

Reference Scan Matching for Global Self-Localization

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

Computer Science Department, Robotics Research Group, University of Kaiserslautern, P. O. Box 3049, D-67653 Kaiserslautern, Germany

Abstract

Especially in dynamic environments a key feature concerning the robustness of mobile robot navigation is the capability of global self-localization. This term denotes a robot's ability to generate and evaluate position hypotheses independently from initial estimates, by these means providing the capacity to correct position errors of arbitrary scale. In the self-localization frame described in this article, the feature based APR scan matching algorithm provides for each new laser scan several possible alignments within a set of reference scans. The generation of alignments depends on similarities between the current scan and the reference scans, only, and is unconstrained by earlier estimates. The resulting multiple position hypotheses are tracked and evaluated in a hybrid topological/metric world model by a Bayesian approach. This probabilistic technique is especially designed to integrate position information from different sources, e. g. laserscanners, computer vision, etc. .

Keywords: global self-localization, mobile robot navigation, scan matching, service robots, probabilistic reasoning

1. INTRODUCTION

Robustness of mobile robot self-localization is determined by the navigation algorithm's capacities to recover from erroneous position estimates. In contrast to industrial applications, where human and machine working spaces are usually separated and the degree of environmental dynamics is controllable, service robots have to deal with populated, unprepared, possibly mutable and generally unpredictable environments. Due to temporary environmental changes, parts of the known world might become unrecognizable for the robot's external sensors. Thus, the robot has to rely on internal position sensors (e. g. odometry or inertial systems), accumulating an uncorrected position error while travelling without external revision. *Tracked* self-localization techniques (e. g. [1], [5]) are designed to *refine an existing* position estimate by external sensor observations within a limited tolerance, only. Hence, they are unable to recover from accumulated position errors beyond that tolerance. In order to overcome these deficiencies, *global* localization techniques (e. g. [6], [4], [11], [9]) create and by probabilistic means evaluate a multitude of position hypotheses in parallel. If the robot loses the correct position in an unrecognizable region, global self-localization will sooner or later award the correct position with the maximum likelihood after having returned to recognizable areas, again. However, since global techniques face a significantly bigger search space than tracked localization methods, they are generally less precise in order to still fit realtime requirements.

Closely connected to the question of how to design global localization algorithms is the preferred environment representation (*world model*), which itself is determined by the available sensors and the intended robot application. Basically, two different types of environment representations can be identified: *metric* and *topological* world models. A metric environment map, storing obstacles and features in a unique coordinate system, is a straight-forward and easy-to-implement approach when rangefinders like laserscanners and ultrasonic sensors are used. However, metric maps are not well suited to integrate non-metric information as required by many pattern matchers (e. g. the neural image classifier introduced by Wichert [16]). Furthermore, metrically consistent map building is a non-trivial problem, currently being under intensive research (e. g. [8], [12]). Global self-localization techniques based on metric world models are [6], [2], [4].

Topological representations, instead, model the environment as graphs of nodes (distinctive places) and connecting edges (pathways). The basic advantages of topological world models compared to metric ones are:

- simplified map building, since only topological, not precise metric consistency has to be achieved,

- path planning is reduced to direct graph search,
- non-metric and metric sensor data is uniformly treated as attributes to nodes and/or edges, which simplifies sensor fusion.

However, topological world models suffer from the poor environment resolution, which is insufficient for any other purpose than navigating from one region to another. Examples for global self-localization based on topological maps are [11], [9].

Whenever metric and non-metric localization techniques are mixed, the use of a *hybrid* topological/metric world model is adequate to benefit from the advantages of both environment representations, while at the same time avoiding their inherent restrictions.

In the CAROL-project (Camera Based Adaptive Robot Navigation and Learning), which serves as a framework for the development of global self-localization techniques and navigation architectures, various metric or appearance-based algorithms for processing laserscanner and vision data have been developed and implemented. Consequently, the natural base to fuse these different techniques is a hybrid world model, which is explained in detail in section 3. One of the fundamental localization methods, the APR scan matching approach described in section 2 is by itself some sort of hybrid, since the idea of recognizing a certain region by a stored sample scan is essentially topological. The APR recognition process, however, results in metric alignment information of current and reference scan. The probabilistic method that tracks and evaluates multiple position hypotheses, integrating results from different sensors, is treated in section 4.

2. APR SCAN MATCHING

In the past, various scan matching techniques have been proposed for different kinds of applications. These approaches can be separated into two main categories:

Feature based matching techniques either align extracted geometric primitives or raw range readings with an existing structural description of the environment. These matching techniques represent an efficient, reliable and popular class of self-localization methods for polygonal environments (e. g. [1], [3]). However, feature based techniques suffer from the poor assortment of practically usable geometric primitives, thus mainly line segments and derived landmarks, e. g. corners and door openings are used.

In recent years a second category of more flexible *raw data matching* techniques without explicit geometric interpretation have been developed (e. g. [4], [5], [10]). Unfortunately, most of these scan matching approaches like [5], [10] are designed to refine an existing position estimate, hence are not suitable for global self-localization. On the other hand, the global approach presented by Crowley, Wallner et al. [4] directly integrates the matching and position tracking technique in a strict metric world model. Especially the close integration of matching and tracking technique render the fusion of localization information from different sensors difficult.

