HAL
open science

Scan Matching Without Odometry Information

Francesco Amigoni, Simone Gasparini, Maria Gini

To cite this version:

Francesco Amigoni, Simone Gasparini, Maria Gini. Scan Matching Without Odometry Information. First International Conference on Informatics in Control, Automation and Robotics (ICINCO2004), Apr 2005, Setubal, Portugal. pp.349-352, 10.5220/0001141303490352 . hal-00817540

HAL Id: hal-00817540

https://hal.science/hal-00817540v1

Submitted on 24 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.

SCAN MATCHING WITHOUT ODOMETRY INFORMATION

Francesco Amigoni, Simone Gasparini
Dipartimento di Elettronica e Informazione, Politecnico di Milano
Piazza Leonardo da Vinci 32, 20133 Milano, Italy
Email: amigoni@elet.polimi.it, gasparini@airlab.elet.polimi.it

Maria Gini
Dept of Computer Science and Engineering, University of Minnesota
Minneapolis, USA
Email: gini@cs.umn.edu

Key words: Scan Matching. Robot Mapping. Laser Range Scanner.

Abstract: We present an algorithm for merging two partial maps obtained with a laser range scanner into a single map. The most unique aspect of our algorithm is that it does not require any information on the position where the scans were collected. The algorithm operates by performing a geometric match of the two scans and returns the best fused map obtained by merging the two partial maps. The algorithm attempts to reduce the number of segments in the fused map, by replacing overlapping segments with a single segment. We present heuristics to speed up the computation, and experimental results obtained with a mobile robot in an indoor environment.

1 INTRODUCTION

The increasing use of mobile robots equipped with laser range scanners has stimulated the development of methods for aligning scan data collected by these sensors. Usually these methods align two scans starting from some information about the relative position of the sensors obtained from odometry (Lu and Milios, 1997; Cox, 1991; Röfer, 2001).

In this paper we present a method for matching two scans that does not require *any* odometry information. For the purposes of this paper, a scan is a collection of segments. In our experimental setting each scan is obtained by acquiring with a laser range scanner (mounted on a mobile robot at a given height) a sequence of distance measurements along directions separated by a programmable angle (one degree, in our case). The result of the sensing operation is thus a set of points expressed in polar coordinates, with the origin of the coordinate frame in the sensor itself. We approximate these points with a set of segments following the method described in (Gonzáles-Baños and Latombe, 2002). The use of segments instead of points reduces the computational complexity of finding the match between scans. Since the method does not use odometry information, it relies exclusively on the geometry of the scans. In particular, we consider the angles between pairs of segments in the scans as a sort of "geometrical landmarks" on which the matching process is based. We assume that the robot moves on a 2D surface and that walls and vertical objects are at the height of the laser scan. The method proposed in this paper can correctly match two scans even when their displacement is significative, provided that they have an overlap containing at least an angle representing the same portion of the environment.

The method integrates two scans, S_1 and S_2, into a final map $S_{1,2}$. It is composed of three major steps:

1. *determine the possible transformations* of S_2 on S_1;
2. *evaluate the transformations* to identify the best transformation $\bar{t}$ of S_2 on S_1;
3. *apply the best transformation* to S_2 (obtaining $S_2^{\bar{t}}$) and *fuse the segments* of S_1 and of $S_2^{\bar{t}}$ to obtain $S_{1,2}$.

The main advantage of our scan matching method is that, since it is independent of any prior knowledge about the relative position of the scans, it is applicable indifferently in situations in which the two scans have been perceived by the same robot at different time instants, as well as in situations in which the two scans have been perceived by two robots in two different locations. In both cases, S_1 and S_2 are matched only on the basis of the geometrical information they contain. For this reason, our scan matching method is naturally applicable to multirobot map building (Simmons et al., 2000; Burgard et al., 2002). In this context, the map merging problem, namely the problem of building a global map from data collected

by several robot, is usually solved by extending the SLAM techniques (Burgard et al., 2002; Williams et al., 2002; Stewart et al., 2003; Fenwick et al., 2002), the EM techniques (Simmons et al., 2000; Thrun et al., 2000), or a combination of the two approaches (Thrun, 2001). All these 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 every time instant; in (Thrun et al., 2000) the robots don't know their relative start position but it is assumed that each robot starts within sight of a robot, called the team leader. Our scan matching method could be employed to merge overlapping maps of different robots without knowing their positions, since the method is fully independent of the displacement between the maps. In the following, we refer to the single robot case but all the results are applicable to the multirobot case.

