Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Use of Range Sensor Information for Improving Positioning Accuracy

Seong Ho Cho*, Sooyong Lee, Wonpil Yu*****

**Dept. of Mechanical Engineering, Hong Ik University, Seoul, #121-791 KOREA (e-mail: yoeast@mail.hongik.ac.kr).*
***Dept. of Mechanical and System Design Engineering, Hongik University, Seoul, #121-791 KOREA (Tel: 82-2-320-1609; e-mail: sooyong@hongik.ac.kr)*
****Electronics and Telecommunications Research Institute ,Daejon, #350-700 KOREA (e-mail:ywp@etri.re.kr)*

Abstract: Range sensors are commonly used for mobile robots in order to avoid collision with obstacles and build an environmental map. Unless the robot uses an external localization device, the dead-reckoning alone incurs the accumulated position errors. This paper presents an algorithm for the position error correction. Edges and nodes are identified from the range sensor information; using the identified feature points the dead-reckoning error becomes smaller. More accurate map is built while the positional error is being reduced. The proposed algorithm is implemented on a map building device and shows the effectiveness of the proposed algorithm.

1. INTRODUCTION

Mobile robots are commonly used for various tasks such as cleaning, surveillance, guide, etc. Most of the robots are equipped with range sensors in order to avoid collision with obstacles. They are also used for map building in case the environment is not known a priori, or many moving obstacles are present.

Commonly, dead-reckoning (open-loop estimation) is used for intermediate estimation of position during path execution. Dead-reckoning is often used when wheel encoders are available for drive wheel position measurement. However, due to errors in kinematic model parameters, wheel slip, or an uneven surface, poor position estimates may occur. Lots of work have been done in mobile robot localization using range sensors, vision cameras, natural/artificial landmarks, indoor GPS, etc in order to reduce the errors from dead reckoning. SLAM (Simultaneous Localization And Mapping) is one of the hot research issues, however, the algorithmic complexity, numerical computation is quite high.

Range sensors are widely used for mobile robots to detect obstacles and/or build the environment map. Commonly, robots have either the ultrasonic range sensor or IR sensor in order to detect nearby obstacles. Laser scanner or IR scanner is used for map building. Each sensor has its own characteristics such as accuracy, maximum range, robustness and price. The laser scanner gives very accurate scanning. IR scanner is far less expensive than the laser scanner, however, its accuracy is highly influenced by the lighting condition. This paper presents the use of range sensor information for improving the dead-reckoning accuracy.

2. RANGE SENSOR INFORMATION PROCESSING

Most of the laser scanners provide 180 [degree] scanning of environment with angular resolution of 0.5 [degree]. Processing the scanner information is required because the raw sensor information is simply the radial range data. One of the features commonly used is the edge. Due to the uncertainties of the sensor information, the sensor ray may not form a straight line even though all of them may hit the same edge. There have been various ways for extracting the edge from the scanner output.

Scanned range data using a scanner or sonar ring gives data set of $S \in [s(n), \theta(n)]$, where $s(n)$ is range and $\theta(n)$ is the corresponding angle. We can construct $X \in [x(n), y(n)]$ in Cartesian coordinates from S. It is commonly observed that the constructed data doesn't form an edge even though the sensor rays hit a single edge due to sensor error. It is very important to extract corners or edges from the sensor data even with errors if there exists at least one. This section describes how to process range sensor data to find edges. If there are two connected edges, they are regarded as a corner.

Finding edge is regarded as finding a line which best fits $X(i:j)$ samples. Simple linear regression is used to find the line equation of the edge in Cartesian coordinates. The results from the linear regression are a_1 and a_0 in $y = a_1 x + a_0$. In case the residual mean square value, S_r is too large, the samples used to form an edge are not valid. Before applying the linear regression to the sampled data set, we first group them into subgroups by calculating the metric distance between two nearby samples. The large distance is usually due to visibility. We start by using three consecutive samples to find out a line parameter a_1 from linear regression and also S_r. Then we include the next sample from the same subgroup and check how much S_r is changed. If the change is small enough, the newly added sample is included in the subgroup, that is, same edge. If not, the newly added sample is not from the same edge. However, it is necessary to add another next sample and remove the previously added one and check S_r because there may exist one false sensor reading.

