

Geometric Robot Mapping

Rolf Lakaemper¹, Longin Jan Latecki¹, Xinyu Sun²,
and Diedrich Wolter³

¹ Temple University, Philadelphia, PA 19122, USA
{latecki, lakamper}@temple.edu

² Texas A&M University, College Station, TX 77840, USA
xsun@math.tamu.edu

³ Universität Bremen, Bremen, Germany
dwolter@informatik.uni-bremen.de

Abstract. The purpose of this paper is to present a technique to create a global map of a robot's surrounding by converting the raw data acquired from a scanning sensor to a compact map composed of just a few generalized polylines (polygonal curves). To merge a new scan with a previously computed map of the surrounding we use an approach that is composed of a local geometric process of merging similar line segments (termed Discrete Segment Evolution) of map and scan with a global statistical control process. The merging process is applied to a dataset gained from a real robot to show its ability to incrementally build a map showing the environment the robot has traveled through.

Keywords: Robot Mapping, Polygon Merging, Polygon Simplification, Perceptual Grouping.

1 Introduction

Imagine a scenario where a robot explores an unknown terrain. The goal is to acquire in real time a global overview map integrating all measurements collected by the robot. Here we deal with measurements obtained by 2D range sensors, called scans, that represent partial top views of the robots environment. Building a global overview map from scans is a typical scenario for rescue robots, where the overview knowledge, in the form of a global map, is particularly important to localize victims in catastrophe scenarios (e.g., in collapsed buildings) and to ensure that the whole target region has been searched [4]. Since odometry information under such conditions is very unreliable, we assume that it is not available. Also landmarks are ambiguous.

The whole process from reception of raw scanning data to the final map refers to the problem of simultaneous localization and mapping (SLAM) in robotics. The proposed approach addresses two main problems in SLAM stated in [10].

1. The measurement errors are statistically dependent, since errors in control accumulate over time, and they affect the way future sensor measurements are interpreted.
2. The second complicating aspect of the robot mapping problem arises from the high dimensionality of the entities that are being mapped, which leads to serious runtime and storage problems.

We address the problem of measurement errors being statistically dependent with a new process of map merging that is based on geometric local process of line segment merging with a global statistical control.

The second problem arises from the fact that in most mapping approaches the objects of which maps are built are simply points. These are either directly scan reflection points or point landmarks, e.g., [2] and [10]. In some approaches simple geometric features, especially line segments [7, 9, 1] are used. However, the maps are still composed of a huge numbers of them, since these approaches do not provide any mechanisms to incrementally reduce the number of building blocks, which can be line segments of simply points. Consequently, the obtained maps are composed of thousands or even millions of points or line segments. An example of such a map is shown in Fig. 1(a). It is composed of 144400 points and obtained by alignment of 400 scans. It is then clear that such maps lead to serious runtime and storage problems, e.g., it is impossible to map larger environments and to perform loop closing in real time. In our map representation, we simply do not run into the second problem. Our representation is built of higher level objects, which are line segments and generalized polylines, and we have an explicit process, called Discrete Segment Evolution, that reduces the number of line segments to a minimal number required to represent the mapped environment. An example map obtained by our approach is shown in Fig. 1(b). This map was obtained from the same scan data as the map in (a), and it is composed of only about 50 line segments (which amounts to about 100 end-points). Videos illustrating our incremental mapping results can be viewed on <http://knight.cis.temple.edu/~shape/robot/>.

A nice probabilistic framework to construct a global map from scan data is presented in [8]. However, this framework is based on the assumption that the uncertainty of scan points' positions is known. Due to the dependence of laser scan measurements on surface characteristics of scanned objects, e.g., glass-like surface, brick wall, and metal surface, this assumption is not satisfied in our example of rescue robots. We approach the problem of constructing a global map using the principles of perceptual grouping, which look for geometric structures in the data without any assumptions about the error characteristics [6].

2 Robot Mapping

In this section we introduce some notation regarding the system used by a robot to create its global map of its surroundings, and we summarize the main steps performed at each iteration of the algorithm, i.e., on the arrival of a new scan.

