



HAL
open science

Merging Partial Maps Without Using Odometry

Francesco Amigoni, Simone Gasparini, Maria Gini

► **To cite this version:**

Francesco Amigoni, Simone Gasparini, Maria Gini. Merging Partial Maps Without Using Odometry. Proceedings from the 2005 International Workshop on Multi-Robot Systems, Mar 2005, Washington, D.C., United States. pp.133-144, <10.1007/1-4020-3389-3_11>. <hal-00818861>

HAL Id: hal-00818861

<https://hal.science/hal-00818861v1>

Submitted on 29 Apr 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

MERGING PARTIAL MAPS WITHOUT USING ODOMETRY

Francesco Amigoni* and Simone Gasparini

Dipartimento di Elettronica e Informazione

Politecnico di Milano, 20133 Milano, Italy

amigoni@elet.polimi.it, gasparin@elet.polimi.it

Maria Gini

Dept of Computer Science and Engineering

University of Minnesota, Minneapolis, USA[†]

gini@cs.umn.edu

Abstract Most map building methods employed by mobile robots are based on the assumption that an estimate of the position of the robots can be obtained from odometry readings. In this paper we propose methods to build a global geometrical map by integrating partial maps without using any odometry information. This approach increases the flexibility in data collection. Robots do not need to see each other during mapping, and data can be collected by a single robot or multiple robots in one or multiple sessions. Experimental results show the effectiveness of our approach in different types of indoor environments.

Keywords: scan matching, map merging, distributed mapping.

1. Introduction

In this paper we show how to build a global map of an environment by merging partial maps without using any relative position information but relying only on geometrical information. The maps we consider are collections of segments obtained from 2D laser range data. They provide a compact and easy-to-use (for example, to plan a path) representation of the environment. No hypothesis is made about the environment to

*Partial funding provided by a Fulbright fellowship and by “Progetto Giovani Ricercatori” 1999.

[†]Partial funding provided by NSF under grants EIA-0224363 and EIA-0324864.

be mapped: experiments demonstrate that our methods work well both in regular and in scattered environments. We reduce the merging of a sequence of partial maps to the iterated integration of two partial maps. The methods we propose are robust to displacements between the partial maps, provided that they have at least a common geometrical feature.

Map merging without odometry information has the advantage of being independent from how the data have been collected. It is indifferent if the partial maps are collected during a single session or multiple sessions, by a single robot or multiple robots. Robots can be added or removed at any time, and they do not need to know their own position. For the experiments in this paper we used a single robot but all the results are applicable to multirobots.

This paper is structured as follows. The next section discusses the main approaches to scan matching and map merging. Section 3 describes our scan matching algorithm, and Section 4 our map merging methods. Experimental results are in Section 5.

2. Previous Work

Scan matching is the process of calculating the translation and rotation of a scan to maximize its overlap with a reference scan. A number of scan matching algorithms have been presented in the last two decades; they differ for the kind of environments in which they perform well, e.g., with straight perpendicular walls (Weiss et al., 1994), for the computational effort, for the choice of operating directly on the scan data (Lu and Milios, 1997) or on segments approximating the data (Zhang and Ghosh, 2000). All these methods require an initial position estimate to avoid erroneous alignments of the two scans.

The most used algorithm (Lu and Milios, 1997) iteratively minimizes an error measure by first finding a correspondence between points in the two scans, and then doing a least square minimization of all point-to-point distances to determine the best transformation. When the environment consists of straight perpendicular walls matching is simpler. Cross-correlation of the histograms of angles between the actual and previous scans provides the orientation of the two maps, while the translation is obtained either by cross-correlation of the distance histogram (Weiss et al., 1994) or by least square minimization (Martignoni III and Smart, 2002). These methods are sensitive to large displacements between the maps and to changes in the environment.

Map merging, namely the problem of building a global map from data collected by several robots, is usually solved by extending SLAM

techniques (Burgard et al., 2002; Ko et al., 2003; Fenwick et al., 2002), or using EM (Simmons et al., 2000; Thrun et al., 2000).