Once the subdivision based on the linear regression is finished, merging the successive subgroups is necessary, thus it filters out noisy data from an edge. This is done by comparing the slopes of two nearby subgroups (now, they are two edges and a_1 values of two nearby edges are compared). If the difference is small enough, then two subgroups are merged into one. The algorithm is described in Table 1.

Table 1. Filtering Algorithm

Input: $S \in [s(n), \theta(n)]$
Output: Groups of Samples which form edges, Slope of edges

1) Construct $X \in [x(n), y(n)]$ from S
2) Initial Grouping:
3) Generate initial Group, $G(m)$ with
   $|X(i) - X(i+1)| > \varepsilon_1$ where $X(i) \in G(j)$ and
   $X(i+1) \in G(j+1)$
4) Subdivision:
5) Subdivide each group by linear regression
6) While $|S_r(k:l,l+1) - S_r(k:l,l)| < \varepsilon_2$
7) Add $X(i+1)$ to current subgroup
8) If $|S_r(k:l,l+2) - S_r(k:l,l)| < \varepsilon_2$ then neglect
   $X(i+1)$ and proceed with current subgroup
9) Merge:
10) If $|a_1(m) - a_1(m+1)| < \varepsilon_3$ then combine $G(m)$
    and $G(m+1)$
11) Repeat 10)

* S_r : Residual Mean Square
* $a_1()$: Slope of the line from linear regression
* $\varepsilon_1, \varepsilon_2, \varepsilon_3$: Threshold values

One of the experimental results is shown in Fig.1. The original range sensor information is converted into points in Cartesian coordinates and then the initial grouping is done in Fig.1-(a). Note that the scanner is located at (0,0). Be repeating the steps in Table 1, the edges are constructed as in Fig.1-(b). We can identify the nodes by connecting the edges as shown in Fig.1-(c) assuming that the small discontinuity occurred due to the angular resolution of the scanner.