The *Anchor Point Relation (APR)* scan matching technique [13] treated in this section supports global localization by the realtime search of matches for an actual laser scan in a given set of reference scans. The algorithm's output in each processing cycle are *multiple, weighted* matching hypotheses. The quality of every single matching alternative is computed depending on the degree of congruence between the raw data of current and reference scan, with respect to the hypothetical alignment of both scans. This quality measure is especially important in probabilistic self-localization architectures.

Since there is no explicit geometrical interpretation by the extraction of structural primitives, APR avoids the typical information loss associated with the concentration on lines and clusters, only. However, in order to be realtime capable, the alignment of two laser scans is computed depending on the geometrical patterns formed by extracted feature points (*anchor points, APs*). In environments which don't provide a sufficient number of these features, alignments can't be determined.

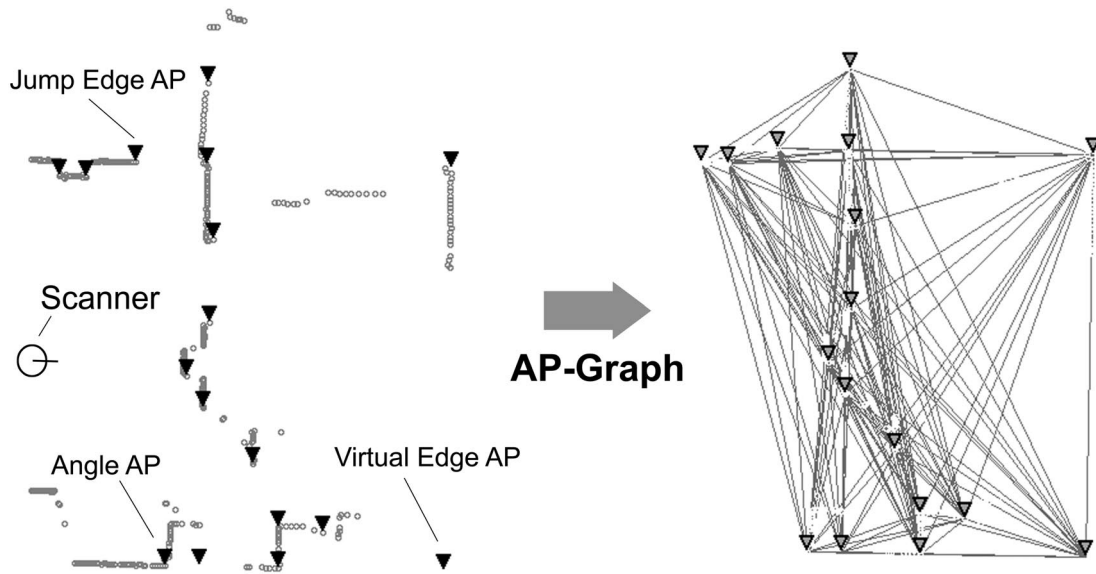


Fig. 1. Laserscan and extracted anchor points (left) and resulting fully connected anchor point graph (right)

A. Anchor Point Extraction

In our current implementation, three types of anchor points are used: *Jump Edge*, *Angle* and *Virtual Edge* anchor points. The extraction steps are rather technical and therefore mainly skipped in this article. Details concerning the APR anchor point extraction can be found in [13].

Jump Edge Anchors and *Angle Anchors* both refer to object edges which intersect the scanning plane. Jump edges refer to convex edges where only one of the enclosing object faces is visible from the scanning point, and are detected basically by a jump of consecutive range readings. Angle Anchors refer to convex or concave edges when both respective object faces are visible. Briefly described, they can be detected where the range values on a scanned object are continuous but the object's surface angles are discontinuous. In order to concentrate on objects of a minimum size for localization purposes, small objects (e. g. chairs, wastebaskets, etc.) are removed from further consideration by executing a segmentation step, first. This segmentation step finds large connected parts in the range scan, and jump edge and angle anchors are searched in these segments, only.

Virtual Edge Anchors are intersection points of the elongations of large straight objects' surfaces (e. g. walls, lockers, etc.). In the most simple case, the corners of a room are extracted as virtual edge anchors, even if the corner itself is hidden by some object. Please note, that the intersections are found without explicit line extraction. Instead, geometrical statistics is used to determine the large objects' positions and orientations, which offers higher robustness against partial occlusion and noise effects.

The basis for the recognition of a scan in the APR matching is its geometrical anchor point pattern, which is mathematically given by the fully connected AP graph (see figure 1).

B. Preselection

Usually, two scans of the same scenery will result in two disjoint sets of anchor points A and A' due to three main sources of error:

- occlusion of significant environment parts by objects which were not present at creation time of the corresponding reference scan (*dynamic environments*),

- fluctuations in position extraction of anchor points due to measurement errors and aliasing effects,
- only partial overlap of the two corresponding 180°-scans due to different scan positions and orientations.