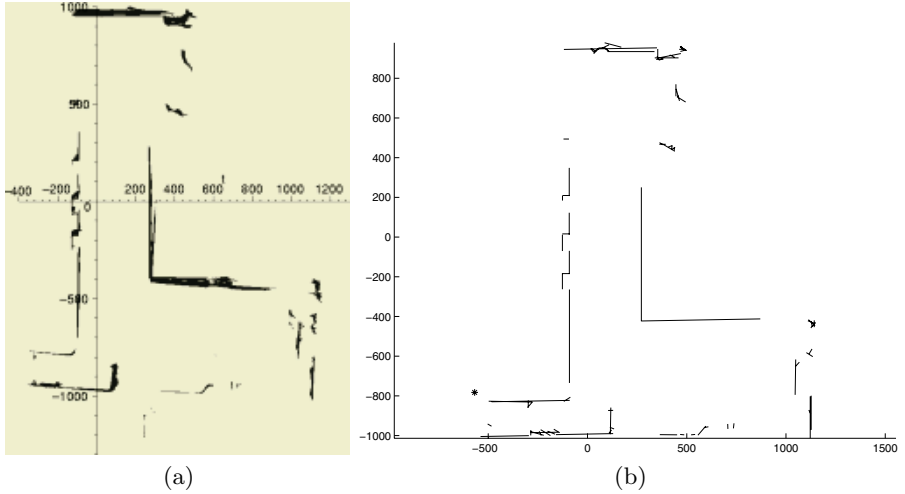


Fig. 1. (a) A global map obtained by alignment of 400 scans is composed of 144400 points. (b) A global map obtained by the proposed map merging algorithm is composed of about 50 line segments (100 endpoints). Both maps are obtained from the same laser range data set showing a hallway at the Univ. of Bremen

The output is a global map that represents a top view of the environment using a small number of polylines. Fig. 1(b) illustrates such a map. For comparison, a global map obtained by alignment only is shown in Fig. 1(a).

The proposed algorithm merges the newest laser range scan S_t at time t with a global map G_{t-1} built from previous scans 0 to $t-1$. The global map is produced incrementally, which means that at every time t we have a ready to use and a very simple global map of the environment. This is very important for all navigation and mapping tasks. Both global map G_t and scan S_t are composed of *generalized polylines*.

A *generalized polyline* is a set of line segments, having a specific ordering, whose vertices may or may not be connected. Observe that a classical definition of a polyline (polygonal curve) requires that the endpoints of consecutive segments coincide. Generalized polylines result naturally when scan points are approximated with line segments, which is our first processing step of the input range data. By dropping the constraint that a polyline be composed of line segments whose vertices are connected, we do not introduce additional noise that would result from connecting these vertices. The usage of generalized polylines is particularly important in the polyline merging and shape similarity algorithms described below.

Our first processing step (approximation of scan points with line segments) is followed by the segment grouping step. We form an ordered list of segments by minimizing the sum of the distances of their endpoints. Finally, if the endpoints

of consecutive segments are too far apart, we split the list into sublists. Thus, generalized polylines are sublists of this list.

To create a global map G , we start with the first global map G_1 being equal to the first scan S_1 . Henceforth, assuming we have created the global map G_{t-1} at time $t-1$ and a new scan S_t has arrived, G_t is created in the following three steps:

Correspondence: We use a very simple and common approach to establish correspondence between global map G_{t-1} and a new scan S_t . We position the new scan S_t at the pose of the previous scan S_{t-1} that has been aligned to the previous global map G_{t-2} (in the process of construction of the actual global map G_{t-1}). The pose is the position and rotation angle in the coordinates of the global map. Then the correspondence is established by mapping each segment endpoint of S_t to a closest point in G_{t-1} and the same with roles of S_t and G_{t-1} interchanged.

This process of correspondence computation works fine if there is continuity in the robot pose, i.e., robot pose changes only slightly from scan to scan. If the continuity assumption is not satisfied, we use shape similarity to establish the correspondence [5] and [11]. However, this approach is outside the scope of this paper.

Alignment: The current scan S_t is rotated and translated until a minimum distance is found between the corresponding points. Then the closest points are found again, and the whole process is repeated until it stabilizes. This algorithm to align scan S_t and map G_{t-1} is called Iterative Closest Point (ICP), and is described in [3].

Merging: This is the main contribution of this paper and its detailed discussion follows in Section 2.1. The output of alignment overlays the actual scan on the global map, but the surfaces of the same objects are still represented by separate polylines. The goal of merging is to represent surfaces of the same objects by single polylines.

