Proceedings of the 2008 IEEE
International Conference on Robotics and Biomimetics
Bangkok, Thailand, February 21 - 26, 2009

Recursive Line Extraction Algorithm from 2D Laser Scanner Applied to Navigation a Mobile Robot

Mohammad Norouzi, Mostafa Yaghobi, Mohammad Rezai Siboni, Mahdi Jadaliha
Department of Electrical, Computer and IT Engineering
Islamic Azad University, Qazvin Branch, Mechatronics Research Lab (MRL)
Daneshgah Street - Qazvin - Iran
mnorouzi@ee.iust.ac.ir, yaghobi@mrl.ir, mhdrezai@mrl.ir, jadaliha@ee.sharif.edu

***Abstract* - This paper presents a novel algorithm for line extraction from 2D laser range finder which has excellent precision to determine of line boundary, while having efficient speed and low complexity. This algorithm is more robust in measurement noise and cluttered data and fit a line to a set of uncertain points via Hough transform and recursive split formalism. In this algorithm first the data from laser is segmented to separate regions by determining start and end points throw Hough transform. After removing noisy or sparse points lines are extracted with a fast recursive split algorithm.**

Furthermore, it has good real-time capability to integrate the information of laser scanner into the navigation algorithm of the mobile robot and it is implemented on NAJI V rescue robot platform.

Index Terms* - *Line extracting, Hough transform, Segmentation, Obstacle avoidance, Navigation.

I. INTRODUCTION

Localization, map building, navigation and obstacle avoidance are important tasks of mobile robot. For navigation task a mobile robot needs a system to see its environment. So environment perception, sensor data processing and environment modeling are fundamental problems of autonomous navigation in an unknown arena. In recent years laser scanners have often been used for environment perception and automatic map building in 2D or 3D especially in outdoor environment [10]. A large number of experiments using various kinds of sensors has shown that range sensor based SLAM techniques using laser [3], sonar [9] and vision work well in real environment for both indoor [14] and outdoor application [6] and robot navigation as well [10]. There are many advantages of laser scanner compared to other sensors: it provides dense and more accurate range measurements, it has high sampling rate, high angular resolution, good range distance and resolution.

In many cases feature extraction is so useful than working with raw data. For example in mapping problem laser scan matching methods are categorized based on their association: point to point and feature to feature. The point to point matching approach [3], [5], [13], is to approximate the alignment of two consecutive scans, and then iteratively improve the alignment by defining and minimizing some distance between the scans. Moreover, it does not require the environment to be structured or contain predefined features.

In the feature to feature matching approach, instead of working directly with raw scan points, the raw scan is transforms into geometric features. These extracted features are used in the matching in the next step and in obstacle avoidance and exploration process as well. Such approaches interpret laser scans and require the presence of chosen features in the environment. Features such as line segments [4], corners or range extrema [6] are extracted from laser scans, and then matched. Features require less memory space while provide rich and accurate information. Algorithms based on parameterized geometric features are expected to be more efficient compared to point-based algorithms. Selecting a best method to extract lines from scan data is the first task for anyone who is going to build a line-based navigation system using 2D laser scanner.

Among different geometric primitives, lines segments are the simplest one. Most office environments are easily described using line segments. Line-based maps are suitable for indoor applications, or structured outdoor applications, where straight edged objects comprise many of the environmental features. They are more robust to noise than raw data, because a line is composed of many points, the noise on a point usually does not affect the position and orientation of the line substantially. The sets of segments can be input to another algorithm that extracts high level features such as doors or corners. The line segments can be used as part of or all of the local map representation at the core of a SLAM algorithm or obstacles data in navigation problem.

Several algorithms have been proposed for extracting line segments from 2D range data. Nguyen et al. [1] presents an experimental evaluation of different line extraction algorithms on 2D laser scans for indoor environment. Six popular algorithms in mobile robotics and computer vision are selected and compared. Diosi et al. [2] consider line fitting systematic errors as they mainly depend on a specific hardware and testing environment. S. T. Pfister et al. [4] suggest a line extraction algorithm using weighted line fitting for line-based map building. T. Pavlidis et al. [11] proposed a split-and-merge algorithm for line extraction which is oriented from computer vision.

