

APR - Global Scan Matching Using Anchor Point Relationships

Joachim Weber, Klaus-Werner Jörg, Ewald von Puttkamer

Computer Science Department • Robotics Research Group

Kaiserslautern University, PO Box 3049, D-67653 Kaiserslautern, Germany

email: {jweber, joerg, puttkam}@informatik.uni-kl.de

Abstract. Global self-localization, i.e. the ability to generate position estimates without initial hypotheses, decisively improves robustness of mobile robot localization since it allows recovery from arbitrary position errors. APR is a pattern matching algorithm designed for the realtime search of best matching laser scans in a set of given reference scans. The algorithm's output is a number of weighted hypotheses which makes APR especially attractive for probabilistic techniques aiming at global localization capabilities. The concept of reference scans makes APR applicable in both, topological and metric-map navigation schemes. Experimental results are presented for the matching of 180° laser scans in an office environment.

1 Introduction

Robustness of mobile robot self-localization in dynamic environments is primarily determined by the navigation algorithm's capabilities to recover from erroneous position estimates. With respect to these abilities, self-localization techniques can be divided into *tracked* and *global* localization techniques. Tracked self-localization (e.g. [13]) depends on an initial position information and can compensate limited position errors, only. Especially in dynamic environments, these error limitations usually can't be guaranteed and recovery from such position errors is rare.

Global self-localization, by observing a multitude of different position hypotheses, usually doesn't require initial position information and is generally able to recover from arbitrary position errors - as long as the target area is not completely symmetrical and unique environmental features can be detected in order to resolve ambiguities.

Closely connected to the question of how to design global localization algorithms for a certain sensor domain is the preferred environment representation (*world model*), which itself is often determined by the intended robot application. Due to their easy implementation and efficient handling, metric representations are an obvious choice when ranging devices like laserscanners or ultrasonic sensors are used. State-of-the-art approaches for global self-localization in the laserscanner domain for metric world models are [2, 11].

Topological maps, which represent the environment as a network of distinctive places and connecting pathways seem to be more adequate when non-metric information has to be integrated into the world model [3, 5]. In combination with probabilistic techniques they are well suited to handle environmental ambiguities which are common in service robot workspaces, e.g. office buildings [4, 14].

Against this background our CAROL-project (**C**amera Based **A**daptive **R**obot Navigation and **L**earning) serves as a framework for the development of global self-localization

techniques and navigation architectures based on topological maps. In order to perform camera-based navigation, we have implemented a modified version of the neural scene classifier described in [5]. Another algorithm detects and decodes bar code labels from camera images [1] as a machine equivalent to the human ability to read door plates and road signs. Besides the camera, other sources of localization information are 180° laser scans provided by a SICK LMS-200, which are used to identify most likely matches in a set of reference scans. The latter is subject of this paper.

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 [6, 7]. 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 openings are used.

In recent years a second category of *raw data matching* techniques without explicit geometric interpretation have been developed. Lu and Milios [8] present an approach for the matching of nearby scans, suitable for a *tracked robot pose estimation*, which takes the correctness of the last position hypothesis for granted. Weiss et. al. [9] suggest a scan correlation algorithm which is more robust against translatorial and rotational differences than [8] but primarily designed for polygonal environments. Crowley, Wallner et. al. developed a matching technique based on principal component analysis suitable for global self-localization [2]. However, this method requires extensive training sets from known positions.

The *Anchor Point Relation matching (APR)* presented in this paper supports global localization by searching matches for an actual laser scan in a given set of reference scans in realtime. The algorithm's output is a certain number of weighted hypotheses, which makes APR especially attractive for probabilistic navigation techniques. Although the reference scan concept resembles topological approaches in so far that a certain part of the area is described by a representable scan, it is also applicable in navigation schemes based on metric models, as will be shown in the experimental results.

Since there is no explicit geometrical interpretation by extraction of primitives, APR realizes a hybrid scan matching technique. This prevents the typical information loss associated with the concentration on line segments, only. Instead, relations between characteristic coordinates of object properties (*anchor points*) are used as heuristics to choose likely candidates out of a set of reference scans to avoid a computationally expensive correspondence search in the whole set.