2.1 Merging

Merging is the task of combining similar segments taken from two aligned maps to form new segments in a joint map. The similarity between pairs of segments is modeled following principles of perceptual grouping. In the case of incremental building of a global map, the task of merging is to combine similar line segments of the new scan, S_t , and the previous global map, G_{t-1} , to form new segments that define a new and current global map G_t . We assume that S_t has been aligned to G_{t-1} . Merging consists of two steps, which integrate the new information contained in S_t with the previous global map to produce G_t . The two steps are restrictive pairing and simplification.

Restrictive Pairing: The result of pairing can intuitively be understood as *visual average* of S_t and G_{t-1} . Since our goal is to combine the information from

both maps, we allow only pairing of segments from different maps. Therefore, we define two classes of line segments, class C_1 is a list consisting of segments from G_{t-1} and class C_2 is a list consisting of segments of S_t .

Although the final task of merging is to decrease the number of line segments by combination, the pairing step goes into the opposite direction: it might create many new segments, which will be simplified in the second step called simplification. Pairing can be compared to a pencil drawing technique known as sketching, e.g., used for cartoon drawings: to find the final outline of an object, it is first approximated by a larger number of light strokes, giving the eye the opportunity to imagine and select the correct position.

The process creates all possible pairs of line segments that are sufficiently similar, taking one segment from C_1 and one from C_2 . Pairing of two segments from the same class is not allowed.

The similarity of line segments L_1 and L_2 is measured with the cost function $C(L_1, L_2, ad)$ (defined below), where ad is an angular direction given by global statistics (defined below). If $C(L_1, L_2, ad)$ is below a given threshold, we create a new line segment $ms(L_1, L_2, ad)$ (defined below) that is visually close to L_1 and L_2 . Figure 2 shows an example of restricted pairing. The newly created segments must follow the main directions, i.e., they are only allowed to have angles of 0° , $+60^\circ$, -60° with the x axis.

A single line segment L_1 can create many *children* line segments $ms(L_1, L_2^i, ad)$ by pairing with segments L_2^i for $i = 1, \dots, n$. We need to allow a single segment to pair with more than one segment from the other class, since we do not know the exact segment correspondence. It might be that the correct shape feature is created by the pairing with a second partner. However, to limit the computational complexity, each line segment is allowed to create only a small number of segments (at most 3 children segments are allowed in our implementation).

We remove all line segments from C_1 and C_2 that were parents of at least one new segment. We denote the resulting lists by C'_1 and C'_2 . We denote with A_t a list of all resulting children together with C'_1 and C'_2 . Formally, the output of restricted pairing is defined as (T_p is a pairing cost threshold):

$$A_t = C'_1 \cup C'_2 \cup \{ms(L_1, L_2, ad) : C(L_1, L_2, ad) < T_p, L_1 \in C_1, L_2 \in C_2\}.$$

Restrictive pairing may create some small artifacts in addition to features present in the reality, such as parallel segments in Fig. 2(b). The artifacts may be introduced, since we do not know the exact correspondence of line segments, and therefore, must allow a single line segment to pair with many line segments in the other class. We therefore need a cleaning process to remove these artifacts, or, in analogy to the sketching example mentioned above, a process that selects or creates an appropriate precise set of strokes based on the approximation. This process is called *simplification* and it is the second step in the merging process. Its result is illustrated in Fig. 2(c).

Simplification: The input is the joint map A_t created by restrictive pairing. Simplification can be viewed as cleaning process after pair creation to create a smaller set of possibly new segments, being the final merging result. To use the

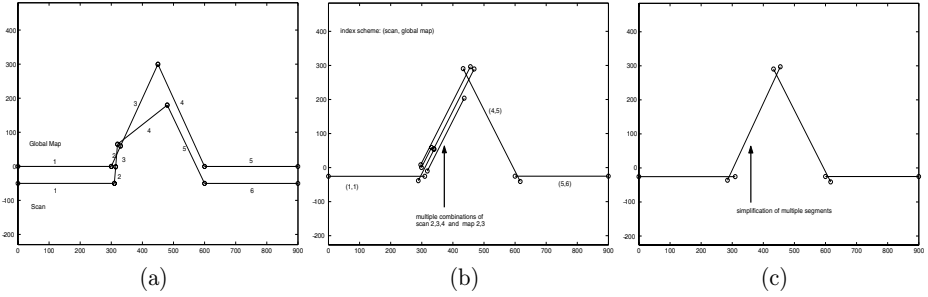


Fig. 2. The figure (b) is obtained from (a) by the proposed pair creation process. The index pairs in (b) refer to parent segments in (a). (c) shows the final result obtained by simplification of (b). The newly created segments in (b) and (c) must follow the main directions of $0^\circ, +60^\circ, -60^\circ$ with the x axis

sketch analogy again, the simplification process creates a single line by visually averaging the approximating bundle of strokes. Pairs of line segments are merged together to form new segments using the same merging process ms and cost function C but with different constraints. Simplification is done without any class restriction, i.e., a line segment in A_t can pair with any other segment in A_t .