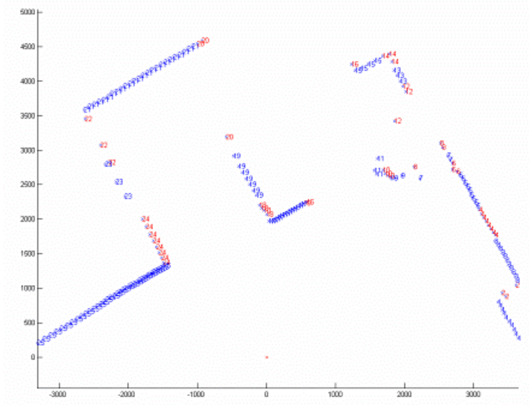

Fig. 1-(a). Lanser Scanner Reading

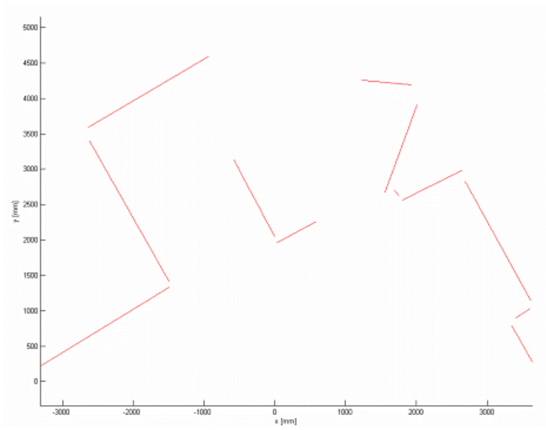

Fig. 1-(b). Constructed Edges

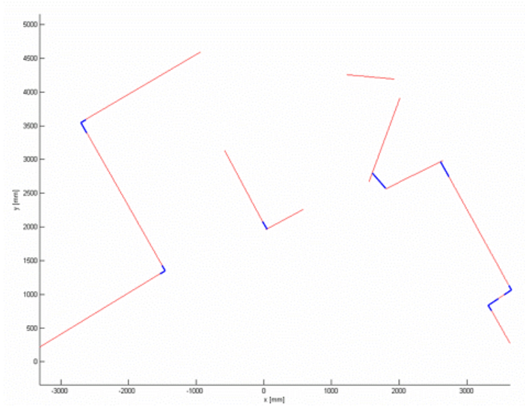

Fig. 1-(c). Identified Nodes

3. SCAN MATCHING

Dead-reckoning is a simple way to estimate the configuration (position and orientation) based on the encoder information. However, the accumulated error becomes large due to several reasons such as wheel slip, kinematic parameter variation, and moving on uneven surface. Fig. 2 shows the map built from several successive measurements using the laser scanner.

The robot made a straight forward movement, then made 180 [degree turn and made another straight forward movement. (light blue line). Measurements are made at several locations (circle on the light blue line). The alternating blue and red edges are constructed from the sequential measurements. The incorrect map in Fig. 2 is caused by the error of dead-reckoning. The edges are to be overlapped if those are constructed from the sensor rays hit on the same edge.

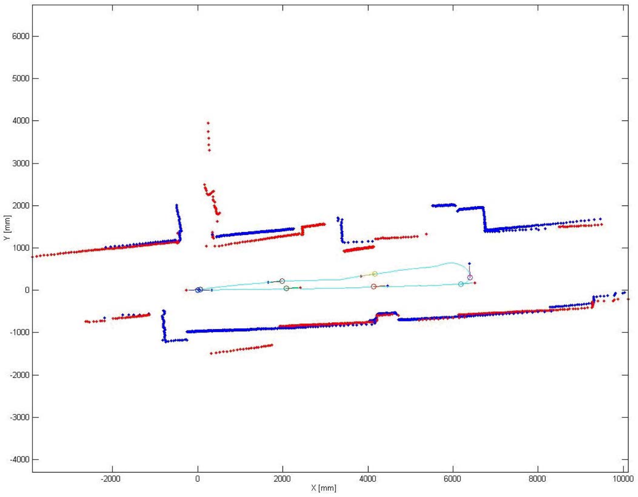

Fig. 2. Map Building with Dead-reckoning: Hallway

We introduce a way of correcting this mismatch of the edges from the sequential measurements. Fig. 3 shows another map constructed from two sequential measurements. The first scan is the black one and the second scan is the blue one. The robot's position and orientation is represented by arrows of its color. Due to wheel slip, the edges do not match because the locations of the edges are made with respect to the robot configuration estimated from dead-reckoning. In other words, the dead-reckoning error caused the incorrect map. The same node presented in two successive scanning is selected as feature points. If the feature points are successfully identified, then the scan matching is simply done by moving the second feature point appropriately so that the second feature point coincides with the first one. Fig. 4 shows the magnified view of fig. 3 around the feature points.