This paper is structured as follows. The next Section reviews the state of the art in scan matching methods. In Section 3 we describe in detail our method and in Section 4 we discuss the experimental activity performed to validate it. Section 5 concludes the paper.

2 THE SCAN MATCHING PROBLEM

Scan matching is the process of calculating the translation and rotation of a scan to maximize its overlap with a reference scan. The translation and rotation can be represented by a displacement vector. Most of the scan matching methods discussed in the following employ laser range scanners, since they guarantee more accuracy and reliability than sensors like sonars.

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., linear, rectilinear or polygonal environments) and for the required computational effort. For example, the method proposed in (Weiss et al., 1994) adopts cross-correlation to approach the problem. From the actual and previous point scans, angle and distance histograms are extracted and cross-correlated. The maxima of the respective cross-correlation functions give the rotation and the translation between the two scans. The method uses odometry for an initial position estimate, since evaluation of cross-correlation functions might be erroneous for large displacements between scans. The major drawback of this method is that the algorithm performs well only in environments consisting of straight perpendicular walls and it only allows for minor changes of the environment. The improvement in (Röfer, 2001) deals with non-perpendicular walls and segment maps, even if it still assumes straight walls and shows poor performances in scattered environments.

Another well-known technique is the iterative algorithm of (Cox, 1991) for matching range scans to an *a priori* map of line segments. Since it assumes small displacements between a scan and the map, the algorithm first finds the correspondence between scan points and line segments and then calculates the translation and rotation that minimize the (square of all) point-to-segment distances. The two steps are repeated until the process converges. Each iteration returns a position correction vector and a variance-covariance matrix that evaluates the match. This approach has been extended by (Gutmann and Schlegel, 1996): instead of having an *a priori* model of the environment, line segments are extracted from the previous scans and used as the reference model for the matching process. Rejection criteria are used for reducing the amount of false point-to-line assignments in order to speed-up the method. These last two methods can be applied only to polygonal environments, a limitation that our method tries to partially overcome.

The IDC (Iterative Dual Correspondence) algorithm (Lu and Milios, 1997) performs well in rectilinear as well as in irregular environments. Like (Cox, 1991), IDC iteratively minimizes an error measure by first finding a correspondence between points in the reference scan and points in the actual scan, and then doing a least square minimization of all point-to-point distances to determine the best translation and rotation. An initial position estimate is provided through odometry to avoid erroneous alignments. The computational cost of IDC is high and the method does not seem to be suited for polygonal environments. Thus, (Gutmann and Schlegel, 1996) proposed a scan matching method that combines (Cox, 1991) (used in polygonal environments) and IDC (used in non-polygonal environments). IDC-S (IDC-Sector) (Bengtsson and Baerveldt, 1999) reduces noise sensitivity of original IDC and copes with dynamic environments. All these methods employ information obtained from odometry.

The method proposed in (Zhang and Ghosh, 2000) extracts line segments from range points and uses a special Center of Gravity representation for describing the uncertainty of line segments. Relying on odometry readings as initial estimate for the displacement vector, it matches pairs of segments and computes the translation by least square minimization, as in (Lu and Milios, 1997).

The method presented by (Einsele, 1997) uses a panoramic range finder to build segment maps. It extracts line segments representing walls or other boundaries of the environment and matches the scans taken from different positions without relying on any additional source of information. This is done by employing dynamic programming algorithms applied to

the vertical lines of the map. It can operate in polygonal or rectilinear environment but it doesn't work well in scattered environments and it relies on small displacement of the robot.

Finally, the method illustrated in (Martignoni III and Smart, 2002) extracts line segments from laser range readings and matches the actual map and a global map incrementally built during the exploration. It first determines the heading between the two maps by computing the histogram of the angle differences and then adjusts the translation by overlapping the line segments with least square minimization. The method works in linear and static environments and with very small displacements between two maps; the method can also perform a global search, but it becomes slower and prone to errors in environments with many similarities.