The rest of the paper is organized as follows: section 2 shortly addresses the angle function as an important tool in different steps of the algorithm, while section 3 treats the extraction of anchor points. The matching technique itself is presented in section 4 while the final verification phase is discussed in section 5. Section 6 offers experimental results.

2 Angle Function

For the purpose of scan analysis it is important to know the distribution of range readings relative to each other rather than relative to the scanner's position. Consequently, Hinkel and Knieriemen [10] propose an analysis method which computes angles of lines (modulo 180°) between succeeding range readings with respect to the scan coordinate system (*angle function*). In a second step the discrete angle frequency distribution (*angle histo-*

gram) of this function is used to determine the robot's orientation relative to the surrounding walls.

However, the angle function itself is a very useful tool to determine geometric properties of surfaces. Fig. 1 shows a laser scan and the corresponding angle function. The curved object part appears as a linear ramp while the planar surfaces are represented as sections of constant angles. As a compromise between information yield and noise reduction Hinkel and Knieriemen recommend a constant offset of several degrees between two range readings instead of taking immediate neighbours to calculate the angles. In the current APR implementation, instead, the angle offsets are chosen individually depending on the measured distances in order to achieve similar metric distances for the point pairs rather than a constant angle offset. This improves information yield while minimizing noise effects at the same time.

3 Anchor Point Extraction

The basic idea of the APR algorithm is the matching of two sets of characteristic 2D coordinates which are reproducible object feature positions (*anchor points*). Currently, three types of anchor points are used: *jump edge*, *angle* and *virtual edge* anchor points.

The observation that large objects tend to be the invariants of an environment is trivial but basic for place recognition. For APR this is taken into account by executing a segmentation step in order to find scan parts belonging to large surfaces, first. Consequently, those kinds of anchor points which represent the coordinates of object surface features (jump edge anchors and angle anchors, see below) are extracted for such segments, only.

Jump Edge Anchors. The easiest detectable anchor type corresponds to edges between two enclosing vertical surfaces when only one of the enclosing object faces is visible from the scanner's position. A jump edge is under these circumstances detectable as a jump in the range signal from near to far at the outer margin of a closed surface. In Fig. 1 a jump edge anchor has been extracted at the righthand part of the large object. To avoid misinterpreting *visibility rims* of round objects as jump edges, the offset between scanning angle and surface orientation angle of the segment margin is checked. At the visibility rim (Fig. 1) of the lefthand curved part the angle between the scanning ray and the surface orientation - provided by the angle function - is too small. Thus, no jump edge is extracted for this segment margin.

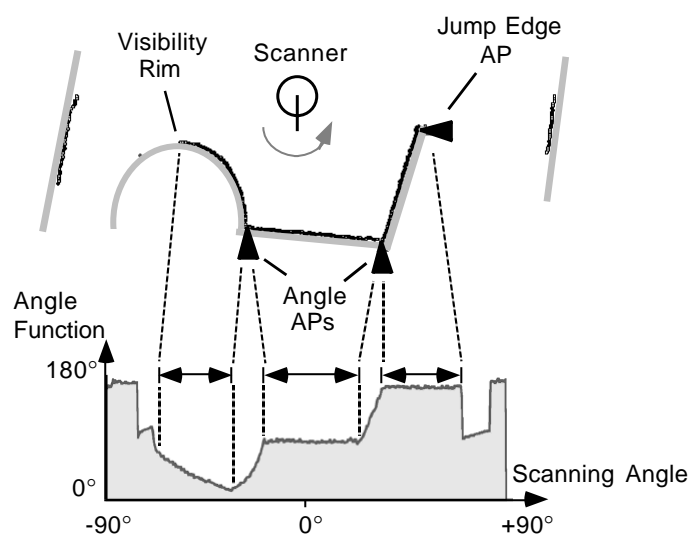


Fig. 1: LMS scan and resulting angle function; extracted anchor points (APs) are marked by triangles

Angle Anchors. When both enclosing object surfaces are visible, vertical edges appear as concurrent occurrence of a jump in the range signal's and a non-zero value in the angle function's 1st derivative (inner anchor points of complex object, Fig. 1). Please note, that convex angle anchors may transform into jump edge anchors and vice versa when seen from another perspective. Like jump edge anchors, angle anchors are extracted only if they are part of a connected segment.