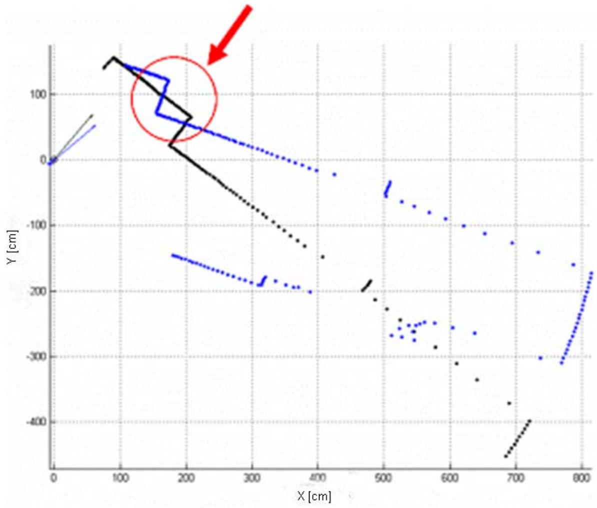

Fig. 3. Scan Mismatching due to Slip

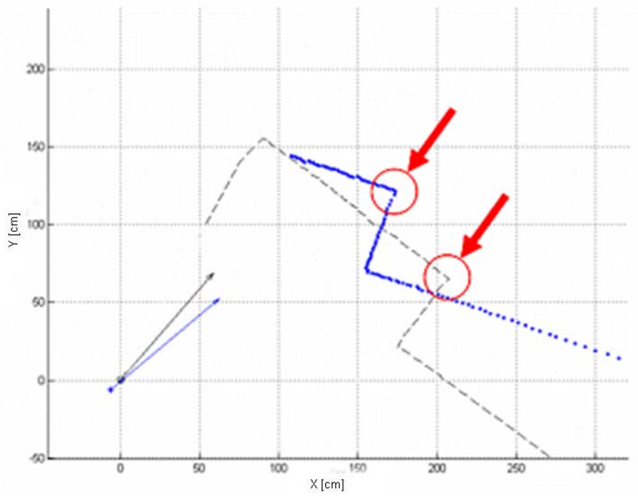

Fig. 4. Feature Points

Fig.5-(a) shows matching the sensor readings. Twelve equally spaced points are selected on the edges. Without any positioning error, the feature point and the constructed edges are to coincide. Assuming the positioning error is bounded, the matching process is finding $\Delta x, \Delta y, \Delta\theta$ which minimizes the sum of the metric distance between the corresponding points as in (1) thereby giving the matched scan as in Fig.5-(b).

$$\sum_{i=1}^{13} \sqrt{(x_{2i} - x_{1i})^2 + (y_{2i} - y_{1i})^2} \qquad (1)$$

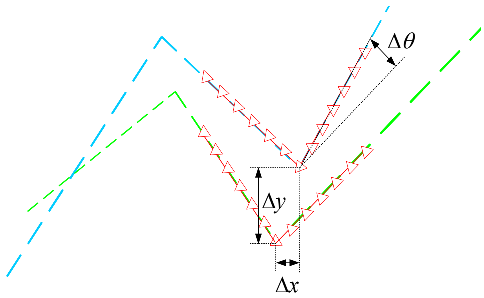

Fig. 5-(a). Mismatched Scan ($\Delta x, \Delta y, \Delta\theta$)

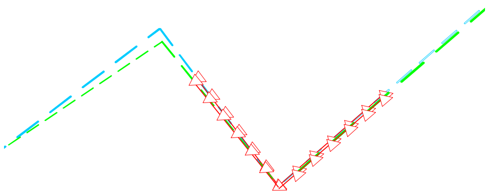

Fig. 5-(b). Matched Scan

Once the correction terms ($\Delta x, \Delta y, \Delta\theta$) are found, the configuration of the robot at the second measurement is corrected with this amount. The feature points become a point with the corrected robot configuration as shown in Fig.6.

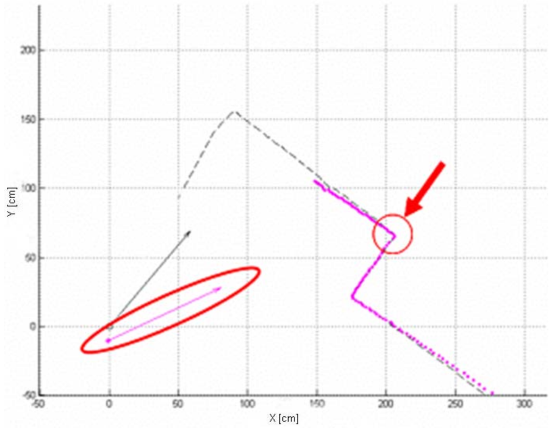

Fig. 6. Feature Points Matching and Corrected Configuration Vector

The corrected map is shown in Fig.7.

