Autonomous Robots 7, 187–200 (1999)

A Stochastic Map Building Method for Mobile Robot using 2-D Laser Range Finder

YOUNG D. KWON AND JIN S. LEE
Department of Electrical Engineering, Pohang University of Science and Technology, San-31, Nam-Gu, Ji-Gok Dong, Pohang, 790-784, South Korea
ydkwon@tau.postech.ac.kr
jsoo@postech.ac.kr

Abstract. This paper presents a stochastic map building method for mobile robot using a 2-D laser range finder. Unlike other methods that are based on a set of geometric primitives, the presented method builds a map with a set of obstacle regions. In building a map of the environment, the presented algorithm represents the obstacles with a number of stochastic obstacle regions, each of which is characterized by its own stochastic parameters such as mean and covariance. Whereas the geometric primitives based map sometimes does not fit well to sensor data, the presented method reliably represents various types of obstacles including those of irregular walls and sets of tiny objects. Their shapes and features are easily extracted from the stochastic parameters of their obstacle regions, and are used to develop reliable navigation and obstacle avoidance algorithms. The algorithm updates the world map in real time by detecting the changes of each obstacle region. Consequently, it is adequate for modeling the *quasi-static environment*, which includes occasional changes in positions of the obstacles rather than constant dynamic moves of the obstacles. The presented map building method has successfully been implemented and tested on the *ARES-II* mobile robot system equipped with a LADAR 2D-laser range finder.

Keywords: map building, mobile robot, environment modeling, laser range finder, stochastic parameters

1. Introduction

A human being exploring untrodden or unfamiliar regions requires a map to learn about the area around him. The same is true for the mobile robot, particularly when it navigates through an unstructured and unknown environment. Since the map may not be available in many cases, the mobile robot may have to build one as it moves around the region. It is difficult, however, to build a good map for several reasons. First, the dead reckoning and the obstacle measurements are not accurate, and thus the correct positions of the mobile robot and the obstacles are difficult to obtain. Second, the environment is composed of a number of irregular shaped obstacles, tiny objects, irregular walls, etc. It is thus difficult to represent these various shapes of objects with just a certain number of fixed geometric primitives such as lines, circles and polygons.Third, the environment may not always be static. The chair in the laboratory can be moved occasionally, or the door can be opened or closed. These occasional changes in position of the objects in the environment will incur a serious mismatch between the map and the real environment. Consequently, the map should be updated continuously to reflect these changes.

Many researchers have addressed these problems (Cho and Lim, 1995; Cho, 1990; Boreinstein and Koren, 1991b; Kwon and Lee, 1996). Many of them used ultrasonic sensors in detecting the obstacles because of their low cost and ease of use. Using ultrasonic sensors, Elfes (1991) introduced a map building method where the environment is represented with a number of two dimensional grid cells and certainty values are assigned to the grid cells. This method was used

to develop a simple and effective navigation algorithm in the static environment. However, extra procedures may be needed to extract the characteristics of the dynamic obstacles such as velocity and moving direction which are necessary for effective navigation. The ultrasonic sensor provides measurement data that are often unreliable, and it is difficult to obtain the correct positions of the mobile robot and the obstacles. To overcome the problem, Crowley (1989) used the Kalman filtering technique in detecting the correct position of the mobile robot from noisy and uncertain data sequences. Leonard et al. (1992) extended the Kalman filter based modeling technique to the dynamically changing environment. They use a mathematical model for the sensor dynamics to predict the obstacle position at the next sampling time. Since it uses simple shapes of beacons such as planes, corners and cylinders, it is in general difficult to represent the obstacles of irregular shapes or highly scattered obstacles. Consequently, the mismatch is likely to happen between the predicted beacons and the real objects, because the plane beacon may be mistakenly predicted as cylinder beacon or vice versa for some obstacles with obscure characteristics.

To cope with the unreliable measurements of the ultrasonic sensors, a number of researchers began to use highly reliable laser range finders in detecting a number of objects in the environment (Vandorpe et al., 1996; Taylor and Probert, 1996; Gonzalez et al., 1995). Since the laser range finder provides two dimensional obstacle data at one scan, various shapes of obstacles around the front side can be detected instantaneously at one sampling. In most of these research works, the obstacles are represented with geometric primitives such as lines, circles (Vandorpe et al., 1996) or ellipses (Taylor and Probert, 1996) by fitting the data into one of the geometric primitives. The parameters which characterize each primitive are determined from the recursive fitting procedure by using the obstacle data from the laser range finder. Because each primitive is limited to either a line or a circle, if the shape of an obstacle is vague, then the obstacle fitted previously as a line could well be fitted as a circle or vice versa. In such a case, an extra procedure is required to identify these changes. Because general obstacles are composed of many irregular shaped objects, the fitting procedure may also fail for the highly scattered environment.

