Proceedings of the 2000 IEEE
International Conference on Robotics & Automation
San Francisco, CA • April 2000

Line Segment Based Map Building and Localization Using 2D Laser Rangefinder

Li Zhang and Bijoy K. Ghosh

Department of Systems Science and Mathematics
Washington University, St. Louis, MO
lzhang@zach.wustl.edu, ghosh@zach.wustl.edu

Abstract

In this paper, we study a new scheme for map building and describe localization techniques for a mobile robot equipped with a 2D laser rangefinder. We propose to use line segments as the basic element for the purpose of localization and to build the map. Line segments do provide considerable geometric information about the scene that can also be used for accurate and fast localization. We introduce a new *Closed Line Segment (CLS)* map, which consists of only line segments and defines a closed and connected region. *Virtual line segments* are drawn, for the spots that do not adequately describe a line segment on the range data. These are further explored via navigation and we argue that the *CLS* map provides an efficient mobile robot exploration scheme. All these techniques have been implemented on our Nomad XR4000 mobile robot and results are described in this paper.

1. Introduction

Map building and localization are two fundamental problems in mobile robotics, and they are very closely connected.

There are basically two types of maps suitable for localization. The occupancy grid map, first introduced by Moravec and Elfes [3], has been widely used [7,8]. Occupancy grids represent the environment as a two dimensional array of cells, each of which indicates the probability of being occupied. It is easy to build, and especially suitable for imprecise range sensors like ultrasonic rangefinders. But it is hard to be used directly for localization and path planning.

Geometric primitive based map has been explored by many researchers [2,6]. Line segment is the most widely used feature. Since not every object provides line segments, these maps cannot give a closed and connected region like the occupancy grid maps.

An interesting fact to notice is that when building a map manually, it is customary to use a polygonal map that consists of only line segments. The reason lies in the fact that it is easy to manipulate and can describe most of the indoor environments.

We introduce a new map called *Closed Line Segment (CLS)* map. It only uses line segments as elements and provides a closed and connected region. Line segment envelopes are created to approximate those spots that do not provide line information in range scans.

There are two types of localization—local and global. In local localization, the rough position estimate of a mobile robot is known. More accurate estimate is then obtained by searching locally for a good match between the current range data and the map [4,5]. If the position of the mobile robot is completely lost, global localization is performed via an extensive search in an adequately large space [1]. In most approaches, raw range data are registered to the map. In this paper, we propose to use line segments for both local and global localization and show that these techniques do not need a large search space, leading to computational efficiency.

The rest of this paper is arranged as follows. In section 2 the local localization technique based on line segment is described. Section 3 introduces the *CLS* map. Section 4 discusses the global localization. The experimental results are in Section 5, and we conclude in Section 6.

2. Local Localization using Line Segments

2.1 Using Line Segment instead of points

The application of a 2D laser rangefinder has become increasingly popular in mobile robotics. It provides denser scans and more accurate measurements as compared to other sensors like ultrasonic and infrared rangefinders. The measurement provides clear line features for an ordinary indoor environment, as shown in Fig. 1.

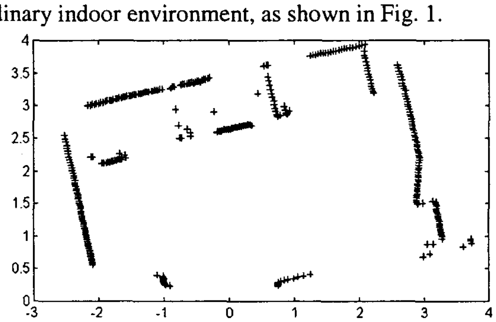

Figure 1: A sample laser scan of our laboratory.

As a geometric feature, lines provide strong and accurate information, but with far less number than that of points. It is therefore quite conceivable that line based algorithms are more efficient compared to algorithms that are point based.

To extract line segments from range points, we use recursive line splitting method, which is reliable and fast. Various steps of this method are described as follows.