There are many approaches for exploration or navigation a mobile robot. J. Minguez et al. [15], [16] presented a reactive collision avoidance for vehicles that move in very dense, cluttered, and complex scenarios. G. Oriolo et al. [17] presented a Sensor-based Random Tree (SRT) for exploration of unknown environments which represents a roadmap of the explored area with an associated safe region. That extended in [18], [19], [20]. S. Wirth and J. Pellenz [21] presented an approach that provides a robust solution for the exploration task: Based on the knowledge of the environment that the robot has already acquired, the algorithm calculates a path to the next interesting "frontier". B. Tovar et al. [7] proposed a planner that selects the next position of the robot based on maximizing a utility function.

This paper introduces a noble algorithm for real time line extraction from sets of dense range data that are generated from a 2D range finder. We modify this line segmentation algorithm such that its quality and robustness increase highly. This paper is organized as follows; in section II recursive line extraction algorithm is described, section III describes navigation algorithm which use this line extraction algorithm, section IV shows details of experiments result and finally section V discuses on conclusion and future works.

II. Recursive Line Extraction Algorithm

The laser range finder delivers raw range data from environment of the robot. From raw data we wish to extract natural features to be used in obstacle avoidance and robot navigation. This section describes two steps: Hough lines and line segments.

A. Hough Transform

In fairly clean environments the Hough transform can be applied to extract walls with good result. Lines in the environment of the robot are represented in Hough space by orthogonal distance d from the robot to the line and direction γ between d and the direction of the robot. In fact the measurement point i in raw range data is represented by range r_i and beam direction φ_i in polar coordinates. The g-weighted Hough transform is then given by

$$C(d,\gamma) = \sum_i w(r_i \cos(\varphi_i - \gamma) - d) g(r_i, \varphi_i, \gamma) \qquad (1)$$

Where w in this section is a window function and g is the weight function. The arguments in (1) are illustrated in Fig. 1.

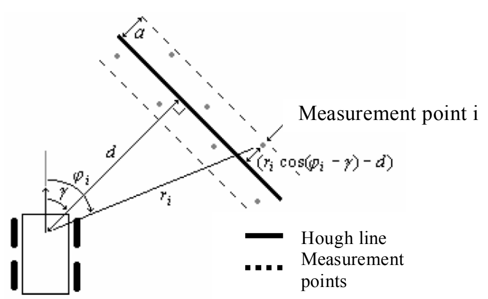


Fig.1 Measuring robot and a Hough line

The window function is a rectangular window picking out the measurement points belonging to the dotted stripe with width $2a$ in Fig. 1 and x is the distance from the line.

$$w(x) = \begin{cases} 1 & |x| \le a \\ 0 & |x| > a \end{cases} \qquad (2)$$

This work was realised in the frame of NAJI V project (Autonomous Rescue Robot). It is constructed at the base of a four wheel differential moving system. It is equipped with an IBM T61 Laptop and a URG-04LX scanning range finder with 240^o measuring area and 0.36^o degree angular resolution (Fig. 2). So it delivers 682 point per each scan and the raw laser data is an array of measurement points 3<P[i]<300, 0<i<682. So the values P[i]>300 and P[i]<3 assumed as noisy data which wouldn't be taken in any data analysis. The laser scan is searched for lines by calculating (1) for all d from 3 to 300cm and γ from 0 to 240^o separated by predefined steps Δd=3cm and $\Delta\gamma=2.47^o$ giving a matrix filled with the C-value for each Hough line. But it may be differ in other platforms based on laser scanner parameters.

$$C_{matrix} = \begin{bmatrix} C(3,0) & C(3,\Delta\gamma) & \dots & C(3,240^o - \Delta\gamma) \\ C(3+\Delta d,0) & C(3+\Delta d,\Delta\gamma) & \dots & C(3+\Delta d,240^o - \Delta\gamma) \\ \vdots & \vdots & \vdots & \vdots \\ C(300,0) & C(300,\Delta\gamma) & \dots & C(300,240^o - \Delta\gamma) \end{bmatrix} \qquad (3)$$