Most map merging techniques rely on the assumption that the robot positions are known. For example, in (Simmons et al., 2000; Burgard et al., 2002) the positions of the robots are assumed to be known at all times; in (Thrun et al., 2000) the robots don't know their relative start position but each robot has to start within sight of the team leader. An exception is the work in (Konolige et al., 2003) where map merging is done using a decision theoretic approach. The robots do not need to know their own position, but to facilitate the match the maps have to be annotated manually with distinctive features. In (Ko et al., 2003), particle filters are used for partial map localization and the robots have to actively verify their relative locations before the maps are merged.

3. Method for Scan Matching

We propose a `MATCH` function for matching two partial maps composed of segments. Our method is exclusively based on the geometrical information and constraints (Grimson, 1990) contained in the partial maps. In particular, we consider angles between pairs of segments in the partial maps as a sort of “geometrical landmarks” on which the matching process is based. This use of “local” geometrical features is significantly different from other related works in map building that use “global” geometrical features (e.g., those represented by an histogram of angle differences). `MATCH` integrates two partial maps into a third one. Let's call S_1 and S_2 the two partial maps and $S_{1,2}$ the resulting map. `MATCH` operates in three major steps:

1. Determine possible transformations. This step first finds the angles between the segments in S_1 and between the segments in S_2 and then finds the possible transformations (namely, the rotations and translations) that superimpose at least one angle α_2 of S_2 to an (approximately) equal angle α_1 of S_1 . Recall that angles between pairs of segments in a partial map are the geometrical landmarks we adopt.

In principle, without any information about the relative positions of the two scans, there are $O(n_1^2 n_2^2)$ possible transformations, where n_1 and n_2 are the number of segments in S_1 and S_2 , respectively. We have devised three heuristics to speed up the computation:

- a *Consider Angles between Consecutive Segments.* In each scan, we consider only the angles between two consecutive segments; let A_1^s and A_2^s be the sets of such angles for S_1 and S_2 , respectively. The number of possible transformations drops to $O(n_1 n_2)$. Finding

the sets A_1^s and A_2^s is easy when the segments in S_1 and in S_2 are ordered, which is usually the case with laser range scanners.

- b *Consider Angles between Randomly Selected Segments.* In each scan, we examine the angles between pairs of segments selected randomly. We assign a higher probability to be selected to longer segments, since they provide more precise information about the environment. Let A_1^r and A_2^r be the sets of the selected angles for S_1 and S_2 , respectively. The number of transformations generated by this method is $O(a_1 a_2)$, where $a_1 = |A_1^r|$ and $a_2 = |A_2^r|$ are the number of selected angles in A_1^r and A_2^r , respectively.
- c *Consider Angles between Perpendicular Segments.* In each scan, we select only angles between perpendicular segments. This heuristic is particularly convenient for indoor environments, where walls are often normal to each other. The heuristic is computed from the histogram of segments grouped by orientation. The direction where the sum of the lengths of the segments is maximal is the *principal direction*. In Fig. 1, the histogram of a scan taken in an indoor environment is shown. The principal direction is the element L_9 and the normal direction is the element L_0 . Let A_1^h and A_2^h be the sets of angles formed by a segment in the principal direction and a segment in the normal direction of the histograms of S_1 and S_2 , respectively. The set of possible transformations is found comparing the angles in A_1^h and A_2^h . The number of possible transformations is $O(p_1 n_1 p_2 n_2)$, where p_i and n_i are respectively the number of segments in the principal and in the normal directions of the histogram of scan S_i .