In this paper, we present an effective map building method by using a stochastic representation technique. This method builds a map with a set of obstacle regions which are characterized by their own stochastic parameters. It is able to represent the obstacles of irregular shapes or highly scattered obstacles because it is free from the fitting procedure which is one of the prerequisites in many laser range finder based map building methods. Moreover, the changes in the environment can be detected by using the matching criteria associated with the stochastic parameters. Since the size, the orientation and the approximate position of the obstacle regions can be determined from their own stochastic parameters, the change of environment can be checked by comparing these parameters with some uncertain bounds. If the resulting map is used in obstacle avoidance, the characteristics of dynamic obstacles such as velocity, moving direction or size of the obstacles can be inferred from the stochastic parameters of the obstacle regions, and thus, the performance of the obstacle avoidance algorithm can be enhanced. Since the presented method is based on clustering and averaging procedures of the collected obstacle distance data, and since these procedures are able to cancel out the gaussian random noise, it is possible to cope with the uncertainty in the measurements of the laser range finder.

In the presented algorithm, to collect the distance data of the obstacles, a LADAR 2D laser range finder manufactured by the IBEO Lasertechnik Co. (1992) is used with a 270° range around the mobile robot. The data coming from the laser range finder are first clustered together, and then mapped into a number of clustered regions characterized by sets of stochastic parameters. Each of clustered regions is then put to the test if it is associated with any of the obstacle regions in the world map. If it is associated, the parameters of the associated obstacle regions are updated. When the changes of obstacles are detected, they are reflected accordingly in updating the map. Consequently, the presented method is readily applicable to the *quasi-static environment* where the obstacles move occasionally or slowly. The map of the whole environment is essentially a collection of the obstacle regions characterized by their own stochastic parameters.

This paper is organized as follows. Section 2 introduces the concept of clustered region and describes the data clustering procedure. Section 3 introduces the concept of obstacle region and presents a map building procedure using the obstacle regions. Experimental results are discussed in Section 4, and conclusions and comments on future research are made in Section 5.

2. Data Clustering and Clustered Region

2.1. Data Clustering

Figure 1 shows the scanning range of the laser range finder that covers about 270° of the front side. The center position $\mathbf{p}_c$ of the mobile robot is represented as (x_c, y_c) in terms of the world coordinate frame. The incoming data are numbered in the counterclockwise direction and l_j, $j = 0, 1, \ldots 270$, are the distance values from the center position of the laser range finder to the objects. The object position $\mathbf{p}_j = (x_j, y_j)$ is determined from the measured distance l_j as follows:

$$\begin{aligned}\mathbf{p}_j &= (x_j, y_j)\\ &= \left(x_c + l_j \cos\left(\frac{(j-45)\pi}{180} + \theta_c - \frac{\pi}{2}\right) - l_d \cos\theta_c,\right.\\ &\qquad \left. y_c + l_j \sin\left(\frac{(j-45)\pi}{180} + \theta_c - \frac{\pi}{2}\right) - l_d \sin\theta_c\right)\\ &= (x_c, y_c)\\ &\quad + \left(l_j \cos\left(\frac{(j-45)\pi}{180} + \theta_c - \frac{\pi}{2}\right) - l_d \cos\theta_c,\right.\\ &\qquad \left. l_j \sin\left(\frac{(j-45)\pi}{180} + \theta_c - \frac{\pi}{2}\right) - l_d \sin\theta_c\right)\\ &= \mathbf{p}_c + \mathbf{l}_j\end{aligned}$$

where l_d is the distance from the center of the laser scanner to that of the mobile robot and θ_c is the angle of the mobile robot measured counterclockwise from the positive x-axis.