3 THE PROPOSED METHOD

This Section details the three main steps of our method for scan matching. In the algorithms, two points are considered to coincide when they are closer than POINTDISTANCETOLERANCE (in our experiments we set this parameter to $15\,\mathrm{mm}$) and two angles are considered equal when their values differ for less than ANGLEDIFFERENCETOLERANCE (in our experiments we set this parameter to $0.2\,\mathrm{rad}$).

3.1 Determining the Possible Transformations

This step, given the scans S_1 and S_2, first finds the angles between segments in S_1 and between segments in S_2 and, second, finds the possible transformations (namely, the rotations and translations) that superimpose at least one angle α_2 of S_2 to an equal angle α_1 of S_1. Recall that angles between pairs of segments in a scan are the geometrical landmarks we adopt. Finding the possible transformations is a difficult combinatorial problem since 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 numbers of segments in S_1 and S_2, respectively. We have therefore devised three heuristics for reducing this complexity and finding a set of (hopefully) significant transformations between two scans. The three heuristics are described in the following.

1. Considering Angles between Consecutive Segments. In each scan, we select 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. Two segments are considered consecutive when they have an extreme point in common. Then, we find the set of all the transformations that make an angle in A_2^s to correspond to an equal angle in A_1^s. The number of possible transformations found by this method is $O(n_1 n_2)$. We note that finding the sets A_1^s and A_2^s is greatly facilitated when the segments in S_1 and in S_2 are ordered. This is usually the case with laser range scanners, since the points returned by the sensor are ordered counterclockwise and it is straightforward to maintain the same order in the segments that approximate the points.

Although this method seems to perform well in indoor environments where the angles are usually regular, the errors introduced by the sensor and by the algorithm that approximates points with segments alter the representation of these regular angles. For example, Fig. 1 shows the segment representation of a portion of an environment with four right angles. It is evident that the angles between consecutive segments sometimes do not constitute a good model of the environment angles.

Figure 1: A portion of a scan representing four right angles of the environment

To improve the performance of this heuristic, it is possible to consider angles between consecutive segments without a common extreme point (when an order is defined for the segments of a scan) by setting the parameter NONCONSECUTIVEEDGES to "yes". Consecutive segments can be considered to form significant angles only if they are longer than a fraction (specified by the parameter SEGMENTLENGTHPERCENTAGE) of the longest segment in the scan. The implicit assumption is that long segments are more reliable than short segments in representing the environment.

2. Considering Angles between Randomly Selected Segments. In each scan, we examine a number of 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. We find the set of all the transformations that brings an angle in A_2^r to correspond to an equal angle in A_1^r. 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.

Instead of assigning directly to each segment the probability of being selected (according to its length) and of selecting the a_1 (respectively a_2) pairs, the following approximate and easy-to-implement technique is employed. Initially only segments longer than SEGMENTDIVISIONFACTOR times the length of the longest segment in S_1 (resp. S_2) are considered for selection. All the segments considered have equal probability of being selected. Then, we proceed to iterate with $k = 1, \ldots, K$. During the k-th iteration, we use a threshold equal to SEGMENTDIVISIONFACTOR^k times the length of the longest segment in S_1 (resp. S_2). Out of the segments longer than this threshold we select one with equal probability. Thus, the parameter SEGMENTDIVISIONFACTOR determines the length of the segments that are considered for selection and, implicitly, the probability of selection. This technique tries first to find transformations based on angles between long segments; then it progressively considers transformations based on angles between shorter and shorter segments. The above technique can be further improved by stopping the generation of transformations when a "good enough" transformation is found. (The evaluation of the quality of a transformation is discussed in Section 3.2.)