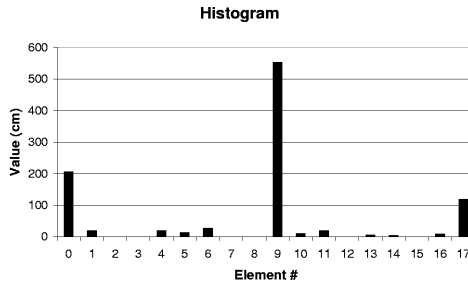


Figure 1. The histogram of a scan

2. Evaluate the transformations. To measure the goodness of a transformation t we transform S_2 on S_1 (in the reference frame of S_1) according to t (obtaining S_2^t), then we calculate the approximate length

of the segments of S_1 that correspond to (namely, match with) segments of S_2^t . Thus, the measure of a transformation is the sum of the lengths of the corresponding segments that the transformation produces. More precisely, for every pair of segments $s_1 \in S_1$ and $s_2^t \in S_2^t$ we project s_2^t on the line supporting s_1 and compute the length l_1 of the common part of s_1 and the projected segment. We repeat the process by projecting s_1 on s_2^t , obtaining l_2 . The average of l_1 and l_2 is a measure of how the pair of segments match. This step evaluates a single transformation by considering all the pairs of segments of the two scans that are $O(n_1 n_2)$.

3. Apply the best transformation and fuse the segments.

Once the best transformation \bar{t} has been found, this step transforms the second partial map S_2 in the reference frame of S_1 according to \bar{t} obtaining $S_2^{\bar{t}}$. The map that constitutes the output of MATCH is then obtained by fusing the segments of S_1 with the segments of $S_2^{\bar{t}}$. The main idea behind the fusion of segments is that a set of matching segments is substituted in the final map by a single polyline. We iteratively build a sequence of approximating polylines P_0, P_1, \dots that converges to the polyline P that adequately approximates (and substitutes in the resulting map) a set of matching segments. The polyline P_0 is composed of a single segment connecting the pair of farthest points of the matching segments. Given the polyline P_{n-1} , call s the (matching) segment at maximum distance from its corresponding (closest) segment \bar{s} in P_{n-1} . If the distance between s and \bar{s} is less than the acceptable error, then P_{n-1} is the final approximation P . Otherwise, s substitutes \bar{s} in P_{n-1} and s is connected to the two closest segments in P_{n-1} to obtain the new polyline P_n .

4. Methods for Map Merging

We now describe methods for integrating a sequence S_1, S_2, \dots, S_n of n partial maps by repeatedly calling MATCH.

Sequential Method. This is the simplest method, which operates as follows. The first two partial maps are integrated, the obtained map then is grown by sequentially integrating the third partial map, the fourth partial map, and so on. Eventually, the final map $S_{1,2,\dots,n}$ is constructed. In order to integrate n partial maps, the sequential method requires $n-1$ calls to MATCH. A problem with this method is that, as the process goes on, MATCH is applied to a partial map that grows larger and larger (it contains more and more segments). This will cause difficulties in the integration of S_i with large i , since S_i could match with different parts of the larger map.

Tree Method. To overcome the above problem, the integration of a small partial map with a large partial map should be avoided. This is the idea underlying the *tree* method, which works as follows. Each partial map of the initial sequence is integrated with the successive partial map of the sequence to obtain a new sequence $S_{1,2}, S_{2,3}, \dots, S_{n-1,n}$ of $n - 1$ partial maps. Then, each partial map of this new sequence is integrated with the successive one to obtain a new sequence $S_{1,2,3}, S_{2,3,4}, \dots, S_{n-2,n-1,n}$ of $n - 2$ partial maps. The process continues until a single final map $S_{1,2,\dots,n}$ is produced. The tree method always integrates partial maps of the same size, since they approximately contain the same number of segments. The number of calls to MATCH required by the tree method to integrate a sequence of n partial maps is $n(n - 1)/2$. Note also that, while the sequential method can be applied in an on-line fashion (i.e., while the robot is moving), the most natural implementation of the tree method is off-line. To speed up the tree method we have developed an heuristic that, given a sequence of partial maps at any level of the tree (let us suppose at level 0 for simplicity), attempts to integrate the partial maps S_i and S_{i+2} ; if the integration succeeds, the final result $S_{i,i+2}$ represents the same map that would have been obtained with three integrations: S_i with S_{i+1} to obtain $S_{i,i+1}$, S_{i+1} with S_{i+2} to obtain $S_{i+1,i+2}$, and $S_{i,i+1}$ with $S_{i+1,i+2}$ to obtain $S_{i,i+1,i+2}$. Moreover, the number of partial maps in the new sequence is reduced by one unit, because $S_{i,i+2}$ substitutes both $S_{i,i+1}$ and $S_{i+1,i+2}$. This heuristic is best used when the partial maps S_i and S_{i+2} are already the result of a number of integrations performed by the tree method and their common part is significant. For example, in the sequence produced at the level 3 of the tree technique the first ($S_{1,2,3,4}$) and the third ($S_{3,4,5,6}$) partial maps have a significant common part, since approximately half of the two partial maps overlaps.