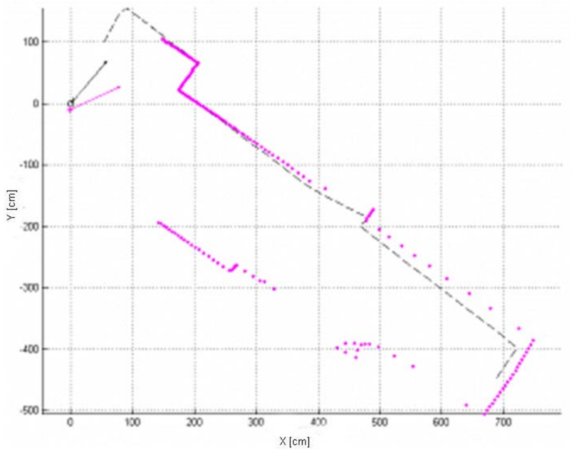

Fig. 7. Corrected Map

Another example (Figs. 8-9) shows the effectiveness of the proposed method.

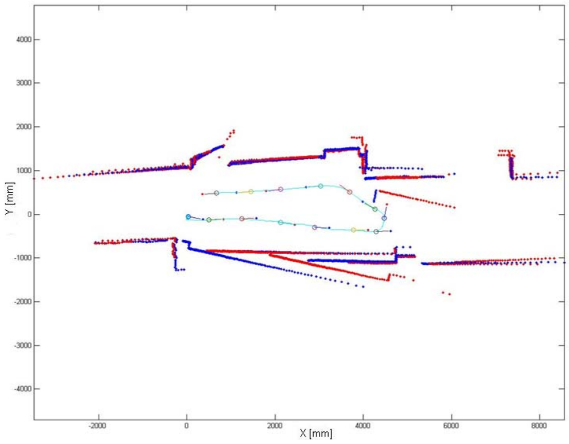

Fig. 8. Incorrect Map due to Deadreckoning Error: Hallway

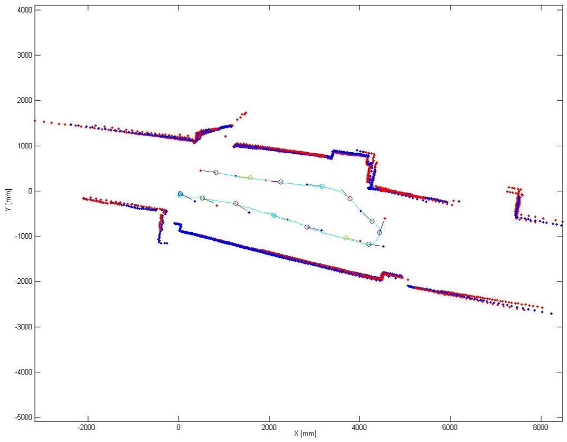

Fig. 9. Corrected Map: Hallway

In order to avoid running the scan matching procedure at every measurement, the estimation of the uncertainty is very useful. It is well known that the size of the uncertainty in robot location grows larger as the robot travels. The error should be reset before it becomes too large. For a differential drive mobile robot, the robot configuration can be estimated starting from a known configuration by integrating the movement.

The configuration vector, p is defined as

$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \qquad (2)$$

If right wheel and left wheel travels $\vartriangle s_r$ and $\vartriangle s_l$ respectively, then the new configuration is represented as

$$p' = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = f(x, y, \theta, \vartriangle s_r, \vartriangle s_l) = p + \begin{bmatrix} \vartriangle s\cos\left(\theta + \vartriangle\theta/2\right) \\ \vartriangle s\sin\left(\theta + \vartriangle\theta/2\right) \\ \vartriangle\theta \end{bmatrix} \qquad (3)$$

where $\vartriangle s = \dfrac{\vartriangle s_r + \vartriangle s_l}{2}$, $\vartriangle\theta = \dfrac{\vartriangle s_r - \vartriangle s_l}{b}$ and b is the distance between the driving wheels.