The matching problem is thus reduced to the identification of maximum matching subgraphs in the set of all reference graphs. In order to avoid a full search in the reference set, APR performs a preselection of likely candidates using the *relationship database*. The database search tries to identify reference graphs with a similar spatial anchor point configuration by statistic means. In office environments usually a high number of similar distances between single pairs of APs will occur (due to standardized furniture, identical door sizes, etc.). However, only the same or mirror-symmetrical configuration of objects will produce a similar, fully connected graph and thus a high number of corresponding edge lengths.

The relationship database is constructed from all edges of all reference graphs. Each edge $e_i=(p_a, p_b, d_i)$ of a reference graph G_n is represented in the database as a tuple (n, e_i) . Both anchor points p_a and p_b are nodes in G_n and d_i is the euclidian distance between anchor point p_a and p_b . The database is implemented as an index table, sorted according to distances d_i .

The preselection of probable matching candidates in the reference set for graph G' of the current scan is done by a search for distances d_i in the relationship database for all $e_i \in G'$. For each edge entry (m, e_j) in the database, which corresponds in length to an $e_i \in G'$, a counter c_m for the respective reference graph G_m is incremented. Thus, after the completed search of all edges of G' in the database, a counter c_m contains the number of edge length correspondences of reference graph G_m and G' . The preselection quality q_m for each reference graph G_m is then computed as:

$$q_m = \frac{c_m^2}{k_{act} \cdot k_m}$$

where k_{act} and k_m are the total numbers of edges in G' and G_m , respectively. Only reference scans with a high preselection value q_m are chosen for further processing: the alignment step. The q_m measure's construction prevents 'big' reference graphs with many edges of becoming 'more attractive' only due to a resulting high number of accidental correspondences.

C. Alignment

The alignment step aims at finding a coordinate transformation to align one anchor point graph G_1 (e.g. a reference graph) with another graph G_2 (e.g. the AP-graph of the actual laser scan). This not only provides the possibility to check the validity of the match by some means of raw data correlation; in the case of a correct match it directly delivers the actual robot position in the reference graph's local coordinate system.

Each length correspondence of two edges $e_1 \in G_1$ and $e_2 \in G_2$ provides two possibilities for the alignment of both graphs. For $e_1=(p_a, p_b, d_1)$ and $e_2=(p_v, p_w, d_2)$, with $d_1 \approx d_2$, the combinations (p_a, p_v) and (p_b, p_w) might represent the same anchor points in the two graphs, or (p_a, p_w) and (p_b, p_v) , or none of both if the correspondence is just accidental.

The alignment decision is computed statistically by creating a $n \times m$ AP correspondence matrix (Fig. 2), where n and m are the total numbers of APs in G_1 and G_2 , respectively. For each of the above mentioned four possible AP combinations (p_i, p_j) per edge correspondence, the respective matrix element (i, j) is incremented by 1. The matrix thus counts the possible anchor point identifications indicated by the edge correspondences as a sort of two-dimensional histogram.

In the example of Fig. 2, G_2 is a true subgraph of of the larger G_1 , but with a different AP enumeration. In G_1 the edges (1, 2) and (2, 4) have the same length (i.e. the same length than (2, 3) in G_2), all other edges have different lengths. The resulting correspondence matrix achieves a maximum in line 2, column 3, which identifies the pair AP 2 of G_1 and AP 3 of G_2 as the most likely AP correspondence.

APR now uses the edge correspondences which have contributed to this maximum entry to compute an equal number of geometric alignment hypotheses. All correct contributions to that peak will produce similar hypotheses,

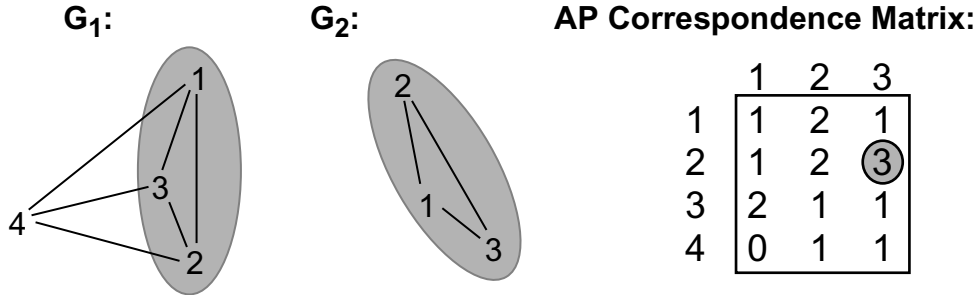


Fig. 2. Alignment of two AP graphs using a correspondence matrix

erroneous contributions are eliminated by a clustering step. The remaining transformations are unified by computing the center of gravity to provide a more precise alignment.

As can be seen in the example of Fig. 2, the 'noise-to-signal-distance' in the correspondence matrix is not satisfactory for small total numbers of APs or poor overlap of G_1 and G_2 . In order to eliminate coincidental edge correspondences, like (2, 4) in G_1 with (2, 3) in G_2 , APR determines the most likely rotational difference between G_1 and G_2 from a *matching angle histogram*. This histogram results from a frequency analysis of the angle differences between all pairs of corresponding edges in G_1 and G_2 . All edge correspondences not belonging to the histogram peak are ruled out from further consideration in the AP correspondence matrix. In Fig. 2, the false edge correspondence of (2, 4) in G_1 with (2, 3) in G_2 will be eliminated because three correct edge correspondences indicate a rotational difference of 30° between the two graphs, whereas the false edge pair suggests an angle of -45° (modulo 180°).