Calculating (1) without weight function (g=1) correspond to counting the number of measurement points contained in the strip over the Hough line (d,γ). For noise canceling the lines with values less than a criterion will be set to zero. This process assigns the points to spider net like Hough lines. Fig 3 illustrates one sample scan and extracted Hough lines by criterion = 1. A weight function with $g(ri, \varphi i, \gamma) = ri$ is although recommended since measurements become less and less dense as range increases. Otherwise the transform will tend to find lines close to the robot easier than lines further away.

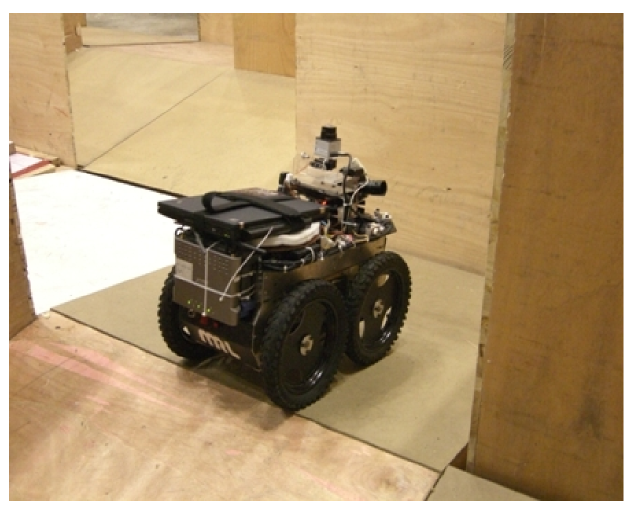
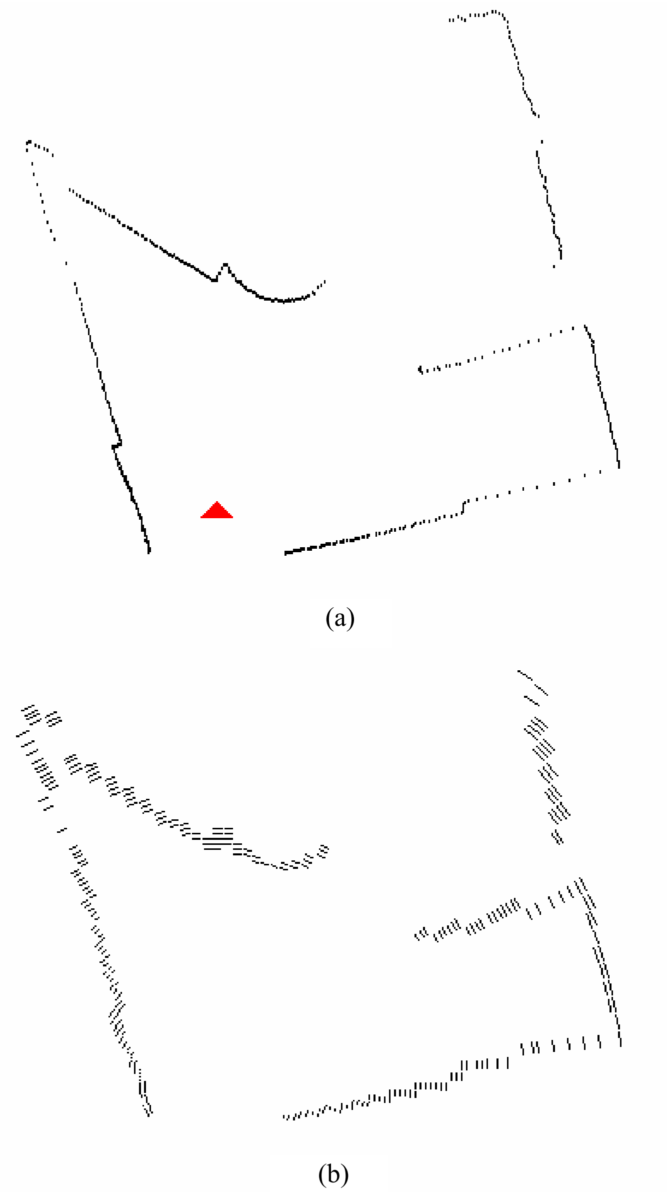