Dead-reckoning results can give only a very rough estimate of the actual position. We assume that the errors of the individually driven wheels are independent and the variances of the errors of the left and right wheels are proportional to the absolute value of the travelled distances as the following equation,

$$\sum\nolimits_{\vartriangle} = covar(\vartriangle s_r, \vartriangle s_l) = \begin{bmatrix} k_r\left|\vartriangle s_r\right| & 0 \\ 0 & k_l\left|\vartriangle s_l\right| \end{bmatrix} \qquad (4)$$

then, the covariance matrix is represented as

$$\sum\nolimits_{p'} = \nabla_p f \cdot \sum\nolimits_p \cdot \nabla_p f^T + \nabla_{\vartriangle} f \cdot \sum\nolimits_{\vartriangle} \cdot \nabla_{\vartriangle} f^T \qquad (5)$$

where $\nabla_p f = \begin{bmatrix} 1 & 0 & \Delta s\sin\left(\theta + \vartriangle\theta/2\right) \\ 0 & 1 & -\Delta s\cos\left(\theta + \vartriangle\theta/2\right) \\ 0 & 0 & 1 \end{bmatrix}$

and $\nabla_\Delta f = \begin{bmatrix} f_1 & f_2 \\ f_3 & f_4 \\ f_5 & f_6 \end{bmatrix}$

$$f_1 = \frac{1}{2}\cos\left(\theta + \vartriangle\theta/2\right) - \frac{\Delta s}{2b}\sin\left(\theta + \vartriangle\theta/2\right)$$

$$f_2 = \frac{1}{2}\cos\left(\theta + \vartriangle\theta/2\right) + \frac{\Delta s}{2b}\sin\left(\theta + \vartriangle\theta/2\right)$$

$$f_3 = \frac{1}{2}\sin\left(\theta + \vartriangle\theta/2\right) + \frac{\Delta s}{2b}\cos\left(\theta + \vartriangle\theta/2\right)$$

$$f_4 = \frac{1}{2}\sin\left(\theta + \vartriangle\theta/2\right) - \frac{\Delta s}{2b}\cos\left(\theta + \vartriangle\theta/2\right)$$

$$f_5 = \frac{1}{b}$$

$$f_6 = -\frac{1}{b}$$

From the analysis above, the robot's uncertainty ellipsoid is estimated and plotted together with the robot's estimated path in Fig. 10.

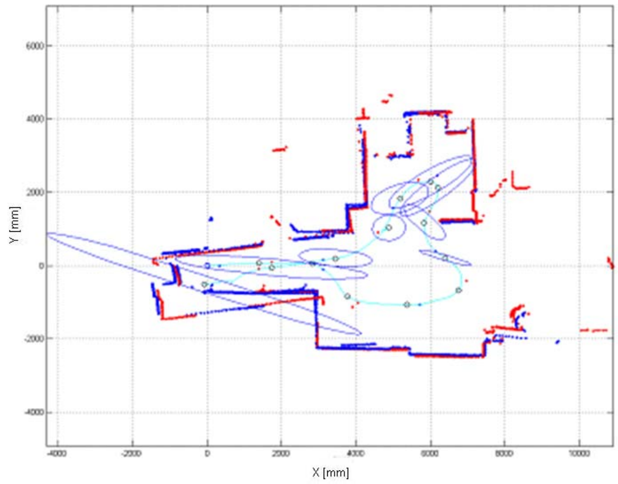

Fig. 10. Incorrect Map with Uncertainty Ellipse

Therefore, the size of the uncertainty ellipsoid indicates when the scan matching is necessary. The corrected map for the environment of the above figure is shown in Fig. 11.