Although the matrix technique is not recommendable for general graph matching problems with high node numbers, it is very efficient and, by its statistical nature, highly robust against AP extraction errors for typical graph sizes in the APR application scenario.

D. Evaluation

APR performs the alignment step of the actual scan for a selected number of reference scans with the highest q_m -values. The evaluation step provides a quality measure e_m for the use in a probabilistic self-localization architecture.

Using 180° laser scans bears the problem of different perspectives: generally, scan A contains environmental information which scan B doesn't, and vice versa. Additionally, range data acquired from different positions, but representing the same part of the environment, can't be correlated directly.

This problem is solved by creating *synthetic scans*: firstly, the relative translation of scans A and B is checked to determine which scan contains the other scan's sensing position and consequently has a better 'overview'. From this scan's range data an artificial scan for the other scan's (*front scan*) position and orientation is computed. This synthetic range image can be directly correlated with the front scan to determine the degree of congruence achieved after alignment.

E. Experimental Results

Fig. 3 shows a set of 30 reference scans of our lab environment, fitted into a global coordinate system. A series of 1405 scans recorded during a second, independent test run was matched with this reference set. The SICK scanner is mounted at a height of approx. 25 cm, thus the algorithm has to cope with a cluttered field of 'vision' due to chair and table legs, waste paper baskets, etc. For each scan matching cycle, 5 reference graphs with the highest preselection values were checked for correspondences with the current laser scan. All scan matches were *global*,

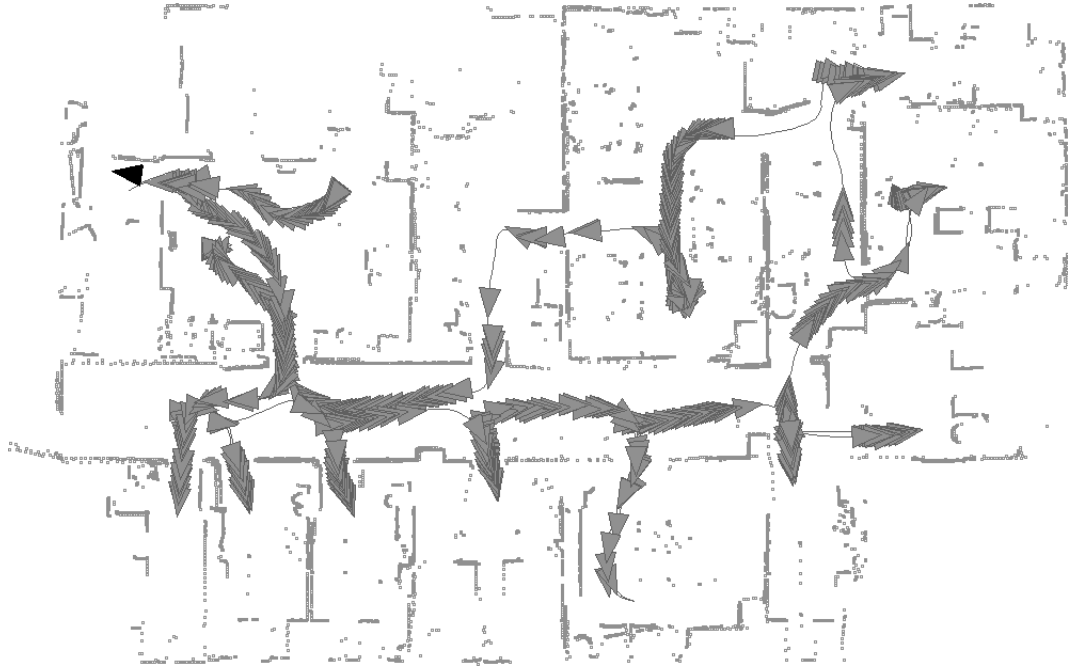


Fig. 3. Global APR scan matching results. Correct matches are indicated by gray triangles, the false match is marked by a black triangle

i.e. no a priori available position information was used to reduce search space.

In Fig. 3, the APR matching results are marked by triangles in the map. For each of the 1405 matching cycles only the alignment result with the highest evaluation rating e_m is displayed. Correct position estimates are colored gray, wrong estimates are colored black. A line between gray triangles indicates a track section without matching result, i.e. at these positions the evaluation value e_m of the best alignment result was below a certain threshold. In total, 613 correct global position estimates and 1 wrong estimate were generated, which shows a high hit/error ratio. Average scan processing time including all preprocessing steps was below 15 msec with a maximum of 37 msec per cycle on a 233 MHz Pentium PC.

If only the reference scan with the highest preselection value q_m is aligned with the current scan, the number of correct matches decreases to 374 with no false matching. If all reference scans are checked in each cycle, 831 correct matches and 1 false are produced with a still acceptable computing time of 57.4 msec (avg.) and 119 msec (max.).

Please note, that the procedure of transforming local alignment results into a global coordinate frame corresponds to global self-localization in a *metric* navigation scheme. At the same time, APR performs well in identifying the *topologically* correct reference scans. Environmental ambiguities are handled by providing multiple different, reasonably quantified hypotheses. This is especially of interest for the probabilistic self-localization technique described in section 4.