A problem with the tree method is due to the presence of “spurious” segments in the integrated maps, namely segments that correspond to the same part of the real environment but that are not fused together. This problem is exacerbated in the tree method since the same parts of the partial maps are repeatedly fused together.

Pivot Method. The *pivot* method combines the best features of the two previous methods. This method starts as the tree method and constructs a sequence $S_{1,2}, S_{2,3}, \dots, S_{n-1,n}$ of $n - 1$ partial maps. At this point, we note that S_2 is part of both $S_{1,2}$ and $S_{2,3}$ and that the transformation $\bar{t}_{1,2}$ used to integrate S_1 and S_2 provides the position and orientation of the reference frame of S_2 in the reference frame of $S_{1,2}$. It is therefore possible to transform $S_{2,3}$ according to $\bar{t}_{1,2}$ and fuse the

segments of the partial maps $S_{1,2}$ and $S_{2,3}^{\bar{t}_{1,2}}$ to obtain $S_{1,2,3}$. In a similar way, $S_{1,2,3,4}$ can be obtained from $S_{1,2,3}$ and $S_{3,4}$ by applying to the latter the transformation $\bar{t}_{2,3}$ and fusing the segments of $S_{1,2,3}$ and $S_{3,4}^{\bar{t}_{2,3}}$. Iterating this process, from the sequence $S_{1,2}, S_{2,3}, \dots, S_{n-1,n}$ the final map $S_{1,2,\dots,n}$ is obtained. The pivot method integrates partial maps of the same size, like the tree method, and requires $n - 1$ calls to MATCH, like the sequential method. (In addition it requires $n - 2$ executions of the not-so-expensive step 3 of MATCH.) The pivot method is naturally implementable in an on-line system. The problem of spurious segments is reduced but not completely eliminated; a way to further reduce this problem is to fuse not $S_{1,2}$ and $S_{2,3}^{\bar{t}_{1,2}}$, but $S_{1,2}$ and $S_3^{\bar{t}_{1,3}}$, where $\bar{t}_{1,3}$ is the composition of $\bar{t}_{1,2}$ and $\bar{t}_{2,3}$.

5. Experimental Results

The described methods have been validated using a Robuter mobile platform equipped with a SICK LMS 200 laser range scanner mounted in the front of the robot at a height of approximately 50 cm. For these experiments we acquired 32 scans with angular resolution of 1° and with angular range of 180° . Each scan has been processed to approximate the points returned by the sensor with segments, according to the algorithm in (González-Baños and Latombe, 2002). The programs have been coded in ANSI C++ employing the LEDA libraries 4.2 and they have been run on a 1 GHz Pentium III processor with Linux SuSe 8.0.

The scans have been acquired in different environments (forming a loop about 40 m long) by driving the robot manually and without recording any odometric information. We started from a laboratory, a very scattered environment, then we crossed a narrow hallway with rectilinear walls to enter a department hall, a large open space with long perpendicular walls, and finally we closed the loop re-entering the laboratory (see the dashed path in Fig. 6). The correctness of integrations has been determined by visually evaluating the starting partial maps and the final map with respect to the real environment.