1) Form a line from start and end points
2) Find the point with the biggest distance to this line
3) If its distance is small, a line segment is found
4) If not, split the point group at this point and repeat

We propose to use a special Center of Gravity (COG) line description for the line segments (Fig. 2). Ordinary COG line description consists of the center of gravity (x_c, y_c) of all points, and the slope k [9]. It can describe the uncertainty of a line segment very well.

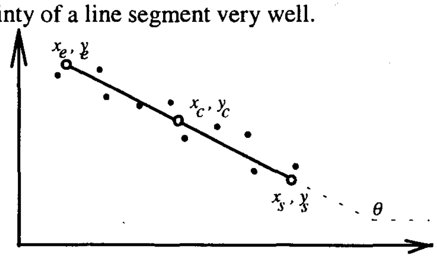

Figure 2: A COG line segment description

Since the range scans are either clockwise or counter-clockwise, the line fitted from range data would have a direction. This is important for checking line correspondence. Even if two line segments are very close to each other, if they point to opposite directions, they do not correspond. So, angle θ is used instead of slope k, for $\theta \in [0,2\pi)$. The variances σ_θ of θ and σ_c of (x_c,y_c) are calculated based on that of range points (symmetric position noises assumed). The start point (x_s, y_s) and end point (x_e, y_e) are computed. The data structure is:

```
typedef COG_Line
{    float θ, σθ;
     float xc, yc, σc;
     float xs, ys;
     float xe, ye; }
```

2.2 Line Correspondence

Two sets of line segments, one extracted from the latest laser scan, and the other from a previously built global map, can be easily matched to compute the current position of the mobile robot. This is the task of *local localization*. The odometer readings are used to provide the approximate movement information.

To find the corresponding line segment pairs, line segments in the current scan are first transformed into the global coordinates based on the approximate movement of the mobile robot. Subsequently, a matching check is performed for each current line segment against all the line segments in the global map, based on the following two criterions:

- The directions θ_L and θ_G are close
- The line center (x_c, y_c) of any one segment is close to the other line segment.

2.3 Local Localization Calculations

Local localization is performed based on the matching pairs of line segments. If line segments being matched are not all parallel, we can compute the current position. Since rangefinders in use always have a big viewing angle, most of the time this condition is satisfied.

Suppose that we have obtained n corresponding line segment pairs. The rotation θ is calculated by (2.1).

$$\theta = \sum_{i=1}^{n} (\theta_G^i - \theta_L^i) / (\sigma_{\theta G}^i + \sigma_{\theta L}^i) \qquad (2.1)$$

where G,L stand for global and local item.

We use the Weighted Least Squares (WLS) method to compute the translation (t_x,t_y). For each corresponding pair, an equation is obtained based on the fact that $(x_c,y_c)_L$ lies on the corresponding global line:

$$(\begin{bmatrix} a_G & b_G \end{bmatrix}(R_\theta \begin{bmatrix} x_{cL} \\ y_{cL} \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}) + c_G)(\sigma_{cG} + \sigma_{cL}) = 0 \qquad (2.2)$$

where (a_G, b_G, c_G) are the parameters of the line using the standard $ax+by+c=0$ description. The variances on the center of gravity are used as a weighting factor. The translation vector (t_x, t_y) is then calculated by the Least Squares method for all the corresponding pairs of lines.

This technique has been utilized in experiments, and the obtained results are very accurate. In most times, at least 3 or 4 matched pairs have been obtained. The errors on location parameters (t_x,t_y) are within 10mm, and the errors on rotation θ are within 0.5 degree.

Uncertainty measures on the estimated position parameters are necessary to study the error propagation along the robot movements when building a map. In this paper, we have used fixed uncertainty bounds on the parameters. The experimental results on the problem of map building have been shown to be satisfactory. It may be remarked that a technique based on Extended Kalman Filter (EKF) approach has been developed earlier [9], to compute the uncertainty measure on the estimated position in a shape estimation task. The proposed scheme complements our earlier approach.

3. Map Building Based on Line Segments

The indoor environment that needs to be mapped can always be treated as a closed 2D region in which the mobile robot can travel. The *occupancy grid* [3] map by its nature gives a closed description. When it comes to using line segments, since not every object in the environment would provide a line segment in the range scans, the maps based on these line segments are not closed [2,6]. Although it can still be used to do localization, the discontinuities in the map would cause uncertainties during navigation.