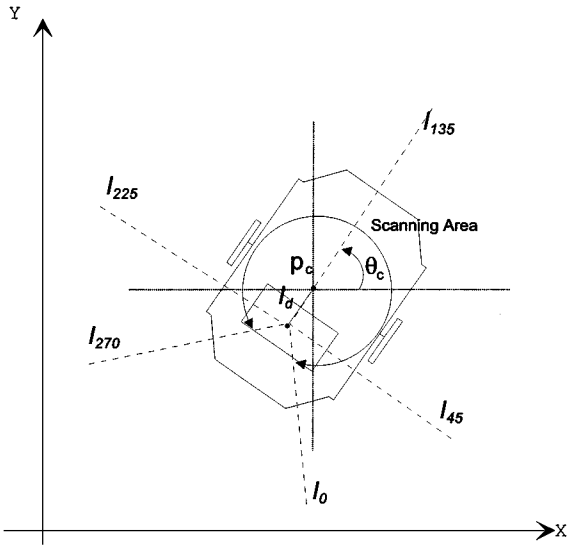

Figure 1. Scanning range of laser range finder.

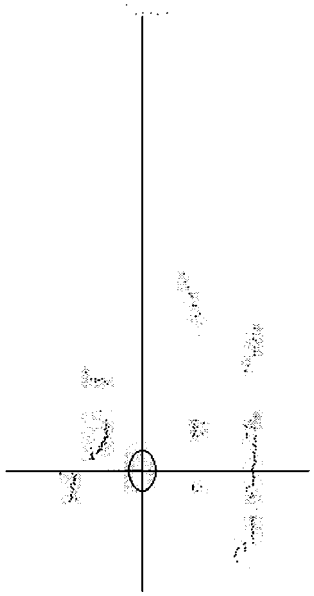

Figure 2. The laser scanner data.

A set of object positions obtained from one scan are as shown in Fig. 2. As noticed from the figure, these object positions can be grouped into several regions at a glance. Grouping the data and separating the regions can be achieved simply by checking the distance between the two successive object positions. Starting from $j = 0$, the distance between the two adjacent positions $\mathbf{p}_j$ and $\mathbf{p}_{j+1}$ are compared. If it is smaller than the preset threshold value D_{th}, then $\mathbf{p}_j$ and $\mathbf{p}_{j+1}$ are considered to be in the same region. In other words, if

$$\|\mathbf{p}_{j+1} - \mathbf{p}_j\| < D_{\text{th}},$$

then $\mathbf{p}_j$ and $\mathbf{p}_{j+1}$ belong to the same region. The threshold value D_{th} is set to 20 cm in our experiment, which takes the mobile robot size into account. If the above condition is not satisfied, then $\mathbf{p}_{j+1}$ belongs to a new region that is different from the region $\mathbf{p}_j$ belongs to. At this point, one region is completed at $\mathbf{p}_j$ and a new region starts from $\mathbf{p}_{j+1}$. This procedure continues until j reaches 269. If a noisy range point is gathered by the laser range finder, an obstacle may be divided into two clustered regions. Since the clustered regions are temporary and used only in the matching procedure, the two clustered regions will be merged into one obstacle region after the matching and updating procedure is

completed. The detailed updating and matching procedure will be presented in Section 3. If the number of object positions in a certain region is too small, then the region is considered negligible and is not counted as a region. The minimum number of object positions that are supposed to be in the region is set by using the parameter n_l. Undoubtedly, the total number of clustered regions depends critically on n_l. The acquired data are then grouped into a number of clustered regions, which are denoted by R_i's where $i = 1, \ldots, N$.

2.2. Clustered Region

The proposed stochastic representation method represents each clustered region R_i as

$$R_i : \{\mathbf{m}_i, \mathbf{C}_i, \boldsymbol{\lambda}_i, \theta_i\},$$

where $\mathbf{m}_i = (m_{xi}, m_{yi})$, $\mathbf{C}_i$, $\boldsymbol{\lambda}_i = (\lambda_{1i}, \lambda_{2i})$ and θ_i represent respectively the mean vector of object positions in R_i, the covariance matrix of R_i, the vector of eigenvalues of $\mathbf{C}_i$ and the direction of the major eigenvector of $\mathbf{C}_i$ measured from the x-axis. As described in the next section, these parameters are used to update the map. When the clustering procedure is completed for R_i, the mean vector of object positions $\mathbf{m}_i$ is then determined as

$$\mathbf{m}_i = (m_{xi}, m_{yi}) = \left(\frac{\sum_{k \in R_i} x_k}{n_i}, \frac{\sum_{k \in R_i} y_k}{n_i} \right),$$