5.1 Scan Matching Experiments

Table 1 shows the results obtained by matching three interesting pairs of scans (see also Fig. 2). S_4 and S_5 were taken inside the laboratory: they contain a large number of short segments since the environment is highly scattered. S_{18} and S_{19} were taken along the hallway: they contain fewer segments than the previous scans and are characterized by long rectilinear segments. S_{25} and S_{26} were taken in the hall: they



Figure 2. Top, left to right: scans S_4 , S_5 , S_{18} , S_{19} , S_{25} , and S_{26} ; bottom, left to right: final maps $S_{4,5}$, $S_{18,19}$, and $S_{25,26}$

contain only few segments since the environment is characterized by long rectilinear and perpendicular walls.

Table 1. Experimental results on matching pairs of scans

Scans	S_4	S_5	S_{18}	S_{19}	S_{25}	S_{26}
# of segments	47	36	24	24	10	12
	Time	# tried	Time	# tried	Time	# tried
All transformations	936	41,260	32	3,096	0.38	231
Consecutive segments	1.25	2	0.73	27	0.13	4
Random segments	7.69	20,000	2.51	20,000	0.78	20,000
Histogram	3.29	73	1.97	192	0.15	32

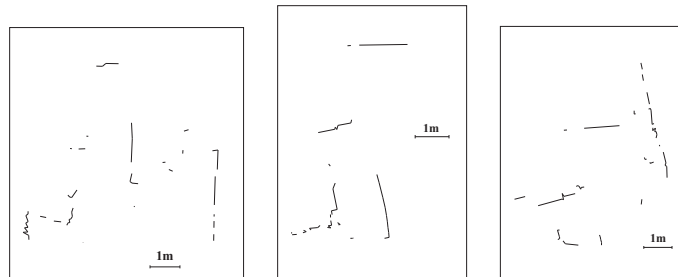


Figure 3. Scans S_1 (left), S_2 (center), and S_3 (right) taken in the lab entrance

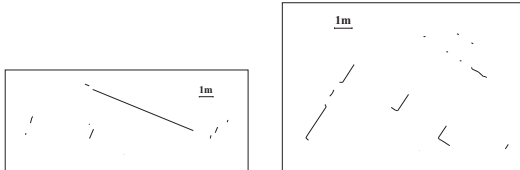


Figure 4. Scans S_{27} (left) and S_{28} (right) taken in the hall

Table 2. Results of scan matching trials using different heuristics

	Successes	Failures
All transformations	13 (41.9%)	18 (58.1%)
Consecutive segments	21 (67.7%)	10 (32.3%)
Random segments	10 (32.2%)	21 (67.8%)
Histogram	9 (29%)	22 (71%)

In general, our experimental results demonstrate that scan matching performs well (Table 2), but not all the pairs can be matched. 28 pairs of scans out of 31 have been correctly matched. Unsurprisingly, the histogram-based heuristic worked well with scans containing long and perpendicular segments, as those taken in the hallway and in the hall. The heuristic based on consecutive segments seems to work well in all three kinds of environment, even if sometimes it needs some parameter adjustments.

For scan pairs $S_1 - S_2$ and $S_2 - S_3$ our method was not able to find the correct transformation. As shown in Fig. 3, the scans are extremely rich of short segments representing scattered small objects (chairs, tables, robots, and boxes). It is almost impossible, even for a human being, to find the correct match between these scans without any prior information about their relative positions. Similar problems emerged in the hall. Fig. 4 shows scans S_{27} and S_{28} , where the second one has been taken after rotating the robot of about 100 degrees. Since the environment is large and has only few objects that can be used as reference, a drastic change of the field of view eliminates any common reference between scans, thus automatic matching is impossible.

5.2 Map Merging Experiments

We considered the sequence composed of 29 scans S_1, S_2, \dots, S_{29} (Table 3). The integration of this sequence of partial maps has been done off-line to test and compare the three methods. In all the three methods, problems arose when integrating the sub-sequence from S_{25} to S_{27} which

represents the hall (Fig. 4). Here, due to a drastic rotation (about 100 degrees) of the robot in such an open and large environment, the partial maps have only one or two segments in common. In order to close the loop and complete the experiments these partial maps were manually integrated together in all the three methods.