3. Considering Angles between Perpendicular Segments. In each scan, we select only angles between perpendicular segments. This heuristic is particularly convenient for indoor environments, where the presence of regular walls usually involves perpendicular segments. The heuristic is based on histograms. The *histogram* of S_1 (and, in similar way, that of S_2) is an array of *nslots* elements, where *nslots* is the number of buckets of the histogram. Each bucket L_i ($i = 0, 1, \ldots, \textit{nslots} - 1$) contains the segments with orientation comprised between $\pi \times i/\textit{nslots}$ and $\pi \times (i+1)/\textit{nslots}$, measured with respect to a given reference axis. To each element L_i of the histogram of S_1 is associated a value calculated as the sum of the lengths of the segments in L_i. The *principal direction* of an histogram is the element with maximum value. The *normal direction* of an histogram is the element that is $\pi/2\,\mathrm{rad}$ away from the principal direction. In Fig. 2, 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 by a segment in the normal direction of the histograms of S_1 and S_2, respectively. The set of possible transformations is then found comparing the angles in A_1^h and A_2^h. The number of possible transformations generated by the above heuristic is $O(p_1 n_1 p_2 n_2)$, where p_i and n_i are the number of segments in the principal and normal directions of the histogram of scan S_i.

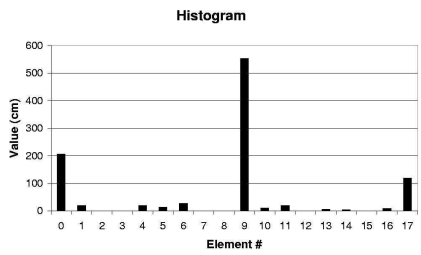

Figure 2: The histogram of a scan

3.2 Evaluating the Transformations

Every transformation found in the previous step is evaluated in order to identify the best one. To determine 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. The measure of a transformation is the length of the corresponding segments that the transformation produces. More precisely, the measure of a transformation is the sum of all the matching values calculated for every pair of segments $s_1 \in S_1$ and $s_2^t \in S_2^t$. The *matching value* between two segments s_1 and s_2^t is calculated as follows. We project s_2^t on the line supporting s_1 thus getting a projected segment s_{2p}^t and then we compute the length l_1 of the common part of $s1$ and s_{2p}^t; we do the same but projecting s_1 on s_2^t, obtaining l_2. The matching value of s_1 and s_2^t is calculated as the average of l_1 and l_2. When s_1 and s_2^t do not intersect, the matching value is multiplied by $0.95^{d(s_1, s_2^t)/\text{POINTDISTANCETOLERANCE}}$ to penalize the match between segments that are far away. Note that $\ln(0.95)$ is an empirical constant whose value has been determined during experimental activities and $d(s_1, s_2)$ is the distance between two segments, calculated as

$$\begin{aligned} d(s_1, s_2) = \min(&\max(\text{dist}(s_1, \text{start}(s_2)), \\ &\text{dist}(s_1, \text{end}(s_2))), \\ &\max(\text{dist}(s_2, \text{start}(s_1)), \\ &\text{dist}(s_2, \text{end}(s_1)))) \end{aligned}$$

where start(s) and end(s) are the extremes of segment s (see Fig. 3). Finally, two special cases can appear during the evaluation of the matching values of s_1 and s_2^t. The matching value is set to 0

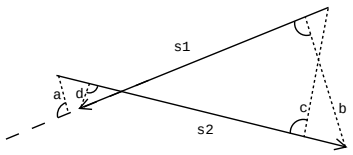

Figure 3: The distance between s_1 and s_2 is the quantity $d(s_1, s_2) = \min(\max(a, b), \max(c, d))$, where the marked angles are equal to $\pi/2$

when the two segments are too far away, namely when $d(s_1, s_2^t)/\text{POINTDISTANCETOLERANCE} > \text{SEGMENTDISTANCETHRESHOLD}$. SEGMENTDISTANCETHRESHOLD is usually set to 5 to obtain good experimental results. The matching value is set to -1 when the two segments intersect and are longer than SEGMENTLENGTHREFUSE; in this case the transformation is discarded.

The above algorithm evaluates a single transformation by considering all the pairs of segments of the two scans that are $O(n_1 n_2)$. A way to limit this computational effort is to stop the evaluation of a transformation t when its measure cannot be larger than the current maximum.