Virtual Edge Anchors. In [9,10] the position of the angle histogram's maximum peak is interpreted as angle difference between the robot's orientation and the main angle of the surrounding walls. To determine the robot's translation with respect to these walls, x - and y -histograms are computed, i.e. discrete frequency analyses of the scan-point distributions in the direction of the walls' main and orthogonal angle, respectively. In the current APR implementation the peak positions of x - and y -histograms are used to create combined 2D positions (Fig. 2). These *virtual edge anchors* realize very stable and thus highly recognizable anchor points in orthogonal environments. Virtual edge anchors due to their statistic nature belong to large objects and don't require segmentation information like jump edge and angle anchors. As pointed out in [12] the concept of x - and y -histograms can be extended to arbitrary, non-orthogonal main angles by providing one point-distribution histogram per angle histogram peak.

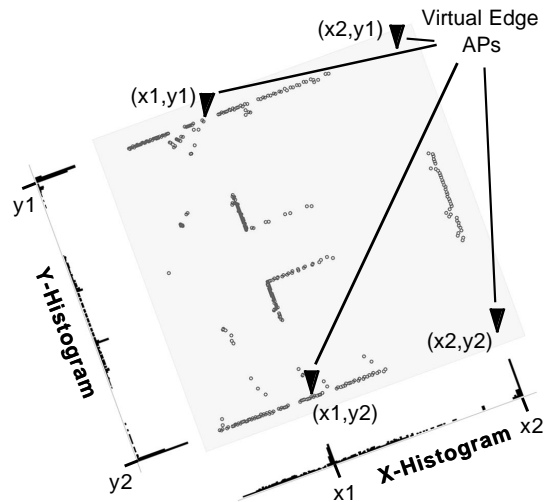


Fig. 2: Virtual anchor points as combined peak positions

4 Matching

Finding matches of actual laser scans in a set of reference scans without additional assumptions about current position and orientation is a computationally expensive operation. The basic idea of the APR algorithm is to represent both, reference scans and actual scan by fully connected graphs constructed from their sets of anchor points. Usually, two scans of the same scenery will result in two disjoint sets of anchor points A and A' due to four 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,
- different outcome of the segmentation step which affects jump edge and angle anchor point extraction,
- 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 sub-graphs in the set of all reference graphs.

4.1 Preselection - Candidate Search

In order to avoid a full search in the reference set, APR performs a preselection of likely candidates using the *relationship database*. This 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 vector $v=(n, e_i)$. Both anchor points p_a and p_b are nodes in G_n and d_i is the euclidian distance between node p_a and p_b . The database is implemented as a Hash table, sorted according to distances d_i .

Preselection of probable matching candidates in the reference set for a new graph G' is done by a search for distances d_i in the relationship database for all $e_i \in G'$. For each edge entry $v=(n, e_i)$ in the database which corresponds in length, a counter for G_n is incremented. After a completed search, a quality measure q_m is computed for each reference graph G_m :

$$q_m = \frac{n_m^2}{r \cdot s}$$

where n_m is the counted number of corresponding edges for G' and G_m . r and s are the total numbers of edges in G' and G_m , respectively. Only reference scans with a high q_m -value are chosen for further processing.

The database search thus tries to identify reference graphs with a similar spatial anchor point configuration by statistic means. In office environments there will always be a high number of similar distances between single pairs of APs (due to standardized furniture, identical door sizes, etc.). But 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 quality measure's construction also prevents reference graphs with a high number of edges of becoming 'more attractive' only due to a resulting higher number of accidental correspondences.