3.1 Introducing the Closed Line Segment Map

We introduce a new type of 2D map—*Closed Line Segment (CLS)* map. It consists of only line segments, which are connected into loops (see Fig. 3).

There are two types of line segments, the *Actual Line Segment*, which comes from real line segments extracted from range scans and *Virtual Line Segment*, which acts as a boundary between the unoccupied area and the area unexplored or explored but cannot be described using *actual line segments*. Fig. 3 shows a simple *CLS* Map. The solid line stands for the *actual line segment*, and dashed line for *virtual line segment.*

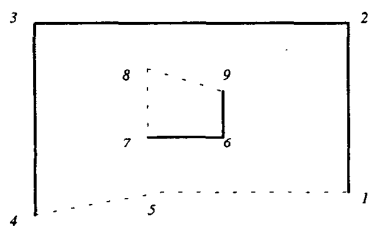

Figure 3: Example of a simple *CLS* map.

The data structure of a CLS map consists of the total number of line segments in the map, and an array of line segment items. The line segment item has the following data structure:

```
typedef CLSM_Item
{       int Index;
        int Next;
        struct COG_Line Line;
        int Line_Mode; }
```

Index indicates sequential number of the line segment in the array. *Next* is the index of the line segment connected to the end point of this line segment. *Line* uses the COG description described in Section 2, and *Line_Mode* tells this line segment is *actual* or *virtual line segment.*

The CLSM_Item array for the sample map in Fig. 3 is:

{ *(1,2,'a'), (2,3,'a'), (3,4,'a'), (4,5,'v'), (5,1,'v'), (6,7,'a'), (7,8,'v'), (8,9,'v'), (9,6,'a')* }

where the symbols *'a', 'v'* stand for *actual* and *virtual* respectively. The *Line* in the structure is omitted here. Note that this sample map contains two line segment loops, due to presence of the object in the middle.

The *virtual line segment* in the *CLS* map provides good answers to question: where to move next It is similar to the *Frontier* idea in [8]. A *virtual line segment* in the map, if it's not too short, indicates a spot the mobile robot needs to explore.. In comparison with the *frontier in* an occupancy grid map, the *virtual line segment* is a much simpler description, yet provides not only the location of the uncertain area, but also the direction (normal to the line segment). So the mobile robot can determine the location as well as rotation of the next stop in order to gain as much information as it can.

The mobile robot always moves towards a *virtual line segment* based on its sequence in the partially built map. If there is no long enough *virtual line segment*, the map is considered complete. We call this exploring strategy the *Virtual Line Segment Pursuit.*

3.2 Local Map Building

In this section, we build a CLS map based only on one set of range readings from the laser rangefinder. It is the map the mobile robot sees at its current position.

The map consists of all the *actual line segments* extracted from the range data. To deal with some objects that do not give line information, and also jumps in the range data due to occlusions, we build envelopes. An envelope consists of at most three *virtual line segments* and always lies between two *actual line segments* (Fig. 4).

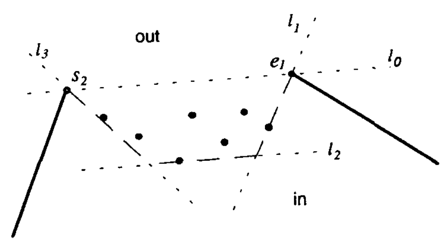

Figure 4: Building envelope for some range points

An envelope ensures that all points lie on or outside of it. These *virtual line segments* come from three lines l_1, l_2, and l_3. Line l_2 passes through the point closest to the mobile robot and is parallel to the line l_0 which connects point e_1 and s_2, the end and start points of the *actual line segments.* Line l_1 and l_3 pass e_1 and s_2 respectively, and are such that all other points are not inside (the mobile robot side) the envelope.

These three line segments can be combined into two or one if some of them are on the same line. For example, for a jump between two *actual line segments* due to occlusion, only one *virtual line segment* is needed.