where n_i is the number of object positions in R_i. Also, the covariance matrix $\mathbf{C}_i$ is computed as

$$\mathbf{C}_i = \begin{bmatrix} \sigma_{xi}^2 & \sigma_{xyi} \\ \sigma_{xyi} & \sigma_{yi}^2 \end{bmatrix},$$

where σ_{xi}^2 and σ_{yi}^2 represent respectively the variances of x and y coordinates of the object positions in R_i and σ_{xyi} is the covariance of the object positions in R_i. These parameters are computed as follows:

$$\sigma_{xi}^2 = \frac{\sum_{k \in R_i} (x_k - m_{xi})^2}{n_i} = \frac{\sum_{k \in R_i} x_k^2}{n_i} - m_{xi}^2,$$

$$\sigma_{yi}^2 = \frac{\sum_{k \in R_i} (y_k - m_{yi})^2}{n_i} = \frac{\sum_{k \in R_i} y_k^2}{n_i} - m_{yi}^2,$$

$$\sigma_{xyi} = \frac{\sum_{k \in R_i} (x_k - m_{xi})(y_k - m_{yi})}{n_i} = \frac{\sum_{k \in R_i} x_k y_k}{n_i} - m_{xi} m_{yi}.$$

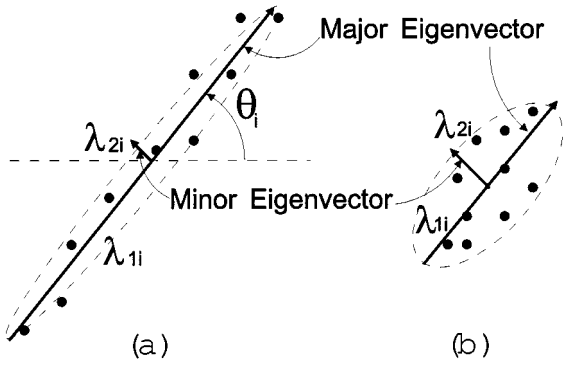

Figure 3. Eigenvalues and eigenvectors.

As shown in Fig. 3, the eigenvalues of the covariance matrix indicate the size of the data set and the corresponding eigenvectors represent the orientation of the data set. The vector of eigenvalues $\boldsymbol{\lambda}_i$ is determined as $\boldsymbol{\lambda}_i = (\lambda_{1i}, \lambda_{2i})$, where λ_{1i} is its larger eigenvalue and λ_{2i} is its smaller eigenvalue. λ_{1i} and λ_{2i} are computed as follows :

$$\lambda_{1i}^2 = \frac{\sigma_{xi}^2 + \sigma_{yi}^2 + \sqrt{\left(\sigma_{xi}^2 - \sigma_{yi}^2\right)^2 + 4\sigma_{xyi}^2}}{2}$$

and

$$\lambda_{2i}^2 = \frac{\sigma_{xi}^2 + \sigma_{yi}^2 - \sqrt{\left(\sigma_{xi}^2 - \sigma_{yi}^2\right)^2 + 4\sigma_{xyi}^2}}{2}.$$

These two eigenvalues clearly indicate how the object positions in the clustered region are scattered from the mean. If $|\lambda_{1i}| \gg |\lambda_{2i}|$ as shown in Fig. 3(a), the object positions are very much aligned as commonly found in the obstacles such as wall. If $|\lambda_{1i}| \approx |\lambda_{2i}|$ as shown in Fig. 3(b), the object positions are widely scattered as commonly found in the tiny obstacles located closely together. Finally, the angle of the major eigenvector θ_i from the x-axis is computed as

$$\theta_i = \tan^{-1}\left(\frac{\lambda_{1i}^2 - \sigma_{xi}^2}{\sigma_{xyi}} \right).$$