4.2 Alignment

The alignment step aims at finding a coordinate transformation that aligns one anchor point graph G_1 (e.g. a reference graph) with another graph G_2 (e.g. the AP-graph resulting from the actual laser scan). This not only provides the possibility to check the validity of the match by some means of correlation (see section 5); 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 edges (e_{1i}, e_{2j}) in G_1 and G_2 provides two hypotheses how to align the two graphs. For $e_{1i}=(p_a, p_b, d)$ and $e_{2j}=(p_v, p_w, d)$ the combinations (p_a, p_v) and (p_b, p_w) might represent the same APs, 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 $n \times m$ *node correspon-*

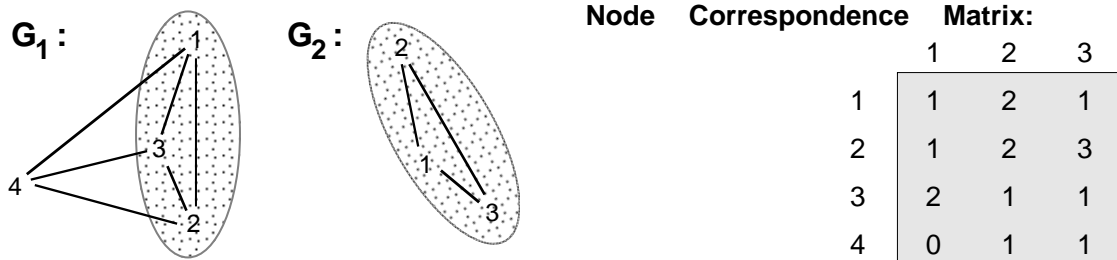


Fig. 3: Alignment of AP graphs using a correspondence matrix

dence matrix (Fig. 3), where n and m are the total numbers of APs in G_1 and G_2 , respectively. For each of the possible AP combinations (p_i, p_j) for each of the edge correspondences of the two graphs, the corresponding matrix element is incremented by 1.

In the example of Fig. 3, G_2 is a true subgraph of 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 a likely node 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, false contributions are eliminated by a clustering step. The remaining transformations are unified by computing the center of gravity, which provides a highly precise alignment for a significant number of recognized APs.

As can be seen in the example of Fig. 3, 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 correspondence matrix. In Fig. 3, 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° , whereas the false edge pair suggests an angle of -45° (modulo 180°). Fig 4 shows two real scans of the same scenery from different position and orientation and the resulting matching angle histogram.

Although the matrix technique is not recommendable for 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.

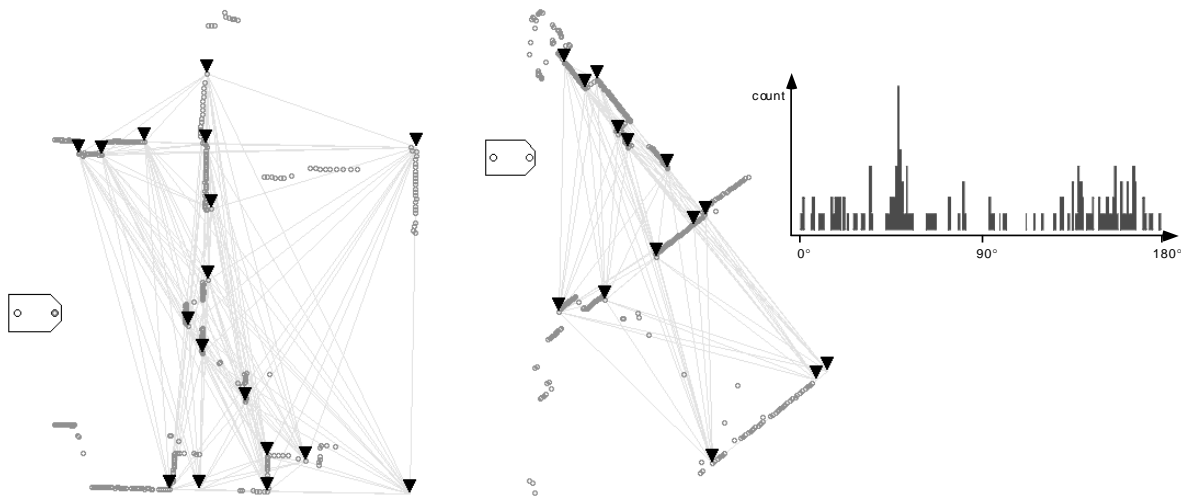


Fig. 4: Scans and AP graphs from different position and orientation, resulting matching angle histogram