3. WORLD MODEL

In industrial applications, a mobile robot often requires its precise position in a unique coordinate system. This is especially necessary for precise track following on defined pathways and global movement coordination of multiple robots. In these applications, the environment can generally be presumed as cooperative. So are employees in

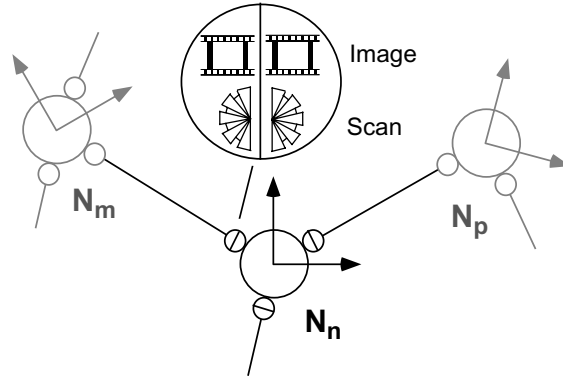


Fig. 4. Local coordinate systems in the hybrid world model and bidirectional storage of sensor data

an automated production plant instructed to keep clear of a moving robot's path to avoid work flow disturbances. In the service domain, robots have to operate in unprepared and populated areas, e. g. office rooms, supermarkets and museums. Persons in the robot's working space, e. g. shoppers in a supermarket, are not necessarily cooperative. The use of predefined pathways and global movement coordination of a robot fleet is too inflexible to cope with that kind of unpredictable and uncooperative environment. Hence, the necessity of a global metric coordinate system in many service applications is questionable. However, *local* metric precision is still often required, e.g. for cleaning path planning, station docking, etc., which rules out a purely topological approach. Furthermore, in order to profit from the geometrical localization data of the APR alignment results or other metric localization methods, the world model has to have some metric component.

Consequently, we argue to use a topological graph as the backbone of environment representation, while at the same time establishing local metric coordinate systems in the nodes' regions (Fig. 4). This preserves the advantages of local precision, simple data fusion and easy path planning, while eliminating the necessity for the expensive and difficult establishment of global metric map correctness. In our scheme, the approximate relative positions and orientations of the local coordinate systems of two topological neighbours are stored in the connecting arc. Thus, a method, which is tracking the robot's position from one local system to the next, has to deal with small errors, only. Lacking a global reference system, the basic difficulty during map building is the decision when to close the connection between two already existing nodes and to determine their local coordinate systems' relative position and orientation. This problem is solved by APR's and other global localization techniques' abilities to recognize distinctive places of the environment.

There is no explicit global reference system defined in the map, although such a common coordinate system can be assembled from the particular metric neighbour information. However, due to the summation of a multitude of minor errors between single nodes, this system will be contorted and 'torn' in those places, where its successive construction closes a topological cycle. During map construction, however, even such an inconsistent global coordinate system can provide a raw idea about the likelihood of a closed cycle, indicated by APR or another technique.

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 (180°-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.

4. PROBABILISTIC POSITION TRACKING

In classic *tracked* self-localization approaches (e. g. [1], [5], [10]) one robot position estimate is observed, only,

and new sensor data is merely used to proceed and optimize that position belief in space. These Kalman filter or correlation techniques generally provide high precision in metric world models. However, since being *unimodal*, these techniques are able to compensate limited position errors, only. Thus, a position error accumulated in a temporarily unrecognizable environment part might be beyond the algorithm's error recovery capabilities, causing the robot to permanently lose its correct position estimate.

In contrast, *global* self-localization has to be *multimodal*, which denotes the ability to track multiple hypotheses in parallel. A common approach (e. g. in [6], [2], [11], [9]) 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. The decision how to discretize the position probability distribution is essentially influenced by the underlying world model. In topological representations (e. g. [11], [9]) 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 introduced by Burgard et al. [2] the volume of 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 grid cube is computed depending on the match of current range data and map contents. Due to its large computational expense, this method has been superseded by a more efficient sample-based method presented by Fox et al. [6].

As mentioned earlier, the purely topological approaches don't provide the desired resolution of position estimate. The metric techniques mentioned (including the eigenspace method by Crowley et al. [4]), on the other hand, presuppose the existence of a consistent metric map, since there is no high-level processing of sensor data which would allow a perspective-independent recognition of distinctive places. Instead, they rely on the map's property to provide for each possible robot position and orientation a set of range data which is directly comparable to the current sensor readings. An inconsistently assembled map is a handicap for this ability in significant parts of the environment. Furthermore, none of the above mentioned metric concepts is really prepared to integrate information apart from range data. However, a great advantage of all of these approaches is their principal ability to cope with arbitrary environmental structures.