Table 3. Experimental sequence of partial maps (the segment lengths are in mm)

Environment	Partial maps	Avg # of segments	Avg length of segments
Laboratory	$S_1 - S_7, S_{28} - S_{29}$	38.1	259.3
Hallway	$S_8 - S_{22}$	19.3	366.3
Hall	$S_{23} - S_{27}$	15.6	607
Total	$S_1 - S_{29}$	24.53	374.5

Fig. 5 shows the final map (composed of 278 segments) obtained with the sequential method. The sequential method could not integrate all the partial maps in order to close the loop: the method suddenly failed when we tried to integrate S_{21} , which has only a few short segments in common with the rest of the map.

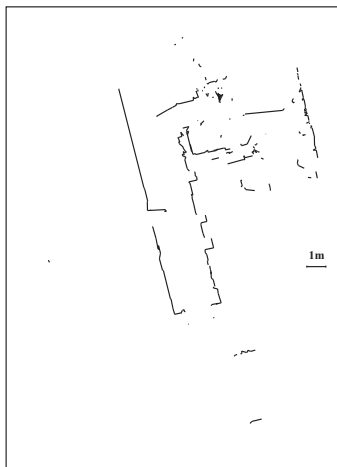


Figure 5. The final map obtained with the sequential method. ©2004 by IEEE (Amigoni et al., 2004)

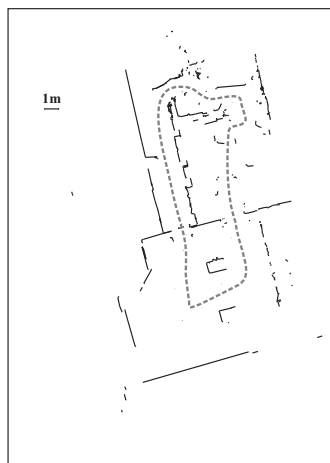


Figure 6. The final map obtained with the tree method. ©2004 by IEEE (Amigoni et al., 2004)

Fig. 6 shows the final map (composed of 519 segments) obtained with the tree method. We applied the standard tree method until level 3 of the tree, then we applied the heuristic presented in Section 4 to speed up the process. As we went down in the tree, the size of the maps grew

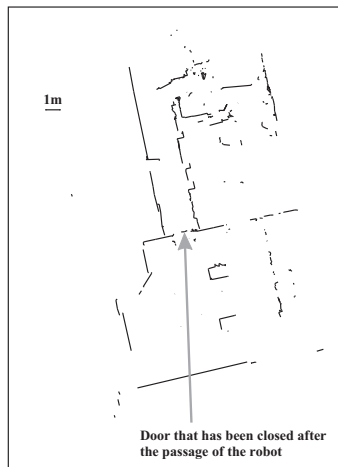


Figure 7. The final map obtained with the pivot method by fusing $S_{i-1,i}$ with $S_{i,i+1}^{\bar{i}-1,i}$. ©2004 by IEEE (Amigoni et al., 2004)

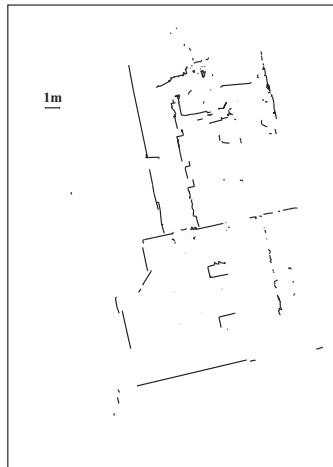


Figure 8. The final map obtained with the pivot method by fusing $S_{i-1,i}$ with $S_{i+1}^{\bar{i}-1,i+1}$. ©2004 by IEEE (Amigoni et al., 2004)

larger and larger and the execution of MATCH slowed down. For example, the integration of two partial maps (composed of 108 and 103 segments) at level 3 of the tree requires 12.8 s. Furthermore, as already noted, when we integrate large-sized maps with many redundant spurious segments that represent the same part of the environment, the resulting maps are more noisy because of the error introduced when attempting to integrate maps with many overlapping segments.