5 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 serves two purposes: finding the most likely match in the resulting set of hypotheses and providing a good quality measure e_m for possible use in probabilistic self-localization architectures.

Using 180° laser scans, only, 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 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.

Experimental results have shown that linear numerical comparison functions like the empirical correlation coefficient are too tolerant against range differences, so a simple threshold operator is applied by APR. The evaluation value e_m is then defined as the percentage of range readings in front and synthetic scan which differ by not more than a certain tolerance, with respect to the scan size.

6 Experimental Results & Conclusions

Fig. 5 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 q_m -values were checked for correspondences.

All scan matches were *global*, i.e. no a priori available position information was used to reduce search space.

In Fig. 5, position hypotheses generated from APR's matching results (only matches with highest evaluation rating e_m) are marked by triangles. Correct hypotheses 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 was below a certain threshold. In total, 606 correct global position estimates and 1 wrong estimate were generated, which shows a high hit/error ratio. Average scan processing time includ-

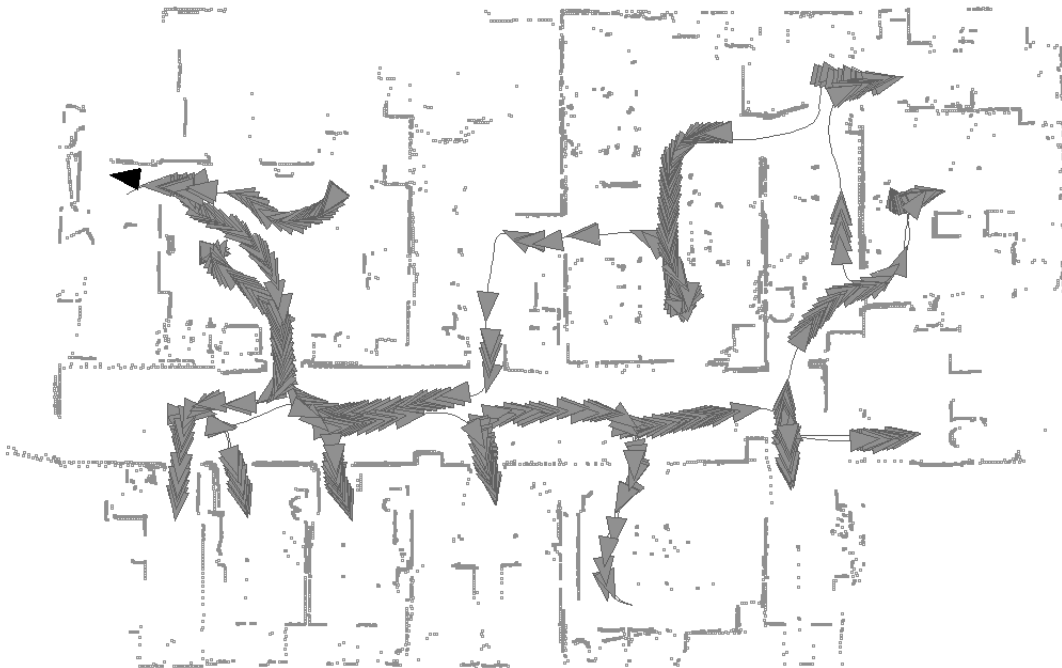


Fig. 5: Global scan matching results. Correct matches are indicated by gray triangles, false by black triangles.

ing all preprocessing steps was below 25 msec with a maximum of 43 msec per cycle on a 233 MHz Pentium PC.