The main difference is that the simplification process has a global control mechanism: We iteratively merge a pair whose merging cost C is the lowest at each pass. More precisely, the segment pair L_1, L_2 with lowest cost $C(L_1, L_2, ad)$, for one of the main angular directions ad , is merged in each pass. This means that $ms(L_1, L_2, ad)$ is inserted to list A_t , and L_1, L_2 are removed from A_t . Thus, each line segment can have at most one child. The process stops when the lowest cost is above a threshold T_s . The resulting simplified version of A_t is the new global map G_t . Fig. 3(c) shows the simplified version of line segments in (b). Observe that both merged shape features from (a) are preserved (the straight line and the tent). This is acceptable, since given the input as in Fig. 3(a), we

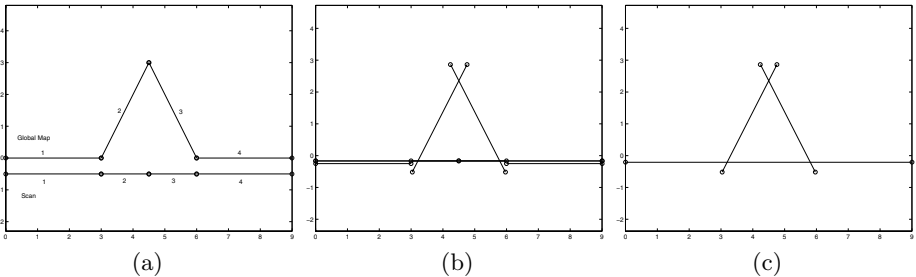


Fig. 3. The figure (b) is obtained from (a) by the proposed pair creation process. The index pairs in (b) refer to parent segments in (a). (c) shows the result simplification of (b). The newly created segments in (b) and (c) have to follow the main directions of $0^\circ, +60^\circ, -60^\circ$ with the x axis

cannot decide which feature is the right one. This decision can be, however, made after merging several consecutive scans.

The main angular directions for simplification are computed with the same global statistics as in pairing process. This allows us to cope with accumulative errors. Section 2.2 gives more details on global statistics.

2.2 Main Directions

The main directions ad are obtained as significant peaks in the direction histogram of line segments (angles with the x axis) in $G_{t-1}(tm)$, where $G_{t-1}(tm)$ is a global map G_{t-1} restricted to line segments created in at most tm time steps ago. tm is a *time memory* factor. This restriction implies that only part of global map recently created determines the main directions. Therefore, the choice of directions is dynamic: the set of main directions ad is created with respect to the most recent part of the global map. This dynamic process assures the robustness of the algorithm with respect to main directions, e.g., for indoor environments, while simultaneously being flexible enough to react to changing or even non present main directions, e.g., outdoor or natural environments.

Each line segment in the restricted global map $G_{t-1}(tm)$ contributes to the bin representing its direction with the weight given by its length. This histogram is cyclic, e.g., a window of size 5 around bin 1 contains bins 179, 180, 1, 2, 3.

An important feature of our approach is the interaction between main directions and merging: Let us assume we have a significant number of segments in S_t , all having a similar direction that is significantly different from main directions present in $G_{t-1}(tm)$, i.e., these segments do not define a main direction yet. For example, this is the case if the robot starts to perceive a new surface.

Since the new segments follow a significantly different direction, the cost function C assures that they create no children, because the pairing cost is above the threshold T_p , i.e., the transformation of the parent segments to child segments is too expensive. Thus, the original segments will remain in A_t . Since A_t , after simplification, becomes the new global map G_t , the new segments will lead to a peak, and will open a new main direction in the direction histogram. Consequently, pairing and simplification in the new direction will be allowed. A real example will be given in Section 3.