Our laser rangefinder only scans 180 degrees. To make the map a closed one, we connect the end point of the last *actual line segment*, the center of the laser rangefinder and the start point of the first *actual line segment* with two *virtual line segments*.

3.3 Build Global Map Incrementally

Based on accurate position estimation using local localization developed in Section 2, we can transform the obtained local map back into the global coordinates and then update the global map. Updating the global map consists of two subtasks: region expansion and *actual line segments* fusion.

In region expansion, we are combining the local map and previously obtained global map to expand the global map. It is done by performing union operation on the polygonal regions enclosed by these two maps.

To do the polygonal region union, we need to find all points and line segments (or some portion of them) of the local map that lie outside of the global map. Fig. 5 shows an example of the polygonal union operation.

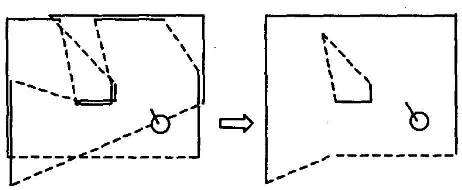

Figure 5: Union operation of two polygonal region

On the left, the local map is drawn in the global coordinates. In order to see both maps clearly, the local map is shifted up and right a little bit. The small circle with a short line segment indicates current robot position. The right side figure shows the updated global map. The enclosed region is exactly the region explored.

In *actual line segment* fusion, we fuse the corresponding *actual line segments* between local and global maps to get updated line segment parameters for the new global map. Note that the number of line segments to be fused can be more than two since the correspondence between local and global line segments is not necessarily one-to-one for fusing purpose.

The computation of the parameters of a new line segment depends on the parameters and the uncertainties in the corresponding line segments. Since the current position computed from local localization has uncertainty, the line parameters of the local map line are blurred before being fused into the global map line.

The direction θ and center of gravity $(x_c\ ,\ y_c)$ are calculated as the weighted mean of the parameters of all contributing line segments. Their uncertainties are also calculated. All end points are projected onto the newly calculated line $(\theta,\ x_c\ ,y_c)$, and furthest projections are chosen as the new end points. In fig. 6, l_n is fused from l_1 and l_2.

Figure 6: Fusing two *actual line segments*

4. Global Localization

There are circumstances when the mobile robot has no idea where it is. It might be completely lost due to disturbance. Then global localization is performed

Global localization always involves some sort of exclusive search for a possible optimal solution, or solutions if not unique. A common approach is to first divide the space into small cells, and then to compute the matching score of the current sensor readings against the map for each such cell. The high scoring cells are chosen as the possible candidates.

We propose a new search algorithm by taking advantage of the line segments. In global localization, we only deal with *actual line segments*. Note that some geometric relations among the line segments, for instance, the relative angles and positions, are well preserved after an arbitrary coordinate change (see Fig. 7).

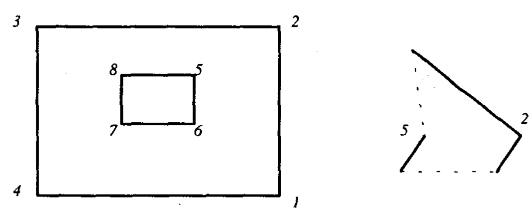

Figure 7: A global and a local map.

The purpose of our search algorithm is to make as many correspondences as possible between the local and the global line segments using the associated geometric relations. Suppose that a line segment l_L of a local map corresponds to a line segment l_G of the global map as shown in Fig. 8. The rotation θ of the current position of the robot with respect to the global coordinates is just the difference between the angles θ_G and θ_L, which are the associated angles of the line segments with respect to the global and local coordinates respectively. The possible translation lies on a line segment $(t_b\text{-}t_e)$, named as *translation line segment*, which is computed by using the rotation θ and the start and end points of the corresponding line pairs, see Fig. 8.