Fig.2 NAJI V Autonomous rescue robot



(a)

(b)

Fig.3 (a) The raw laser scan, (red triangle shows robot position)
(b) The Hough line with values bigger than criterion

B. *Grouping and Segmentation*

After Hough transform and noise canceling the matrix C is generated which each component has value between 0-7. The values of Hough lines in Fig.3 (b) are between 2-7. Then adjacent lines in each radius are grouped together which are illustrated by blue lines in Fig. 4 (a). Now by starting with first group all neighbor groups are merged together and make new group and the process continues until it can't find any neighbor in around. This big isolated group is labeled as data segment which are indicated by red lines between start and end points in Fig. 4 (b). The radius grouping and merging the adjacent groups together in all directions results two noise canceling as well, first by removing Hough lines with value under an special criterion, second the measurement points which aren't belong to any group will be removed too that means points with low density aren't accepted as a line. Therefore all the cluttered and noisy points which haven't enough correlation will be removed and the raw measurement point array P[i] is modified to Pm[i] by setting the removed points to 0. So correlated measurement points which are belong to an object or an visible part of it are segmented and identified by start and end point in polar coordinates system which is illustrates by red lines between start and end point in each segment in Fig. 4 (b). Then segments array SA[t][j] generated. Where t is number of segments which is variable in other shots and SA[t][0], SA[t][1] correspond to number of start and end point of segment t in Pm[i] array. So the brief segmentation process has two steps.

1. Neighbor Hough lines in each radius are grouped together with an adjustable threshold Fig 4 (a) illustrates the radius grouping
2. Adjacent Groups are merged together to perform data segments. Fig 4 (b) illustrates adjacent merged radius groups.