The direction histogram represents the statistical distribution of line segment directions. This statistical control provides a solution to the problem of cumulative errors (Problem 1 in the introduction). Cumulative errors introduce systematic distortions in the directions of line segments that accumulate slowly. The main issue is that accumulative errors do not lead to peaks in the direction histogram, and consequently, appearing line segments are correctly mapped to the existing main directions. On the other hand, as we have just described, if a surface of a new object is oriented into a new direction, it will lead to a peak in the direction histogram after a few scans of the surface have been acquired. This solution is based on quantization of the angular directions. Thus, a new main direction is created in the direction of the new surface if the difference

between the new direction and the existing main directions is larger than the quantization factor, which is determined by the thresholds T_p and T_s . The fact that the direction histogram provides a solution to the problem of cumulative errors is also true for other histogram-based approaches, e.g., [9].

2.3 New Line Segment Creation and Cost Function

This section will describe the merging function ms and the associated cost function C used for the pairing and simplification process. The merging function is the most important module in the merging system, since it is responsible for the creation of the segments finally seen in the new map. Given a pair of line segments, L_1 and L_2 , and the angular direction ad , it computes a merged segment $ms(L_1, L_2, ad)$ with the angular direction ad . The cost function $C(L_1, L_2, ad)$ is responsible for the filtering step: it produces the basic values for the decision if a created segment will be accepted or rejected as a member of the approximated map A_t or the new global map G_t . It measures the similarity of L_1 and L_2 in the context of the main direction ad .

The geometric intuition of the presented merging process and, in particular, the definition of merging cost $C(L_1, L_2, ad)$ is based on cognitively motivated principles of perceptual grouping. We followed the approach presented in [6] on grouping line segments to form longer line segments. It states that proximity of endpoints, parallelism, and collinearity are the main geometric relations that influence the perceptual grouping of line segments. Our setting is slightly different, since we merge two line segments only with respect to a given main direction ad . Therefore, we developed a new cost function. As mentioned above, the usage of main directions is necessary to cope with cumulative errors.

Before we explain the meaning of the perceptual grouping principles in our setting, we need to introduce one more concept of a straight line ld that follows one of the main directions ad . Let two line segments L_1 and L_2 and a main angular direction ad be given. A first step in our cost computation is to position a line following direction ad between two line segments L_1 and L_2 . The straight line ld with direction ad is positioned between L_1 and L_2 so that the equation

$$d_1 \cdot l_1 = d_2 \cdot l_2$$

is satisfied (see Figure 4), where l_i is the length of segment L_i and d_i is the distance of the midpoint of L_i to line ld for $i = 1, 2$. The **merged segment** $ms(L_1, L_2, ad)$ is defined by the convex hull of the projections of L_1 and L_2 on line ld . It is the segment from P'_1 to P'_4 in Figure 4.

Now we can explain the meaning of the perceptual grouping principles in our setting.

- **Parallelism:** The greater the angles between L_1 and L_2 , and between L_1 and L_2 and $ms(L_1, L_2, ad)$, the greater the cost of merging them together. Likewise, the angle difference of longer segments have more weight than shorter ones.

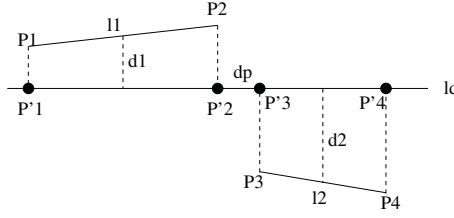


Fig. 4. Geometric illustration of the cost function $C(L_1, L_2, ad)$

- **Collinearity:** The greater the distance of the endpoints of L_1 and L_2 from the target line ld , the higher the value of the cost function.
- **Proximity:** The greater the distance between the projections of L_1 and L_2 on line ld , the higher the value of the cost function.