3.3 Transforming and Fusing Scans

Once the best transformation $\bar{t}$ has been found, the third and last step of our method transforms the second scan 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 our scan matching method is then obtained by fusing the segments of S_1 with the segments of $S_2^{\bar{t}}$. To this end, we use the idea of matching chains. A *matching chain* of the pair of scans S_1 and $S_2^{\bar{t}}$ is a set $C = \{\langle s_1, s_2^{\bar{t}}\rangle | s_1 \in S_1$ and $s_2^{\bar{t}} \in S_2^{\bar{t}}$ have a positive matching value for $\bar{t}\}$ algebraically closed under segment belong-to relation. Specifically, a matching chain C is such that if $\langle s_1, s_2^{\bar{t}}\rangle \in C$, then also $\langle s_1, s\rangle \in C$ and $\langle s, s_2^{\bar{t}}\rangle \in C$ for all the segments s that have a positive match value (namely, have matched with) s_1 or with $s_2^{\bar{t}}$. We explicitly note that, given an element $\langle s_1, s_2^{\bar{t}}\rangle$, the matching chain C that contains (that is generated by) $\langle s_1, s_2^{\bar{t}}\rangle$ is uniquely identified. A transformation $\bar{t}$ generates a set of (disjoint) matching chains. The main idea behind the fusion of segments is that each matching chain (i.e., each set of matching segments) is substituted in the final map by a single polyline. Therefore, the final map is obtained by adding the polylines that represent the matched segments to the unmatched segments of S_1 and $S_2^{\bar{t}}$. The problem is thus reduced to build a polyline that approximates the segments in a matching chain C. With this polyline, it is easy to smoothly connect the different segments inserted in the final map.

The solution to the above problem consists in iteratively building a sequence of approximating polylines $P_0, P_1, \ldots$ that converges to the polyline P that adequately approximates (and substitutes in the resulting map) the matching segments in C. The polyline P_0 is composed of a single segment connecting the pair of farthest points in C. Given the polyline P_{n-1}, call s the segment in (a pair belonging to) C that is at maximum distance from its (closest) corresponding segment $\bar{s}$ in P_{n-1}. If the distance $d(s, \bar{s})$ 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 EXPERIMENTAL RESULTS

The method we presented has 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 $50cm$. For our purposes we chose to acquire scans with angular resolution of 1 degree and with 180 degrees angular range. As already said, each scan obtained by the laser range scanner has been processed to find a set of segments that approximate the points returned by the sensor, according to the algorithm described in (Gonzáles-Baños and Latombe, 2002). The method presented in this paper has been coded in ANSI C++ employing LEDA libraries 4.2 (LEDA Library, 2004) for two-dimensional geometry and has been run on a $1GHz$ Pentium III processor with Linux SuSe 8.0. We considered 31 pairs of scans (from $S_1 - S_2$ to $S_{31} - S_{32}$) that have been acquired by driving the robot manually and without recording any odometric information. The scans have been collected in a laboratory, a very scattered environment, in a narrow hallway with rectilinear walls, and in a department hall, a large open space with long perpendicular walls. The correctness of the scan matches has been determined by visually evaluating the initial scans and the final map with respect to the real environment. For every scan match, we tested the basic method and the three heuristics, sometimes modifying the values of the parameters.

In general, our experimental results demonstrate that our method performs very well: 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

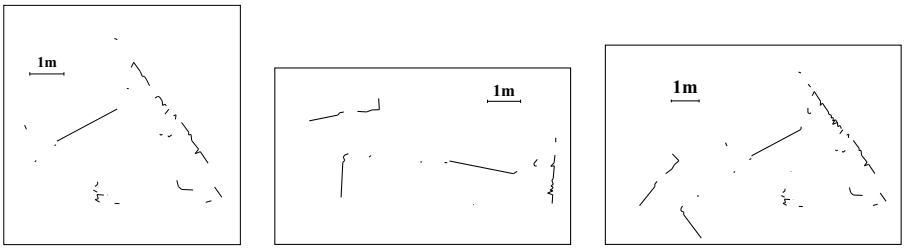

Figure 4: Matching scans S_4 (left) and S_5 (center) to get final map $S_{4,5}$ (right)