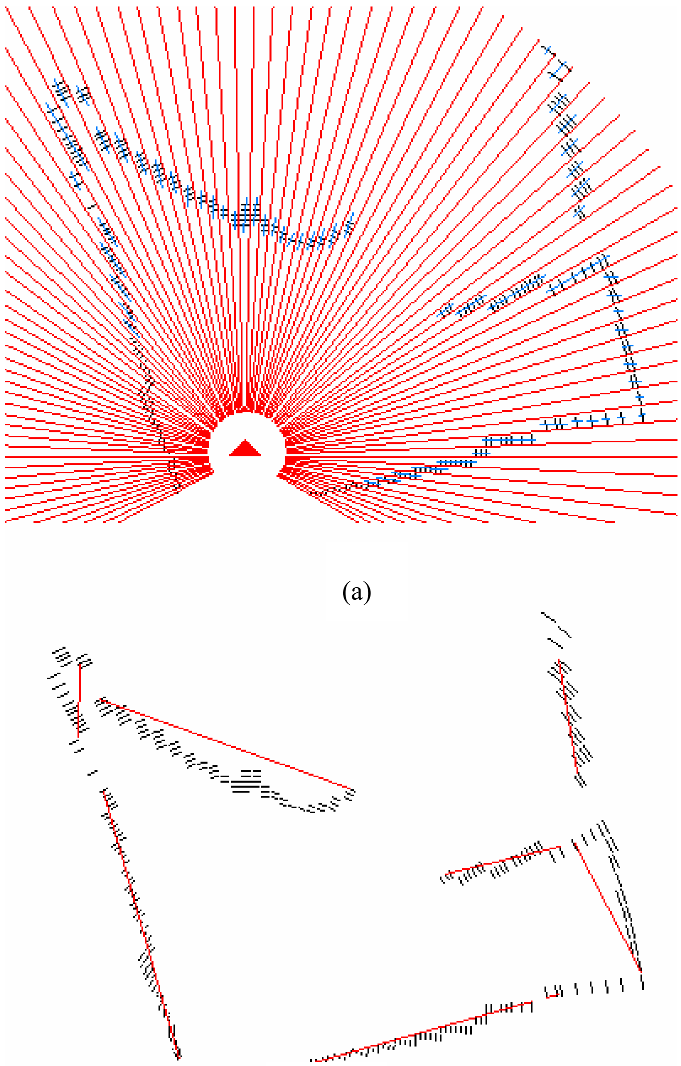
C. *Recursive Line Extraction*

After computing segments array SA[t][j] each red line between start and end point of segment will be split to sub lines by a recursive algorithm.

Consider SA[t][0] and SA[t][1] are transferred to recursive function *Linseg(start,end)* as *start* and *end* respectively. The body of *Linseg(start,end)* function is:

```
Linseg(start,end)
{
    1. Find the measurement point with maximum deviation
       from the line and assign it as middle
    2. If middle fulfill the criterion
       return
    else
       {
           LineNumbers++;
           Linseg(start,middle)
           Linsegmiddlet,end)
       }
}
```

Computing *Linseg(start,end)* function for all data segments results an array of points which are starts and ends of lines in whole Pm[i] array. Note that some end points in fact are the next line's start point. Fig 5 illustrates the result for Fig 4. There are some parameters to adjust the algorithm, Δd and $\Delta\gamma$ in Hough transform, the criteria for removing Hough lines with low value and the grouping criteria for estimating the segments. So one can do trade off between speed and accuracy of fitted lines. But the algorithm is so robust to sporadic dropouts and faulty measurements and efficient as well.



(a)

(b)

Fig.4 (a) Radius grouping (b) Adjacent merged radius groups

The brief algorithm is:

```
1. Initial: A set of N points
2. Initialize the accumulator array (model space)
3. Set radius grouping

4. Merged radius groups in any direct and
   determine data segments with theirs start and end point
5. Extract lines in all data segment with recursive function:
      Linseg(start,end)
   {
   1.     Find the measurement point with maximum deviation from the line
          and assign it as middle
   2.     If middle fulfill the criterion
   return
   else
        {
         LineNumbers++;
         Linseg(start,middle)
        Linsegmiddlet,end)
        }
   }
```

The Hough transform algorithm which is implemented in [1] in order to compare with other algorithms is:

```
1. Initial: A set of N points
2. Initialize the accumulator array (model space)
3. Construct values for the array
4. Choose the element with max. votes V_max
5. If V_max is less than a threshold, terminate
6. Otherwise, determine the inliers
7. Fit a line through the inliers and store the line
8. Remove the inliers from the set, goto 2
```

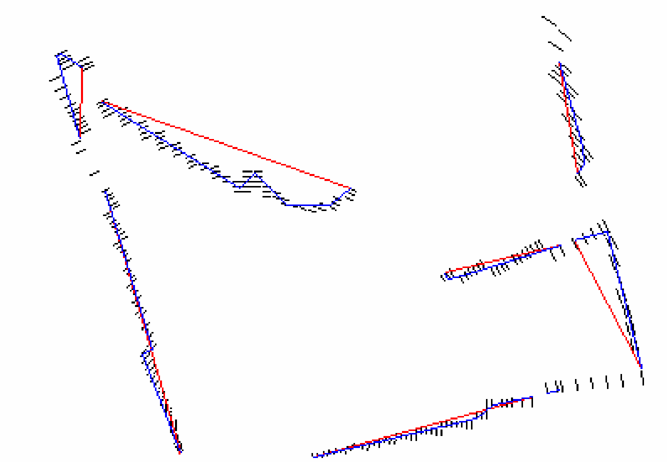
In compare with the algorithm in [1] our approach calculate time consuming step 2 one time and doesn't inter it in its recursive loop so the algorithm is faster. Also it removes all the cluttered and noisy points which haven't enough correlation and spare data as well.