The position tracking mechanism developed in the CAROL project, like the topological methods of Simmons&Koenig [11] and Kortenkamp&Weymouth [9] directly supports the fusion of sensor information from different sources, thus providing more robustness by a higher degree of environment information. On the other hand, our technique provides precise position estimates like the metric techniques [6], [2] and [4]. However, due to the renunciation of a unique coordinate system, these estimates refer to local coordinates in the nodes' regions, only. Due to the renunciation of a consistent and complete metric map, the algorithm won't be able to provide the environment's appearance for every possible position and orientation. Instead, our tracking algorithm has to rely on the applied localization techniques' (e. g. APR) abilities to generate independent position hypotheses. In contrast to the global metric approaches [6], [2] and [4], this principally limits the possible target environments to those whose characteristics can be handled (i. e. recognized) by the sum of the applied pattern matchers. In order to increase respective flexibility, self-organizing pattern matchers which adapt themselves to the environmental characteristics, have been implemented (e. g. the neural image classifier proposed by Wichert [16]) and developed (e. g. neural scan classification [14]) in the CAROL project.

The following description of the probabilistic tracking technique is an outline of the basic idea. The mechanism has been formally described in [15].

A. Position Tracking

With respect to a node's local coordinate system, we define an associated *local 3D configuration space* for the possible robot positions (translation x , y and orientation ϕ) in this node's region. The union of all local configuration spaces forms the non-consistent *super space*, in analogy to the assembled, inconsistent global coordinate system. We furthermore assume the existence of at least one *primary source* of localization information, at least one

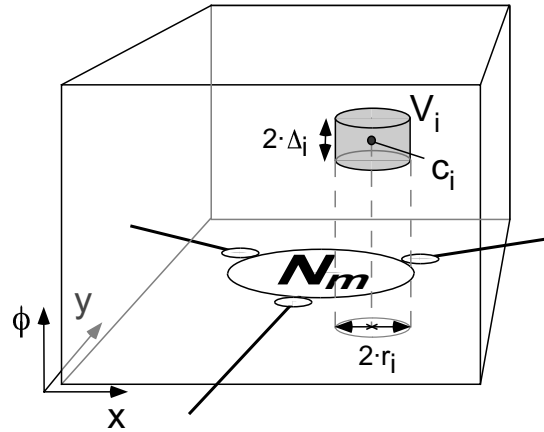


Fig. 5. Local configuration space and position hypothesis

tracking sensor and possibly further *validation sources*.

Primary sources, like APR, are localization techniques which generate global position estimates independently of earlier assumptions. To stay with the APR example, all alignment alternatives for the current laserscan exceeding a certain quality threshold are used to generate or confirm so-called *explicit position hypotheses*. These data structures discretize the robot's current position probability distribution in the super space. Each position hypothesis i is defined by a volume V_i in a certain local configuration space and a probability value λ_i , indicating the likelihood of the current robot position actually being inside V_i (Fig. 5). The volume V_i is given by a center c_i with an uncertainty region defined by a radius r_i of translational tolerance and the angle tolerance Δ_i . The λ_i 's have to sum up to 1 together with the value of a special 'rest' hypothesis. This value represents the likelihood that the actual robot position is *not* represented by one of the explicit hypotheses.

First step after receiving new valid APR results is the check whether their geometrical position estimates *confirm* existing position hypotheses, i. e. if any of APR's position alternatives lies in one of the V_i 's. Since the robot might currently be heading out of one node's context into another, all position hypotheses have to be checked. This is done in the super space with careful attention to its known contortions. If an existing position hypothesis is confirmed by an APR result, it changes to the configuration space of the respective reference scan's 'home' node, the center c_i is set to the APR position estimate, the uncertainty region is reset to a small default value, and probability λ_i is rewarded according to the confirming APR result's quality. APR results which *don't* confirm already existing hypotheses are used to create new ones in the respective local configuration spaces with an initially low λ_i -likelihood. The probabilities of unconfirmed position hypotheses are „punished“ and hypotheses with λ_i -values below a certain threshold are deleted.

At this point the question is still open how to modify the probability of the 'rest' hypothesis. Since this value should reflect the system's 'skepticism' against the explicit position hypotheses, it is rewarded or punished periodically, depending on whether or not existing hypotheses have been confirmed by sensor data.

Validation data sources, in contrast to primary sources, are inadequate to create new hypotheses. This might for example result from high computational requirements, which don't allow a global search (e. g. the iterative IDC algorithm [10] which requires a good starting point). However, they are able to provide useful information to confirm or reject existing position hypotheses and to refine a c_i 's local precision. Confirmations from validation sources are treated exactly like those from primary sources.

Tracking sensors, e. g. odometry or inertial systems, are used to move all position hypotheses (i. e. to modify their c_i 's) according to the measured relative movement since creation or last confirmation, respectively. The tracking sensors are important to bridge passages in unrecognizable environment parts. Since they don't provide any