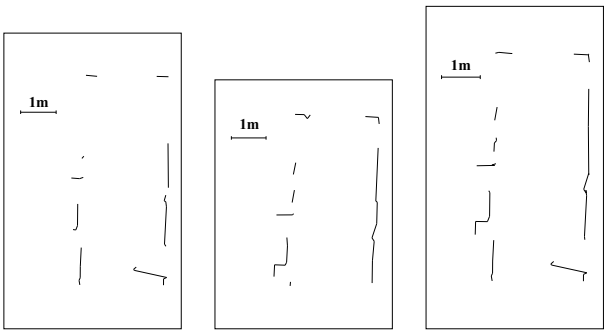

Figure 5: Matching scans S_{18} (left) and S_{19} (center) to get final map $S_{18,19}$ (right)

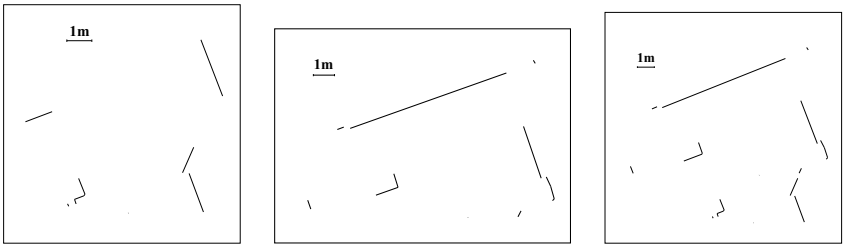

Figure 6: Matching scans S_{25} (left) and S_{26} (center) to get final map $S_{25,26}$ (right)

in all three kinds of environment, even if sometimes it needs some parameter adjustments.

Table 1: Summary of experimental results for the different methods of computing transformations between two scans

	S_4	S_5	S_{18}	S_{19}	S_{25}	S_{26}
# of segments	47	36	24	24	10	12
All	936 s [41260][1]		32 s [3096]		0.38 s [231]	
Consecutive	1.25 s [2]		0.73 s [27]		0.13 s [4]	
Random[2]	7.69 s		2.51 s		0.78 s	
Histogram	3.29 s [73]		1.97 s [192]		0.15 s [32]	

[1] [Number of possible transformation that have been evaluated]
[2] Obtained by generating about 20000 angles

In the following we discuss some interesting scan matches (Table 1). S_4 and S_5 were taken inside the laboratory: they contain a large number of short segments since the environment is highly scattered and the presence of chair and table legs produces an high number of small segments (Fig. 4). The heuristic that works better is that based on consecutive segments: it was able to find a good transformation evaluating only two transformations; on the other hand, the evaluation of all the possible transformations is infeasible (over 40000 matches to evaluate!). S_{18} and S_{19} were taken along the hallway: they contain fewer segments than the scans previously discussed and are characterized by long rectilinear segments (Fig. 5). Even in this case, evaluating all the transformations is expensive, while the consecutive-segments heuristic performs well. Finally, S_{25} and S_{26} were taken in the hall: they contain only few segments since the environment is characterized by long rectilinear and perpendicular walls (Fig. 6). All the heuristics perform well in this case because, starting from a small num-

ber of segments, there are only few transformations that are easy evaluate.

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. 7, the scans, taken in the lab entrance, 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. 8 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.

We now briefly discuss the role of the parameters that mostly influence the performance of our method. POINTDISTANCETOLERANCE influences the matching value of two segments and the transformation discharging. In the same way, large values for SEGMENTDISTANCETHRESHOLD make segments that do not represent the same object in the environment to match; small values reduce the number of matching segments thus making the method more sensitive to measurement errors. Large values of ANGLEDIFFERENCETOLERANCE facilitates the search of the best transformation by allowing a lot of possible transformations to be considered (but their evaluation requires more time). Small values of SEGMENTLENGTHPERCENTAGE lead to consider more short segments and to generate more transformations to evaluate, while large values allow to consider only longer segments in the map, this is useful in the case of long segments representing walls.

5 CONCLUSIONS

We have described a powerful method for scan matching that works without any information about the relative positions of the two scans but relies exclusively on the geometrical features of the scans. This is the major feature which distinguishes our method from most of the scan alignment and matching methods reported in the literature. Extensive experimental results validate the effectiveness of the approach.

We are working to apply the method described in this paper to the integration of n scans. This problem will be tackled in two steps: initially we will try to integrate a *sequence* of n scans and then we will try to extend the techniques to the integration of a general set of n scans, acquired by a single robot or by different robots.