III. Robot Navigation

As it is mentioned it is easy to describe most office environment using line segments. Line-based maps are most suited for indoor applications, or structured outdoor applications, where straight edged objects comprise many of the environmental features. So the line segments can be used as part of or all of the local map representation at the core of a SLAM algorithm and it's so useful for obstacle avoidance and path planning too. In order to test the algorithm it is used in wall tracking in NAJI V robot.

The autonomous mobile rescue robot (NAJI V) is a mobile system capable of interpreting, planning, and searching simulated arena for probably victims without any external support. The goal of the urban search and rescue (USAR) robot competitions is to increase awareness of the challenges involved in search and rescue applications, provide objective evaluation of robotic implementations in representative environments, and promote collaboration between researchers. It requires robots to demonstrate their capabilities in mobility, sensory perception, planning, mapping, and practical operator interfaces, while searching for simulated victims in unstructured environments. During exploration phase the robot's perception system has to incrementally build up internal models of the environment which most accurately represent topological and geometrical properties of the environment. The major difficulty in robot perception is to find out what the sensor signals tell about the real world – in the other words: the problem is to interpret the sensor signals in such a way that each world model represents an optimal estimate of the real world's state with respect to the models specific purpose. Fig 6 illustrates the match arena in USA 2007 RoboCup competition.

Due to maze structure of the arena, line segment is the best geometric primitive for navigate and obstacle avoidance algorithm because robot can extracts information like distance and direction of wall from a line segment by some mathematic operation. As it's obvious from Fig 5 the number of points is reduced to around 30 point rather than 682 in raw data. For the navigation and obstacle avoidance we divide the robot around to 3 sectors: left side, front and right side. Left and right sides angles are 70^{o} and front 100^{o}. In each sector the biggest and nearest object is selected and labeled as select_obj[i] i=0,1,2 for left, front and right respectively. Fig 7 illustrates robot GUI and the selected blue objects in each sector for a sample environment.



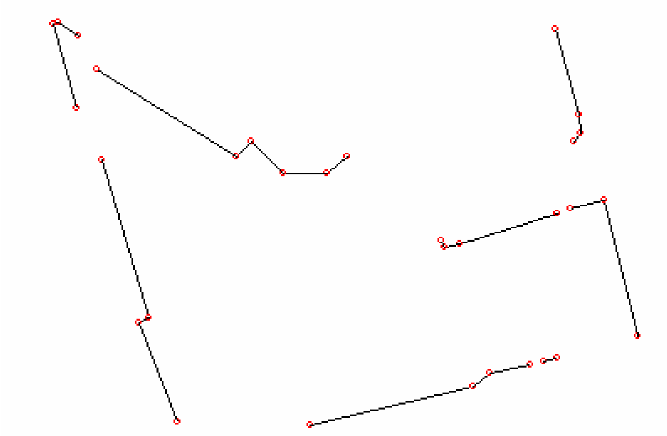


Fig.5 Extracted lines from Fig. 4, red circles shows start and end points

Fig. 6 Rescue robot arena in USA 2007 competition

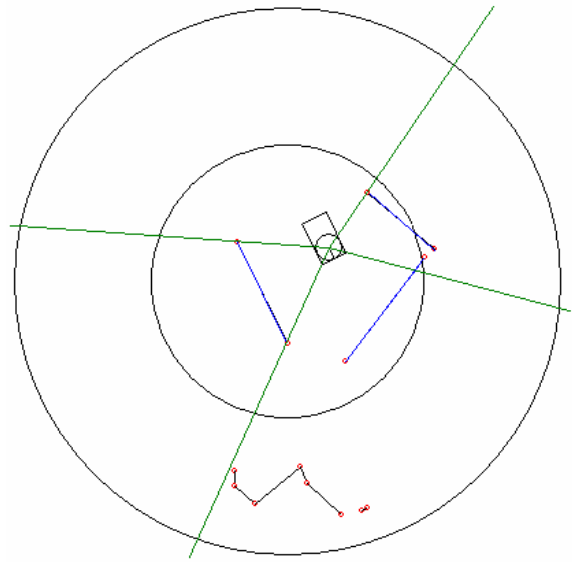


Fig. 7 Three sector around robot and selected objects

For each selected object the distance from robot, wideness and line's angle in robot's coordinate system are extracted then the model of environment is performed. The brief flow chart of exploration algorithm is shown in chart 1.