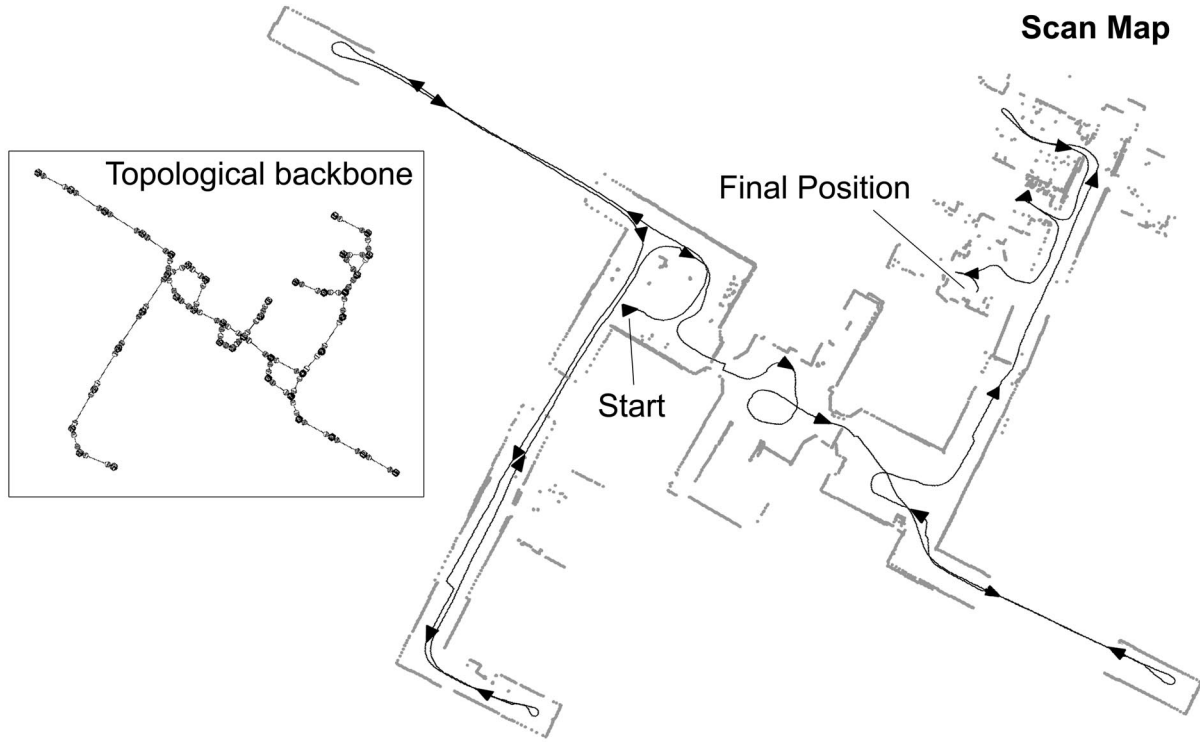


Fig. 6. Topological map of test environment with APR reference scans; test course for self-localization test

information about a hypothesis' likelihood, the λ_i -proportions are unaffected by tracking sensors. However, since their data is generally erroneous, the growing position uncertainty is reflected by expanding the V_i -volumes.

B. Experimental Results

In the experimental setup, APR serves as the primary, odometry as tracking and a neural image classification technique [16] as validation source. The image classification is performed by a growing neural gas net [7] structure created by a sample set of environment images. In this example, the image classifier represents the sample set through 150 reference vectors (neurons), enabling the algorithm to distinguish between 150 different classes of visual scenes. Fig. 6 shows the topological backbone of the environment map and the 53 APR reference scans stored in the graph. They are assembled in the inconsistent global coordinate system (size approx. 50x70 m) for visualization purposes. Additionally to the reference scans, image scene classes have been registered in the topological graph during map building. The path indicated in the scan map is the robot's test course.

In order to give an impression of the creation and tracking of position hypotheses, the hypotheses c_i 's are visualized in Fig. 7 for the first 30 meters of the robot's test course. The hypotheses c_i 's are indicated by triangles, the strongest hypothesis is marked by a circle. From the first APR results on, the correct position estimate dominates the other explicit hypotheses. (In Fig. 7 λ_b indicates the probability of the best, λ_0 the probability of the 'rest' hypothesis.) 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 had been detectable, no reference scans were inserted into the topological graph. Although APR doesn't work in these track sections, the correct position hypothesis receives confirmations from the image classification. Thus, it remains on a high relative probability level, reinforcing the correct position estimate even in monotonous environment parts until reaching the final position.