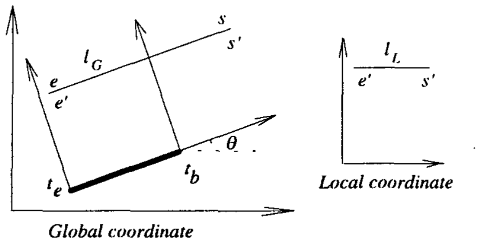

Figure 8: *Translation line segment* $(t_b\text{-}t_e)$

If two pairs of local and global line segments are actual corresponding pairs, the *translation line segments* computed from each pair should intersect (overlap if parallel). This is the underlying technique to find the corresponding line segment pairs.

The length of the line segment plays an important role. For a local line segment corresponding to a global one, the length of that global line segment cannot be shorter than that of the local one. Also, note that longer the line segments are, the more reliable is the correspondence. The global localization algorithm consists of the following steps:

1) Sort the local *actual line segments* based on their lengths, the first one is the longest one.
2) Match the longest line segment with each long enough global line segments, compute rotation θ_0 and the *translation line segment* $(t_s\text{-}t_e)_0$
3) For each of the other local line segments, try to find a corresponding global line segment that creates a *translation line segment* intersecting or overlapping with $(t_s\text{-}t_e)_0$. Compute the matching score.

The matching score is calculated as the total length of the local line segments being matched. If the matching score is close to the total length of all local *actual line segments*, the rotation θ_0 and the intersection of the *translation line segments* give a possible position of the mobile robot in the global coordinates.

To deal with uncertainty, the *translation line segments* are blurred in the computation. So, the result position is based on only one corresponding pair in rotation and intersection of some intentionally blurred *translation line segments* in location. To improve accuracy, we need to perform the local localization here based on the just computed position.

Sometimes the algorithm can have multiple solutions due to similar environment features at different locations. The mobile robot needs to move around to get further observations in order to pick the right one. An efficient way is to do pure rotation first. The above algorithm will be performed again for the new range scan. Compare the result to previous ones to narrow down the choices until the unique solution is obtained. Note that these movements can be computed accurately by local localization described in Section 2.

5. Experimental Results

All the techniques presented in this paper have been tested on the mobile robot Nomadic XR4000 in our laboratory. This mobile robot has a holonomic drive system and is equipped with a SICK LMS 200 laser rangefinder with resolution of 0.5 degree over 180 degree scanning angle.

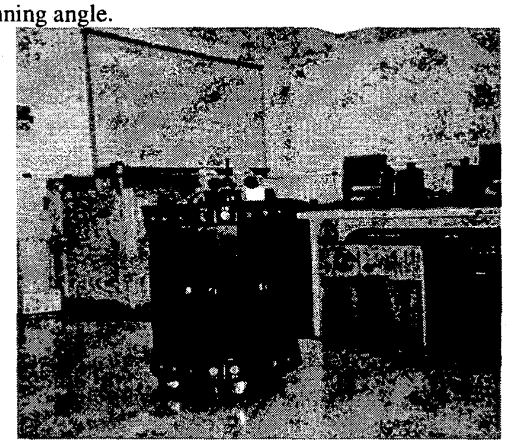

Figure 9: Our Nomad XR4000 in the laboratory

The first task is to build a 2D global map of our laboratory. The room is about *7.5m* x *6m* in size, and has some tables with flat legs along the wall, several chairs near the tables, and some boxes scattering in the room. The movement is determined based on *Virtual Line Segment Pursuit* strategy. At each location, a local map is built and the local localization is performed against the previously built global map. Then, the global map is updated. The mobile robot makes 30 movements to complete the whole map, which consists of 51 *actual line segments* and 33 *virtual line segments*. Two intermediate results and the final map are shown in Fig. 10,11.

The second task is to do global localization. The mobile robot is moved to some location in the room first. Then it makes a range scan and performs the global localization based on previously obtained global map. Fig. 11 and 12 show one result.