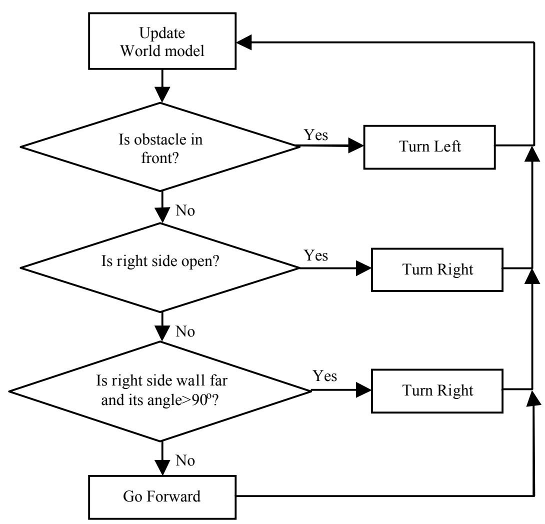


Chart. 1 Brief Exploration Algorithm

IV. EXPERIMENTAL RESULT

The algorithm is implemented on autonomous rescue robot NAJI-V in real simulated rescue arena shown in Fig. 6 and the result is illustrated in Fig. 8. The circular and rectangular grids are 1 meter. The black lines show walls around the arena, green circle shows robot's initial point and red ones show probably victims placed in map. Green line shows the robot path. In the arena construction the floor has 10^{o} roll and pitch ramps so the walls are duplicated in somewhere especially in left side but the path is acceptable in maze arena. The extracted lines are suitable for object base mapping approaches as well.

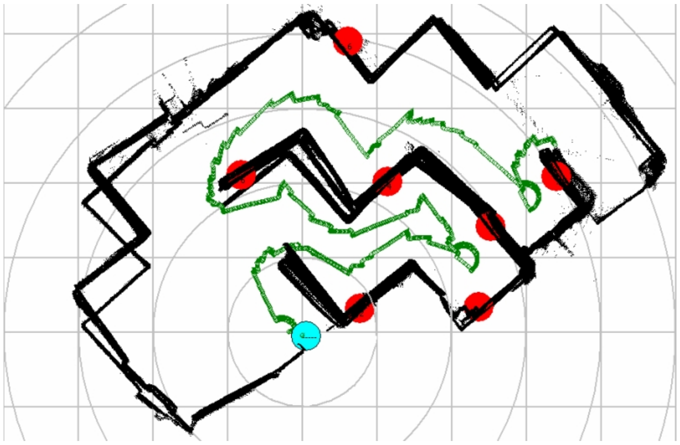


Fig. 8 The test result in real arena

V. Conclusion

For navigation a mobile robot, environment perception and its modeling is necessary. For NAJI V perception of environment, we have chosen URG-04LX. It is fast, robust, low in cost and weight enough for our application. For the environment modeling among many geometric primitives, lines segments are the simplest. It is easy to describe most office environment using line segments where straight edged objects comprise many of the environmental features. We have developed a fast algorithm for processing the laser data gives the good result for environment representation and building navigation map in real time. Our current work is building the real time 2D navigation map, the fusion of laser data with vision and sonar data for developing an autonomous search and rescue robot. We have implemented a mapping system by iterative closet point (ICP) approach and it is planned to develop a line based SLAM in the future.

Acknowledgment

This research was supported by Mechatronics Research Laboratory (MRL) of "Islamic Azad University Qazvin Branch". Authors are gratefully acknowledged for technical support of MRL Rescue Robot team's members.