Acknowledgments

The authors would like to thank Jean-Claude Latombe for his generous hospitality at Stanford University where this research was started, Héctor Gonzáles-Baños for sharing his programs and expertise with collecting laser range scan data, Paolo Mazzoni and Emanuele Ziglioli for the initial implementation of the fusing algorithm. Francesco Amigoni was partially supported by a Fulbright fellowship and by "Progetto Giovani Ricercatori" 1999 grant.

REFERENCES

Bengtsson, O. and Baerveldt, A.-J. (1999). Matching of laser range-finder scans in changing environments. In *Proceedings of the European Workshop On Advanced Mobile Robots*, pages 103–109.

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.

Cox, I. (1991). Blanche: An experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transaction on Robotics and Automation*, 7(2):193–204.

Einsele, T. (1997). Real-time self-localization in unknown indoor environments using a panorama laser range finder. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 697–702.

Fenwick, J. W., Newman, P. M., and Leonard, J. J. (2002). Cooperative concurrent mapping and localization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1810–1817.

Gonzáles-Baños, H. H. and Latombe, J. C. (2002). Navigation strategies for exploring indoor environments. *International Journal of Robotics Research*, 21(10-11):829–848.

Gutmann, J. S. and Schlegel, C. (1996). Amos: Comparison of scan matching approaches for self-localization in indoor environments. In *Proceedings of the Euromicro Workshop on Advanced Mobile Robots*, pages 61–67.

LEDA Library (2004). http://www.algorithmic-solutions.com/enleda.htm.

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.

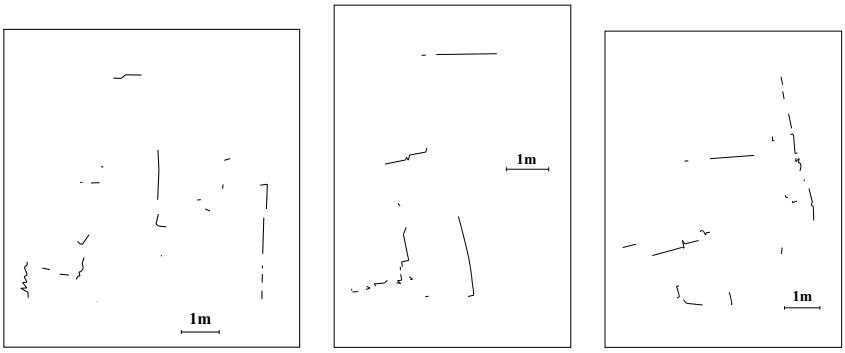

Figure 7: Scans S_1 (left), S_2 (center), and S_3 (right) taken in the lab entrance

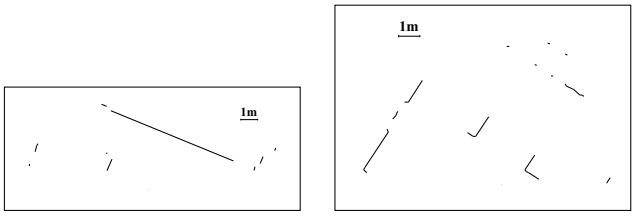

Figure 8: Scans S_{27} (left) and S_{28} (right) taken in the hall

Martignoni III, A. and Smart, W. (2002). Localizing while mapping: A segment approach. In *Proceedings of the Eighteen National Conference on Artificial Intelligence*, pages 959–960.

Röfer, T. (2001). Building consistent laser scan map. In *Proceedings of the European Workshop On Advanced Mobile Robots*, pages 83–90.

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 *Proceedings of the National Conference on Artificial Intelligence*, pages 852–858.

Stewart, B., Ko, J., Fox, D., Konolige, K., and Limketkai, B. (2003). A practical, decision-theoretic approach to multi-robot mapping and exploration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3232–3238.

Thrun, S. (2001). A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20.

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 *Proceedings of the IEEE International 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 *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 12–16.

Williams, S., Dissanayake, G., and Durrant-Whyte, H. (2002). Towards multi-vehicle simultaneous localisation and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 2743–2748.

Zhang, L. and Ghosh, B. (2000). Line segment based map building and localization using 2D laser rangefinder. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2538–2543.