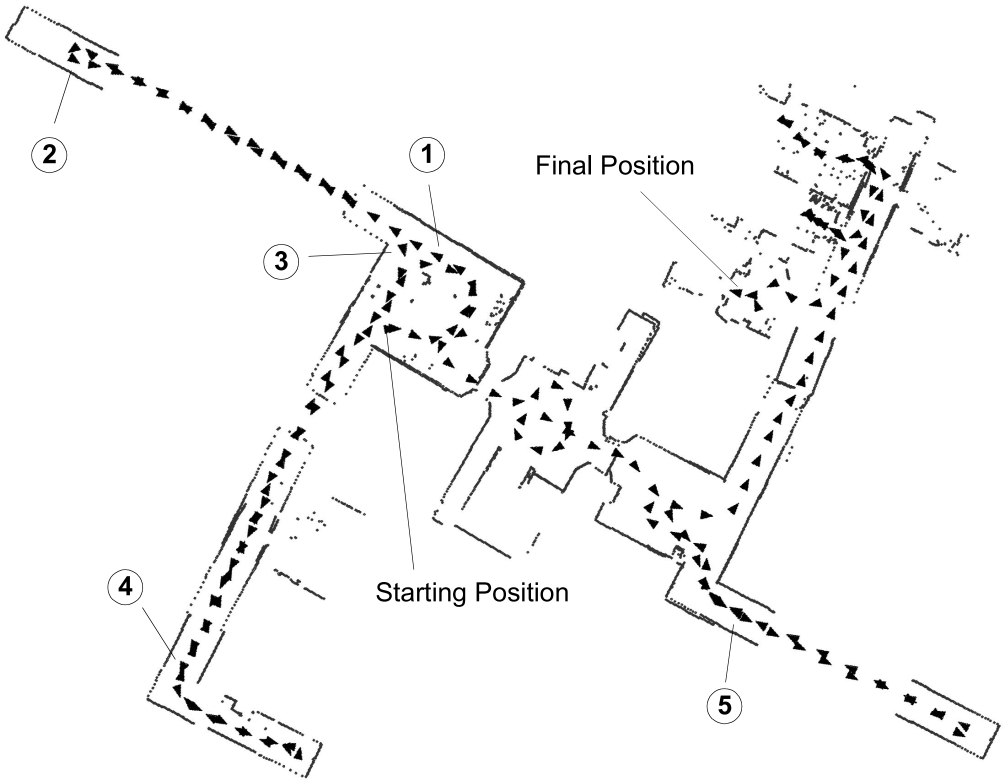

Figure 3. Environment structure (approx. 50x70 m) and test course

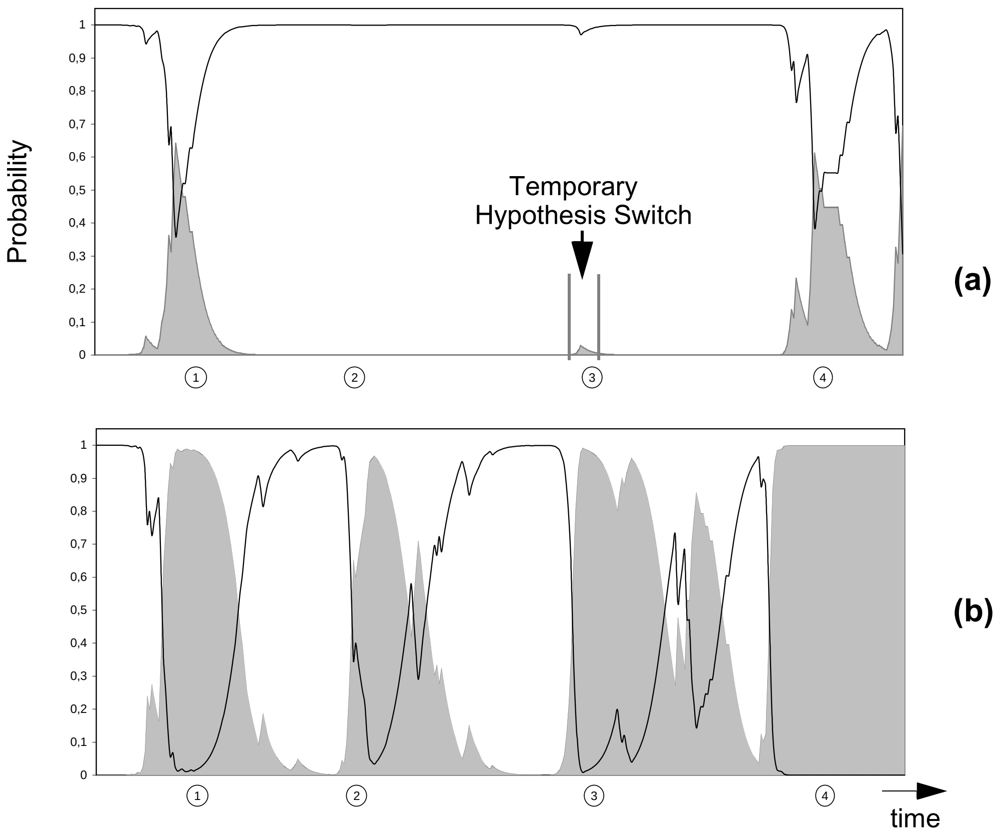

Figure 4. Probability trend for best hypothesis (filled area) and 'rest' hypothesis (solid line) using odometry/laserscanner (**a**) and additional image classification (**b**)