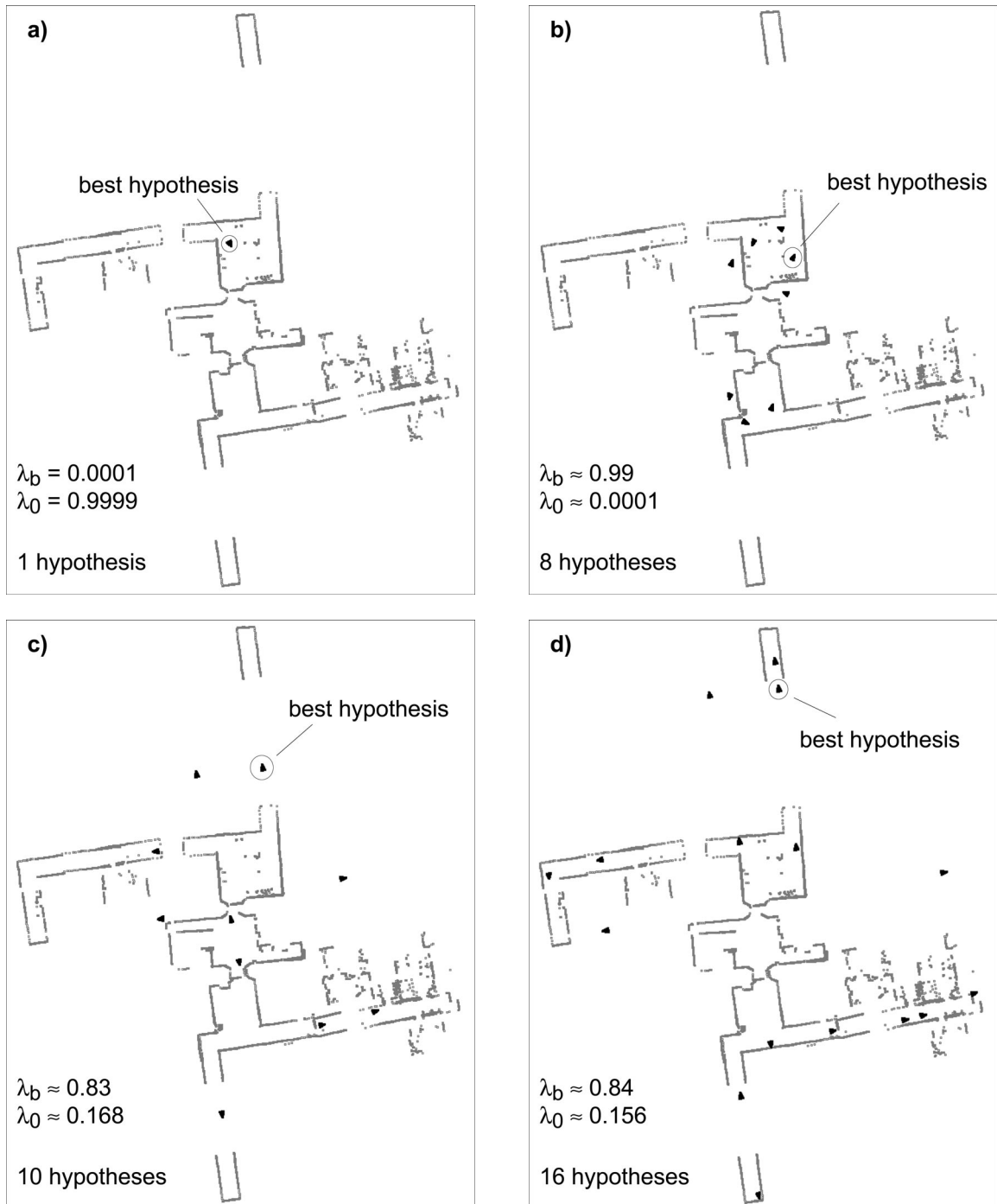


Fig. 7. Position hypotheses (black triangles) after first APR matching cycle (a), 32 cycles (b), 75 cycles (c) and 106 cycles (d)

References

- [1] 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), Leuven, Belgium, 1998
- [2] W. Burgard, D. Fox, D. Hennig, T. Schmidt: *Position Tracking with Position Probability Grids*, Proceedings 1st Euromicro Workshop on Advanced Mobile Robots (EUROBOT), Kaiserslautern, Germany, 1996
- [3] I. J. Cox: *Blanche - An Experiment in Guidance and Navigation of an Autonomous Robot Vehicle*, IEEE Transactions on Robotics and Automation, Vol 7 (2), 1991
- [4] J. L. Crowley, F. Wallner, B. Schiele: *Position Estimates Using Principal Components of Range Data*, Proceedings IEEE International Conference on Robotics & Automation, Leuven, Belgium, 1998
- [5] T. Edlinger, G. Weiß: *Exploration, Navigation and Self-Localization in an Autonomous Mobile Robot*, Autonome Mobile Systeme (AMS), Karlsruhe, Germany, 1995
- [6] 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), Orlando, Florida, USA, 1999
- [7] B. Fritzke: *A growing neural gas network learns topologies*, Advances in Neural Information Processing, 7, 1995
- [8] J.-S. Gutmann, K. Konolige: *Incremental Mapping of Large Cyclic Environments*, International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Monterey, California, USA, 1999
- [9] 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), Seattle, Washington, USA, 1994
- [10] 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), Seattle, Washington, USA, 1994
- [11] R. Simmons, S. Koenig: *Probabilistic Navigation in Partially Observable Environments*, Proceedings International Joint Conference on Artificial Intelligence (IJCAI), Montreal, Canada, 1995
- [12] 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
- [13] J. Weber, K.-W. Jörg, E. v. Puttkamer: *APR - Global Scan Matching Using Anchor Point Relationships*, Proceedings Intelligent Autonomous Systems (IAS-6), Venice, Italy, 2000
- [14] J. Weber, A. Spieß, K.-W. Jörg, E. v. Puttkamer: *Selbstorganisierende Klassifikation von 2D-Laserscans zur Navigation eines AMR*, Autonome Mobile Systeme (AMS), Karlsruhe, Germany, 2000
- [15] J. Weber, L. Franken, K.-W. Jörg, K. Schmitt, E. v. Puttkamer: *An Integrative Framework for Global Self-Localization*, Proceedings IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Baden-Baden, Germany, 2001
- [16] G. v. Wichert: *Selforganizing Visual Perception for Mobile Robot Navigation*, Proceedings 1st Euromicro Workshop on Advanced Mobile Robots (EUROBOT), Kaiserslautern, Germany, 1996