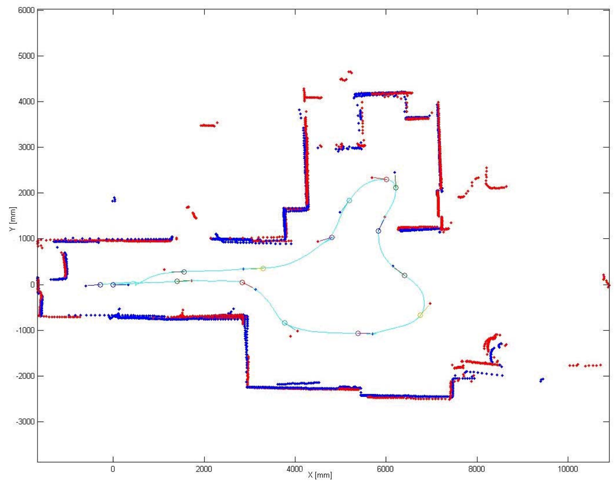

Fig. 11. Corrected Map: Room

4. CONCLUSIONS

Dead-reckoning alone gives the accumulated positioning errors. Localization is required in order to reset the errors. Instead of using external localization device and algorithm, the range sensor information is used for improving the positioning accuracy. Efficient algorithm for identifying the edges from the range sensor information is presented. Scan matching algorithm is implemented so that the positioning error becomes smaller and the map becomes more accurate.

ACKNOWLEDGEMENT

This work was supported by MIC & IITA through IT Leading R&D Support Project.

REFERENCES

Atsushi Fujimori, Takafumi Murakoshi Yoshinao Ogawa, “Navigation and Path-Planning of Mobile Robots with Real-Time”, IEEE, pp.7–12, IEEE ICIT’02 Bangkok, THAILAND

Matteo Golfarelli, Dario Maio, Stefano Rizzi, “Elastic Correction of Dead-Reckoning Errors in Map Building” IEEE/RSJ Intl. Conference on Intelligent Robots and Systems Victoria, B.C., pp.905 - 911, Canada October 1998

Matteo Golfarelli, Dario Maio, Stefano Rizzi, “Correction of Dead-Reckoning Errors in Map”, IEEE Transactions on Robotics and Automation, Vol. 17, No. 1, pp.37-47, February 2001

Dongheui Lee, Woojin Chung, Munsang Kim, “Autonomous Map Building and Smart Localization of the Service Robot PSR”, IEEE October, pp.454–459, 2003

John Leonard, Hugh Durrant-Whyte, Ingemar J.Cox, “Dynamic Map Building for an Autonomous Mobile Robot”, IEEE International Workshop on Intelligent Robots and System, pp.89–96, IROS ’90

F.J. Toledo, J.D. Luis, L.M.Tomás, M.A. Zamora, H. Martínez, “Map Building with Ultrasonic Sensors of Indoor Environments Using”, IEEE, pp.920–925, 2000

Alessandro Corrêa Victorino, Patrick Rives, Jean-Jacques Borrelly, “Localization and Map Building Using a Sensor-Based Control Strategy”, 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.937–942, 2000

Zezhong Xu, Jilin Liu, Zhiyu Xiang, Han Li, “Map Building for Indoor Enviroment with Laser Range Scanner”, The IEEE 5th International Conference on Intelligent Transportation System, pp.136-140, 2002, Singapore

Li Zhang, Bijoy K. Ghosh, “Line Segment Based Map Building and Localization Using 2D Laser Rangefinder”, 2000 IEEE International Conference on Robotics and Automation, pp.2538–2543, San Francisco, 2000

Jinag Zhu, Nafis Ahmad, Hideichi Nakamoto, nobuto Matsuhira, “Feature Map Building Based on Shape Similarity for Home Robot ApriAttenda”, IEEE International Conference on Robotics and Biomimetics, pp.1030–1035, 2006 Kunming, China