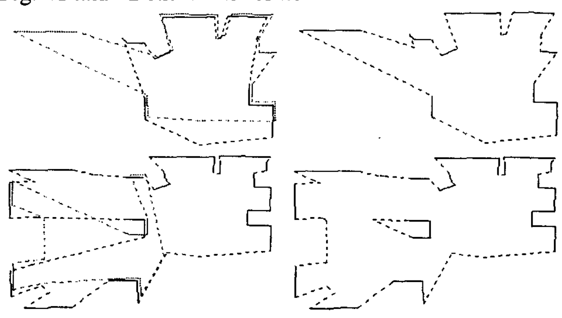

Figure 10: Current local map drawn (shifted) on previous global map (left), updated global map (right)

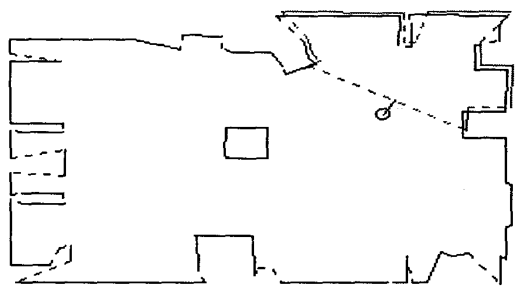

Figure 11: The global map and a global localization result

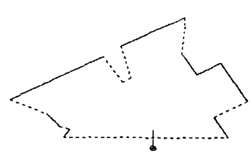

Figure 12: A local map at some position

The local map is translated and rotated based on the global localization result and drawn (shifted up and right for better look) on the global map in Fig 11. The perfect match indicates the global localization result is accurate.

All programs are running on a PC with a 300MHz Pentium II processor. The following table shows the maximal time for each task in the experiments:

Build local map	2.2ms
Update global map	170ms
Local Localization	1.6ms
Global Localization	17ms

The computations are very fast, which is the direct benefit of using range line segments. It is possible to localize the mobile robot with high accuracy in real time.

6. Conclusions

In this paper we have presented local and global localization methods and map building technique using line segments. We show via experiments that the proposed approach is accurate and is computationally fast. In order to build a closed map, *virtual line segment* has been introduced in *CLS* map resulting in a new *Virtual Line Segment Pursuit* strategy. This strategy can guide the mobile robot to explore the environment more efficiently.

The accuracy for both local and global (for unique solution) localization is within 10mm in location and 0.5 degree in orientation. The times used for local and global localization are within 4ms and 20ms. This ensures that localization is possible in real time.

Our future work will be focusing on using vision and adding more geometric features, such as arcs and corners, to enhance the performance and do tasks in 3D.

References

[1] F. Dellaert, D. Fox, W. Burgard, S. Thrun, "Monte Carlo localization for mobile robots", *IEEE Int. Conf. On Robotics and Automation*, pp 1322-1328, 1999.

[2] J. Gonzalez, A. Ollero, A. Reina, "Map building for a mobile robot equipped with a 2D laser rangefinder", *IEEE Int. Conf. On Robotics and Automation,* pp 1904-1909, 1994.

[3] Hans Moravec, Alberto Elfes, "High resolution maps from wide angle sonar", *IEEE Int. Conf. On Robotics and Automation*, pp.116-121, 1985.

[4] J. Neira, J. D. Tardos, J. Horn, G. Schmidt, "Fusing range and intensity images for mobile robot localization", *IEEE Trans. On Robotics and Automation,* Vol. 15, No. 1, Feb. 1999

[5] A. C. Schultz, W. Adams, "Continuous localization using evidence grids", *IEEE Int. Conf. On Robotics and Automation*, pp 2833-2839, 1998.

[6] J. Vandorpe, H. Van Brussel, H. Xu, "Exact dynamic map building for a mobile robot using geometrical primitives produced by a 2D range finder", *IEEE Int. Conf. On Robotics and Automation*, pp 901-908, 1996.

[7] S. Wei, Y. Yagi, M. Yachda, "Building local floor map by use of ultrasonic and omni-directional vision sensor", *IEEE Int. Conf. On Robotics and Automation*, pp 2548-2553, 1998.

[8] Brian Yamauchi, Alan Schultz, William Adams, "Mobile robot exploration and map-building with continuous localization", *IEEE Int. Conf. On Robotics and Automation*, pp 3715-3720, 1998.

[9] Li Zhang, Bijoy K. Ghosh, "A multisensor fusion approach to shape estimation using a mobile platform with uncalibrated position", *IEEE/SICE/RSJ Int. Conf. on MFI*, pp205-210, Aug. 1999