If only the reference scan with the highest q_m -value is processed, the number of correct matches decreases to 366 with no false matching. If all reference scans are checked in each cycle, 819 correct and 10 false matchings are produced with a still acceptable computing time of 72 msec (avg.) and 142 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.

The algorithm handles environmental ambiguities by providing an a priori determined number of different, reasonably quantified hypotheses. This is especially of interest for probabilistic self-localization techniques like Markov approaches.

APR's matching precision can be neatly controlled by the choice of the evaluation function and the e_m -threshold, since these settings are used to determine the validity of a scan match. However, APR's primary goal is to find a robot's global position while the accuracy of this estimate is secondary, only.

Consequently, we don't argue for using APR as a standalone technique in applications with high-precision requirements, but as a method to ensure localization robustness. We are convinced that an autonomous service robot needs to be equipped with such global localization capabilities for reliable long-term operation.

References

- [1] K.-W. Jörg, T. Gattung, J. Weber: *Supporting Mobile Robot Localization by Visual Bar Code Recognition*. Proc. IASTED International Conference on Robotics and Applications, IASTED/Acta Press, pp. 352-357, 1999
- [2] J. L. Crowley, F. Wallner, B. Schiele: *Position Estimation Using Principal Components of Range Data*. Proc. 1998 IEEE International Conference on Robotics & Automation, pp. 3121-3128, 1998
- [3] D. Kortenkamp, T. Weymouth: *Topological mapping for mobile robots using a combination of sonar and vision sensing*. Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94), pp. 979-984, 1994
- [4] R. Simmons, S. Koenig: *Probabilistic Navigation in Partially Observable Environments*. Proc. International Joint Conference on Artificial Intelligence, 1995
- [5] G. v. Wichert: *Selforganizing Visual Perception for Mobile Robot Navigation*. Proc. of the 1st Euro-micro Workshop on Advanced Mobile Robots (EUROBOT'96), IEEE Computer Society Press, pp. 194-200, 1996
- [6] I. J. Cox: *Blanche - An Experiment in Guidance and Navigation of an Autonomous Robot Vehicle*. IEEE Transactions on Robotics and Automation, Vol 7 (2), pp. 193-204, 1991
- [7] K. O. Arras, S. J. Vestli: *Hybrid, High-Precision Localisation for the Mail Distributing Mobile Robot System MOPS*. Proc. 1998 IEEE International Conference on Robotics & Automation, pp. 3129-3134, 1998
- [8] F. Lu, E. E. Milios: *Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans*. Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 935-938, 1994
- [9] G. Weiss, E. v. Puttkamer: *A Map Based on Laserscans Without Geometric Interpretation*. Proc. Intelligent Autonomous Systems (IAS-4), 1995
- [10] R. Hinkel, T. Knieriem: *Environment Perception with a Laser Radar in a Fast Moving Robot*. Symposium on Robot Control 1988 (SYROCO'88), pp. 68.1-68.7, 1988
- [11] D. Fox, W. Burgard, F. Dellaert, S. Thrun: *Monte Carlo Localization: Efficient Position Estimation for Mobile Robots*, Proc. 16th National Conference on Artificial Intelligence (AAAI-99), 1999
- [12] J.-S. Gutmann, C. Schlegel: *AMOS: Comparison of Scan Matching Approaches for Self-Localization in Indoor Environments*, Proc. of the 1st Euro-micro Workshop on Advanced Mobile Robots (EUROBOT'96), IEEE Computer Society Press, pp. 61-67, 1996
- [13] T. Edlinger, G. Weiss: *Exploration, Navigation and Self-Localization in the autonomous mobile robot MOBOT-IV*, International Symposium on Measurement and Control in Robotics 1996 (ISMCR'96)
- [14] I. Nourbakhsh, R. Powers, S. Birchfield: *DERVISH - An Office-Navigating Robot*, AI Magazine 16 (2): pp. 53-60, 1995