Finally we can define the new cost function that integrates the three perceptual grouping principles. Given the line ld , the cost of merging L_1 and L_2 to $ms(L_1, L_2, ad)$ is defined by the following measure that incorporates our realization of the perceptual grouping principles (see Figure 4):

$$C(L_1, L_2, ad) = \frac{lr \left((l_1 + 1) \frac{d(P_1, ld) + d(P_2, ld)}{1 + K \cos(a_1)} - l_1 \frac{d(P_1, ld) + d(P_2, ld)}{1 + K} \right)}{1 + K \cos(a_{12})} + \frac{lr \left((l_2 + 1) \frac{d(P_3, ld) + d(P_4, ld)}{1 + K \cos(a_2)} - l_2 \frac{d(P_3, ld) + d(P_4, ld)}{1 + K} \right)}{1 + K \cos(a_{12})}$$

where a_1 is the angle between L_1 and ld , a_2 is the angle between L_2 and ld , a_{12} is the angle between L_1 and L_2 , P_1, P_2 are endpoints of L_1 , P_3, P_4 are endpoints of L_2 , $d(P_i, ld)$ is the distance between point P_i and line ld . The constant K depends on the metric units used, and needs to be adjusted to obtain a balance between angular and metric units. The length ratio

$$lr = \frac{l(ms(L_1, L_2, ad))}{l(p(L_1)) + l(p(L_2))}$$

is the quotient of the length of the merged line segment $ms(L_1, L_2, ad)$ to the sum of the length of the projections $p(L_1)$ and $p(L_2)$ of line segments of L_1 and L_2 on line ld .

3 Implementation Details

3.1 Merging

Pairing: As described in Section 2.1, the merging consists of pairing followed by simplification. The pairing step creates a rough approximation of the new map,

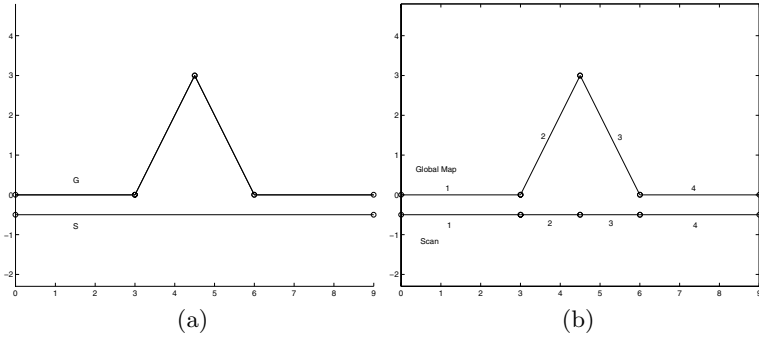


Fig. 5. New vertices are inserted in S in (a) as projections of vertices in G . The vertex insertion is necessary to make the correspondence of line segments possible

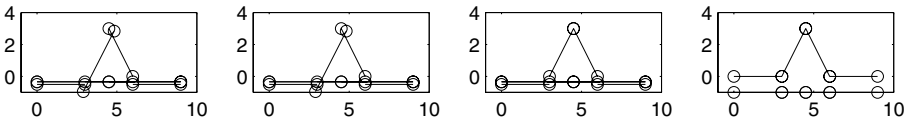


Fig. 6. Input map is from Figure 5(b). Pairing with different threshold T_p and global direction control with main directions of $0, +62^\circ, -62^\circ$. From left to right we have $T_p = 0.05, 0.04, 0.03, 0.01$

and it must take into account features in both the scan and the global map, even if a certain feature is only present in one of them. An example is given in Fig. 5, where the triangle feature is only present in the global map G . To permit appropriate combinations with segments of this feature, the straight segment of the scan S is split up to create segments corresponding to the feature segments of the global map. In order to do so a correspondence between endpoints of line segments is established. For every endpoint E in G , we find a closest point $p(E)$ in S . If distance from E to $p(E)$ is below a predefined threshold and $p(E)$ is not an existing end point in S , then $p(E)$ is inserted to S , splitting an existing line segment in S into two collinear line segments (that meet at $p(E)$). We perform the same for every endpoint E in S . The maps modified this way are the input to pair building.

The creation of descendants in the pairing process is dependent on the pairing threshold T_p . Fig. 6 shows the output of the pairing with global direction control of the input shown in Fig. 5 with different thresholds T_p , leading to different constellations: The leftmost map consists of newly created segments only. In the second figure the triangle segments could only pair once. The rightmost figure finally did not change the input; the threshold was too low to create any new segments.

