

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

Maria Gini
Dept of Computer Science and Engineering, University of Minnesota
Minneapolis, USA

Keywords: 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 original aspect of our algorithm is that it does not require any information on the position where the scans were collected but uses only geometrical features of the scans.

1 INTRODUCTION

The increasing use of mobile robots equipped with laser range scanners has stimulated the development of methods for matching and 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 Milius, 1997; Cox, 1991; Röfer, 2001). Some methods (Einsele, 1997; Martignoni III and Smart, 2002) do not use information coming from odometry, but they work well only in rectilinear environments and for small displacements of the robot.

In this paper we present a method for matching two scans that does not require *any* odometry information and that works for *significant* displacements of the robot. 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 SICK LMS 200 laser range scanner (mounted on a Robuter mobile platform at a height of approximately 50 cm) a sequence of distance measurements along directions separated by a programmable angle (1° , in our case) sweeping 180° . 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 (González-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 an indoor 2D surface and that walls and vertical objects are at the height of the laser scan.

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

1. *find 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 $S_{1,2}$.

This paper is structured as follows. The next Section describes in detail our method, while in Section 3 we discuss the experimental activity performed to validate it. Section 4 concludes the paper.

2 THE PROPOSED METHOD

In the algorithms, two points are considered to coincide when they are closer than POINTDISTANCE-TOLERANCE (in our experiments we set this parameter to 15 mm) and two angles are considered equal when their values differ for less than ANGLEDIFFERENCE-TOLERANCE (in our experiments we set this parameter to 0.2 rad).

2.1 Finding Transformations

This step, given the scans S_1 and S_2 , first finds the angles between the segments in S_1 and between the seg-

ments 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 . 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. They 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 normal, the errors introduced by the sensor and by the algorithm that approximates points with segments alter the representation of these angles.

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 it-

erate with $k = 1, \dots, K$. In 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.

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 ($i = 0, 1, \dots, nslots - 1$) contains the segments with orientation comprised between $\pi \times i/nslots$ and $\pi \times (i + 1)/nslots$, 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 $\pi/2$ rad away from the principal direction. 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 .

2.2 Evaluating 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 s_1 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 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 $d(s_1, s_2) = \min(\max(\text{dist}(s_1, \text{start}(s_2)), \text{dist}(s_1, \text{end}(s_2))), \max(\text{dist}(s_2, \text{start}(s_1)), \text{dist}(s_2, \text{end}(s_1))))$ where $\text{start}(s)$ and $\text{end}(s)$ are the extremes of segment s . 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 when the two segments are too far away, namely when $d(s_1, s_2^t)/\text{POINTDISTANCETOLERANCE} > \text{SEGMENTDISTANCETHRESHOLD}$. $\text{SEGMENTDISTANCETHRESHOLD}$ 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 $\text{SEGMENTLENGTHREFUSE}$; 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 $O(n_1 n_2)$.

2.3 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 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 $C = \{\langle s_1, s_2^{\bar{t}} \rangle \mid s_1 \in S_1 \text{ and } s_2^{\bar{t}} \in S_2^{\bar{t}} \text{ have a positive matching value for } \bar{t}\}$ 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, \dots 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 .

3 EXPERIMENTAL RESULTS

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

Table 1 shows the results obtained for three interesting scan matches (see also Fig. 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. S_{18} and S_{19} were taken along the hallway: they contain fewer segments than the previous scans

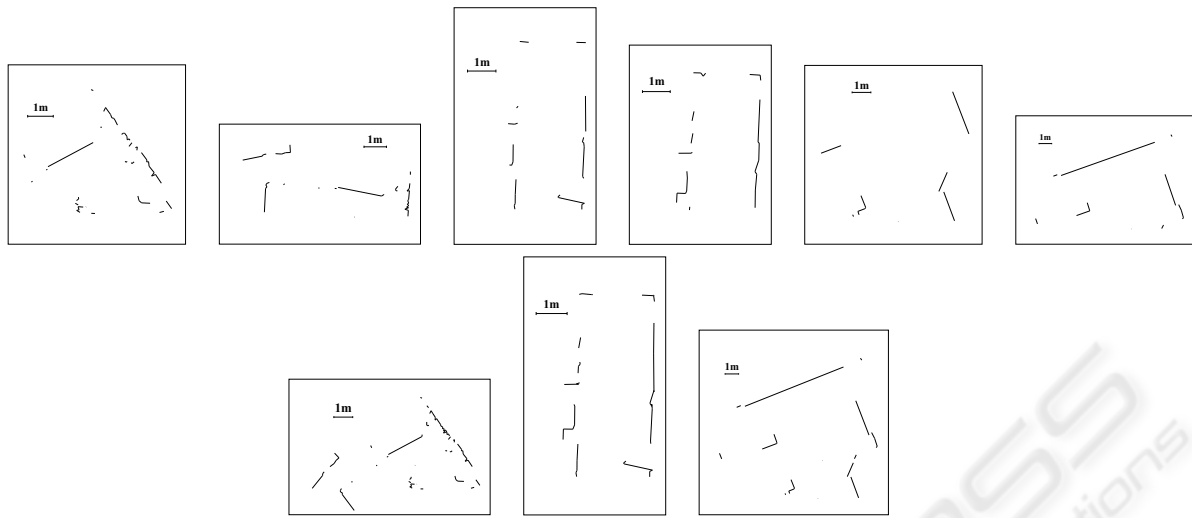


Figure 1: Top, left to right: scans S_4 , S_5 , S_{18} , S_{19} , S_{25} , and S_{26} ; bottom, left to right: final maps $S_{4,5}$, $S_{18,19}$, and $S_{25,26}$

and are characterized by long rectilinear segments. 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.

Table 1: Some experimental results

	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 transformations that have been evaluated]

² Obtained by generating about 20000 angles

4 CONCLUSIONS

We have described a 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 validated the effectiveness of the approach. We are working to apply this method to the integration of n scans.

ACKNOWLEDGEMENTS

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 Mazzone 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, Maria Gini by NSF under grants EIA-0224363 and EIA-0324864.

REFERENCES

Cox, I. (1991). Blanche: An experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Trans Rob Autom*, 7(2):193–204.

Einsele, T. (1997). Real-time self-localization in unknown indoor environments using a panorama laser range finder. In *Proc. IROS*, pages 697–702.

González-Baños, H. H. and Latombe, J. C. (2002). Navigation strategies for exploring indoor environments. *Int J Robot Res*, 21(10-11):829–848.

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.

Martignoni III, A. and Smart, W. (2002). Localizing while mapping: A segment approach. In *Proc. Eighteen Nat. Conf. on AI*, pages 959–960.

Röfer, T. (2001). Building consistent laser scan map. In *Proc. EUROBOT*, pages 83–90.