'residual' or 'background' likelihood. Unfortunately, by reaching a real position near milestone 3 the first new APR results favour another hypothesis, temporarily causing a switch of the best rated position estimate to a position near milestone 5. However, these few confirmations of a wrong position hypothesis are surmounted after a few seconds. Beyond milestone 4 the correct hypothesis' probability stabilizes at higher values and remains as the best rated until reaching the final position. A less excessive reward of $h_{t,0}$ averts the position tracking discontinuity easily, but this example shows the system's ability to provide bootstrap-information and to recover from arbitrary position errors, even under the application of extreme parameter settings.

The second experiment takes the same test course sensor data and parameter settings, but additionally considers visual information. Principally, the image classification technique could also be used for the *creation* of position hypotheses by searching the current image class in the topological graph. However, since the nodes in the experiment are up to 5 meters apart, the information density of image classes is merely sufficient to serve as validation source.

Fig. 4(b) shows the probability diagram between starting position and milestone 4 with the integration of image classification results. Although, due to the excessive $h_{t,0}$-reward, the correct estimate's likelikood sometimes drops down to near-zero, it never falls near residual probability. Thus, confirmations of wrong position hypotheses are never able to threaten the correct hypothesis' domination, i. e. the erroneous hypothesis switch has been avoided this time. Additionally, average $h_{t,0}$-probability is reduced significantly over the whole course.

VI Conclusions

In this paper we have proposed a hybrid topological/metric navigation architecture for mobile service robots, which preserves the basic advantages of both world models, e. g. local precision, simple map building and path planning, etc. . Based on this architecture, we furthermore presented a global self-localization technique, which solves the bootstrap-problem and is able to recover from arbitrary position errors, while at the same time being open to the integration of arbitrary information sources. Experiments with extreme parameter settings in a partly monotonous environment have proven the method's robustness.

In the senor domain, future work concerns the development of additional validation and primary sources, especially in the challenging field of computer vision and image understanding. Regarding map construction, the currently used mapping procedure will be replaced by a probabilistic approach inspired by [13].

References

[1] W. Burgard, D. Fox, D. Hennig, T. Schmidt: *Position Tracking with Position Probability Grids*, Proceedings 1st Euromicro Workshop on Advanced Mobile Robots (EUROBOT), 1996

[2] D. Fox, W. Burgard, F. Dellaert, S. Thrun: *Monte Carlo Localization: Efficient Position Estimation for Mobile Robots*, Proceedings 16th National Conference on Artificial Intelligence (AAAI), 1999

[3] R. Simmons, S. Koenig: *Probabilistic Navigation in Partially Observable Environments*, Proceedings International Joint Conference on Artificial Intelligence (IJCAI), 1995

[4] D. Kortenkamp, T. Weymouth: *Topological mapping for mobile robots using a combination of sonar and vision sensing*, Proceedings 12th National Conference on Artificial Intelligence (AAAI), 1994

[5] G. v. Wichert, K. Kleiner: *Selbstorganisierende Bildanalyse für die Navigation von mobilen Robotern*, Autonome Mobile Systeme (AMS), Springer, 1996

[6] J. Weber, K.-W. Jörg, E. v. Puttkamer: *APR - Global Scan Matching Using Anchor Point Relationships*, Proceedings Intelligent Autonomous Systems (IAS-6), 2000

[7] J. Weber, L. Franken, K.-W. Jörg, E. v. Puttkamer: *Globale Selbstlokalisation mittels Referenzscan-Matching*, Autonome Mobile Systeme (AMS), Springer, 2000

[8] J. J. Leonard, H. F. Durrant-Whyte: *Mobile Robot Localization by Tracking Geometric Beacons*, IEEE Transactions on Robotics & Automation, 7(6), 1991

[9] K. O. Arras, S. J. Vestli: *Hybrid, High-Precision Localisation for the Mail Distributing Mobile Robot System MOPS*, Proceedings IEEE International Conference on Robotics & Automation (ICRA), 1998

[10] T. Edlinger, G. Weiß: *Exploration, Navigation and Self-Localization in an Autonomous Mobile Robot*, Autonome Mobile Systeme (AMS), 1995

[11] J.-S. Gutmann, K. Konolige: *Incremental Mapping of Large Cyclic Environments*, International Symposium on Computational Intelligence in Robotics and Automation (CIRA), 1999

[12] F. Lu, E. Milios: *Globally Consistent Range Scan Alignment for Environment Mapping*, Autonomous Robots 4: 333-349, 1997

[13] S. Thrun, W. Burgard, D. Fox: *A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots*, Machine Learning and Autonomous Robots (joint issue), 31/5, 1998

[14] S. Thrun, J.-S. Gutmann, D. Fox, W. Burgard, B. J. Kuipers: *Integrating Topological and Metric Maps for Mobile Robot Navigation: A Statistical Approach*, Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1999

[15] F. Lu, E. E. Milios: *Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans*, Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1994