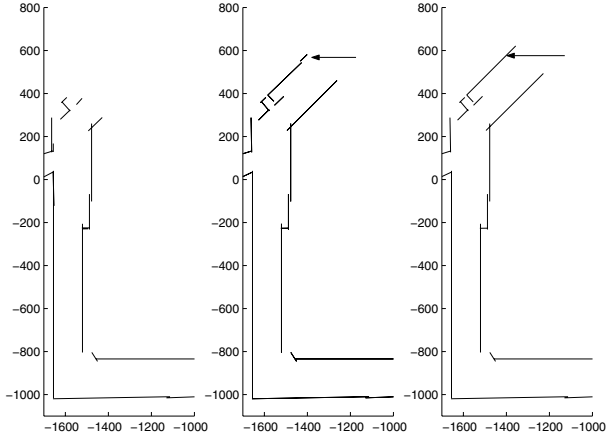


Fig. 7. Bremen hallway experiment: a new main direction is detected in the right figure and used for merging

3.2 Main Directions

The following section shows an example for the interaction between the merging process and the dynamic detection of main directions. The basic idea is not to merge segments with a cost above T_p or T_s respectively, but to keep them in their original position/direction. Such segments can be described as outliers with respect to the current main directions ad . If these outliers become dominant, they will open a new main direction and will be merged. This process can be observed in Fig. 7 showing a section of an experiment with real laser finder data collected in a hallway at the University of Bremen. The robots' position is in the upper left corner, the robot moves up right and scans backward, i.e., down left.

In the left figure the robot approaches a 45° corner, the current main directions consist of 0° and 90° angles with the x -axis. New 45° segments are not simplified but transferred unchanged into the new global map. New 45° segments still do not have enough weight to create a new direction in the center figure. Additionally the 0° walls are out of sight, and therefore, the histogram loses this entry (which does not effect any segments in this situation). The only main direction remaining is 90° . In the right figure, new 45° segments finally create a peak in the direction histogram. Hence the new main directions are now 90° and 45° . Consequently, the new segments are merged together and simplified following the new 45° direction.

4 Conclusions and Future Work

We presented a novel approach to robot mapping, based on an iterative process that merges similar line segments of subsequent scans. The process is locally controlled by a similarity of line segments, which is motivated by principles of

perceptual grouping. Additionally the process is controlled by global statistics, giving emphasis to the main directions present in the map to eliminate effects of accumulated errors of local scans. The experiments performed on real robot data show that the interaction between local and global control is able to successfully adapt to changes in directions without any pre-knowledge of the environment to build the environmental map given by subsequent scans. The merging process can be useful for any application where (visual) simplification of sets of line segments is needed; further experiments will include simplification of images gained by edge detection and hand drawn sketches for stroke recognition.

Acknowledgment

This work was supported in part by the National Science Foundation under grant INT-0331786 and grant R3 [Q-Shape] in the framework of the SFB/TR 8 Spatial Cognition from German Research Foundation (DFG). We would like to thank Sebastian Thrun for a helpful discussion regarding the topic of this paper. Thomas Röfer is acknowledged for providing scan data.

References

1. Ingemar J. Cox. Blanche: Position estimation for an autonomous robot vehicle. In Ingemar J. Cox and G.T. Wilfong, editors, *Autonomous Robot Vehicles*, pages 221–228. Springer-Verlag, 1990.
2. G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions of Robotics and Automation*, 2001.
3. A. Fitzgibbon. Robust registration of 2d and 3d point sets. In *Proc. British Machine Vision Conference, volume II, Manchester, UK*, pages 411–420, 2001.
4. A. Jacoff, E. Messina, and J. Evans. Performance evaluation of autonomous mobile robots. *Industrial Robot: An Int. Journal*, 29(3), 2002.
5. Longin Jan Latecki, Rolf Lakämper, and Diedrich Wolter. Shape similarity and visual parts. In *Proceedings of the 11th International Conference on Discrete Geometry for Computer Imagery (DGCI), Naples, Italy*, November 2003.
6. D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395, 1987.
7. F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic Systems*, 1997.
8. S. T. Pfister, S. I. Roumeliotis, and J. W. Burdick. Weighted line fitting algorithms for mobile robot map building and efficient data representation. In *ICRA*, 2003.
9. T. Röfer. Using histogram correlation to create consistent laser scan maps. In *Proceedings of the IEEE International Conference on Robotics Systems (IROS-2002)*, 2002.
10. S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.
11. D. Wolter and L. J. Latecki. Shape matching for robot mapping. In Chengqi Zhang, Hans W. Guesgen, and Wai K. Yeap, editors, *Proc. of 8th Pacific Rim Int. Conf. on Artificial Intelligence*, Auckland, New Zealand, August 2004.