Fig. 7 shows the final map, composed of 441 segments, obtained with the pivot method by fusing the partial map $S_{i-1,i}$ with $S_{i,i+1}^{\bar{i}-1,i}$. The map in Fig. 8 is composed of 358 segments and has been built by fusing the partial map $S_{i-1,i}$ with $S_{i+1}^{\bar{i}-1,i+1}$. This map presents fewer spurious segments and appears more “clean”.

6. Conclusions

In this paper we have presented methods for matching pairs of scans composed of segments and for merging a sequence of partial maps in order to build a global map. In future research we aim at generalizing these methods to cases where the order in which the partial maps have to be integrated is not known. These generalized methods will provide an elegant solution to the problem of multirobot mapping since they will work when partial maps are acquired by a single robot at different times as well as when acquired by different robots in different locations.

Acknowledgments

The authors would like to thank Jean-Claude Latombe for his generous hospitality at Stanford University where this research was started, Héctor González-Baños for sharing his programs and expertise with collecting laser range scan data, Paolo Mazzoni and Emanuele Ziglioli for the initial implementation of the fusion algorithm.

References

- Amigoni, F., Gasparini, S., and Gini, M. (2004). Scan matching without odometry information. In *Proc. of the IEEE Int'l Conference on Robotics and Automation*, pages 3753–3758.
- Burgard, W., Moors, M., and Schneider, F. (2002). Collaborative exploration of unknown environments with teams of mobile robots. In *Advances in Plan-Based Control of Robotic Agents*, pages 52–70. Springer-Verlag.
- Fenwick, J. W., Newman, P. M., and Leonard, J. J. (2002). Cooperative concurrent mapping and localization. In *Proc. of the IEEE Int'l Conference on Robotics and Automation*, pages 1810–1817.
- González-Baños, H. H. and Latombe, J. C. (2002). Navigation strategies for exploring indoor environments. *Int'l Journal of Robotics Research*, 21(10-11):829–848.
- Grimson, W. E. L. (1990). *Object recognition by computer: the role of geometric constraints*. The MIT Press.
- Ko, J., Stewart, B., Fox, D., and Konolige, K. (2003). A practical, decision-theoretic approach to multi-robot mapping and exploration. In *Proc. of the IEEE/RSJ Int'l Conference on Intelligent Robots and Systems*, pages 3232–3238.
- Konolige, K., Fox, D., Limketkai, B., Ko, J., and Stewart, B. (2003). Map merging for distributed robot navigation. In *Proc. of the IEEE/RSJ Int'l Conference on Intelligent Robots and Systems*.
- Lu, F. and Milios, E. (1997). Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic Systems*, 18(3):249–275.
- Martignoni III, A. and Smart, W. (2002). Localizing while mapping: A segment approach. In *Proc. of the Eighteen National Conference on Artificial Intelligence*, pages 959–960.
- Simmons, R. G., Apfelbaum, D., Burgard, W., Fox, D., Moors, M., Thrun, S., and Younes, H. (2000). Coordination for multi-robot exploration and mapping. In *Proc. of the National Conference on Artificial Intelligence*, pages 852–858.
- Thrun, S., Burgard, W., and Fox, D. (2000). A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *Proc. of the IEEE Int'l Conference on Robotics and Automation*, pages 321–328.
- Weiss, G., Wetzler, C., and Puttkamer, E. V. (1994). Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans. In *Proc. of the IEEE/RSJ Int'l Conference on Intelligent Robots and Systems*, pages 12–16.
- Zhang, L. and Ghosh, B. (2000). Line segment based map building and localization using 2D laser rangefinder. In *Proc. of the IEEE Int'l Conference on Robotics and Automation*, pages 2538–2543.