References

[1] V. Nguyen, A. Martinelli, N. Tomatis, R. Siegwart. A Comparison of Line Extraction Algorithms using 2D Laser Rangefinder for Indoor Mobile Robotics. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2005.

[2] A. Diosi and L. Kleeman. Uncertainty of Line Segments Extracted from Static SICK PLS Laser Scans. In Proceedings of the Australasian Conference on Robotics and Automation 2003.

[3] F. Lu and E. Milios. Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 935–938, 1994.

[4] S. T. Pfister, S. I. Roumeliotis, and J. W. Burdick. Weighted Line Fitting Algorithms for Mobile Robot Map Building and Efficient Data Representation. In Proceedings of the IEEE/ICRA International Conference on Robotics and Automation, 2003.

[5] A.t Diosi and L. Kleeman. Laser Scan Matching in Polar Coordinates with Application to SLAM. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2005.

[6] K. Lingemann, H. Surmann, A. Nüchter, and J. Hertzberg. Indoor and outdoor localization for fast mobile robots. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2004.

[7] B. Tovar, R. Murrieta-Cid and C. Esteves. Robot motion planning for map building. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2002.

[8] A. Diosi, G. Taylor and L. Kleeman. Interactive SLAM using laser and advanced sonar. In Proceedings of the IEEE/ICRA International Conference on Robotics and Automation, 2005.

[9] A. Diosi and L. Kleeman. Advanced sonar and laser range finder fusion for simultaneous localization and mapping. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2004.

[10]A. Siadat, A. Kaske, S. Klausmann, M. Dufaut, and R. Husson. An Optimized Segmentation Method for a 2D Laser-Scanner Applied to Mobile Robot Navigation. In Proceedings of the 3rd IFAC Symposium on Intelligent Components and Instruments for Control Applications, 1997.

[11]T. Pavlidis and S. L. Horowitz. Segmentation of Plane Curves. IEEE Transactions on Computers, C-23(8):860–870, 1974.

[12]P. J. Besl and N. D. McKay. A method for registration of 3d shapesIEEE Transactions on Pattern Analysis and Machine Intelligence,14(2):239–256, 1992.

[13]Minguez, L. Montesano, F. Lamiraux: Metric-Based Iterative Closest Point Scan Matching for Sensor Displacement Estimation. IEEE Transactions on Robotics. Volume: 22, Issue: 5. pp: 1047- 1054.

[14]I. J. Cox. Blanche: An Experiment in Guidance and Navigation of an Autonomous Robot Vehicle. *IEEE Transations on Robotics and Automation*, 7(2):193–204, 1991.

[15]Javier Minguez and Luis Montano: Nearness Diagram (ND) Navigation: Collision Avoidance in Troublesome Scenarios. IEEE *Transations on Robotics and Automation*, vol. 20, no. 1, Feb. 2004.

[16]J. Minguez, J. Osuna, L. Montano. A "Divide and Conquer" Strategy based on Situations to achieve Reactive Collision Avoidance in Troublesome Scenarios. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2004.

[17]G. Oriolo, M. Vendittelli, L. Freda and G. Troso: The SRT Method: Randomized strategies for exploration, IEEE International Conference on Robotics and Automation, 2004.

[18]Antonio Franchi, Luigi Freda, Giuseppe Oriolo, Marilena Vendittelli: A Randomized Strategy for Cooperative Robot Exploration. International Conference on Robotics and Automation. April 2007.

[19]Antonio Franchi, Luigi Freda, Giuseppe Oriolo, Marilena Vendittelli: A Decentralized Strategy for Cooperative Robot Exploration, Proceedings of the first International Conference on Robot Communication and Coordination, 2007.

[20]Luigi Freda, Giuseppe Oriolo: Frontier-Based Probabilistic Strategies for Sensor-Based Exploration. Proceedings of the IEEE International Conference on Robotics and Automation April 2005.

[21]Stephan Wirth and Johannes Pellenz: Exploration Transform: A stable exploring algorithm for robots in rescue environments. IEEE International Workshop on Safety, Security and Rescue Robotics, SSRR 2007.