3. Obstacle Region and Map Building Procedure

3.1. Obstacle Region

Following the data clustering procedure as described in the previous section, we obtain a series of clustered regions R_i's at each sampling time. The stochastic parameters of R_i indicate where the clustered obstacles

are and how large they are. For example, the mean vector $\mathbf{m}_i$ of R_i represents the approximate center position of R_i and the vector of eigenvalues λ_i represents the size of R_i occupied by the clustered obstacles as shown in Fig. 3. The environment can thus be effectively described by introducing the concept of region characterized by its own stochastic parameters. The clustered regions themselves, however, are not appropriate to build a map because their parameters are computed every sampling time from the laser scanner data. In other words, the same obstacles could be represented by different R_i's for different positions and orientations of the mobile robot. Since the subscript i of R_i is assigned every sampling time in the counterclockwise direction, it could well be different for the same clustered obstacles when the mobile robot is located at different position and orientation. Consequently, the clustered regions R_i's are not appropriate to build a map and a new set of obstacle regions M_k's that are independent of the posture of the mobile robot are introduced as elements of the map. The subscript k of M_k ranges from $k = 1, 2, \ldots, L$, where L is the total number of obstacle regions in the map. The structure of each M_k is similar to those of R_i and consists of $\{\bar{\mathbf{m}}_k, \bar{\mathbf{C}}_k, \bar{\lambda}_k, \bar{\theta}_k\}$, where $\bar{\mathbf{m}}_k$, $\bar{\mathbf{C}}_k$, $\bar{\lambda}_k$, and $\bar{\theta}_k$ represent respectively the mean vector of the object positions in M_k, the covariance matrix of M_k, the vector of eigenvalues of $\bar{\mathbf{C}}_k$ and the direction of the major eigenvector of $\bar{\mathbf{C}}_k$. Whereas R_i's are temporary obstacle regions to build M_k's, M_k's themselves represent the actual obstacle regions as elements of the map. The parameters of R_i's are newly assigned each sampling time, but the parameters of M_k are preserved and updated only when some R_i is associated with M_k. R_i is said to be associated with M_k when R_i is roughly aligned and overlapped with the obstacle region that is already described as M_k. It's more precise definition is given in Section 3.3. A simple example is described in Fig. 4 where R_i's and M_k's are shown for three different places in the path of the mobile robot. In Fig. 4(a), R_i's and M_k's are initially the same. Each R_i is assigned to each M_k, because R_i's are the only information available. In this case, M_2 does not represent the full obstacle since the part described as a dashed line is not detected at this time. At point B in Fig. 4(b), however, the full obstacle is detected as R_2, and M_2 is updated accordingly to include the undetected part. R_1 in Fig. 4(b) is supposed to be the same as that of Fig. 4(a), but may be slightly different due to sensor noises and possible moves of obstacles. Consequently, the parameters of M_1 may be updated by using the parameters of new R_1. When the mobile robot is located at point C as in Fig. 4(c), the obstacle described as M_2 in Fig. 4(b) is now identified as R_1 and the one described as M_1 in Fig. 4(b) is now identified as R_2. Consequently, as with the previous procedures, the parameters of M_1 are updated with those of R_2 and the parameters of M_2 are updated with those of R_1.

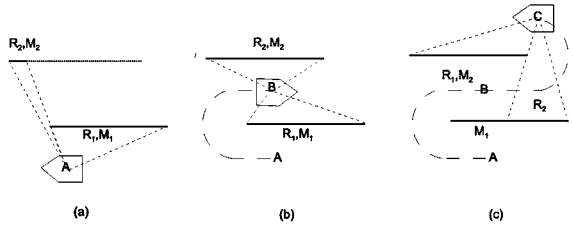

Figure 4. Difference between R_i and M_k.

3.2. Map Building

The world map composed of several obstacle regions may be updated by using the stochastic parameters of the clustered regions determined from the laser range finder every sampling time. Generally, there are many obstacle regions in a world map, but only some of them are associated with the clustered regions. Consequently, it will be time consuming to test all of these obstacle regions to see if they are associated with the clustered regions. In the presented map building method, the obstacle regions located within some distance from the mobile robot are only considered to test its association with the clustered region. We call this set of obstacle regions as local map. To select obstacle regions which included in the local map, we introduce the concept of active circle that represent the region around the mobile robot within the distance d_c. Figure 5 shows the example of an active circle. The local map is composed of M_1, M_2 and M_4. The presented world map building method then consists of the following steps.

- *Step 1.* Determining clustered regions R_i's at the present sampling time.
- *Step 2.* Selecting obstacle regions that are located within the local map.
- *Step 3.* Determining the obstacle region M_k's that are associated with R_i's and updating their parameters.
- *Step 4.* Deleting obstacle regions that are in the field of view but are not associated with any of R_i's.

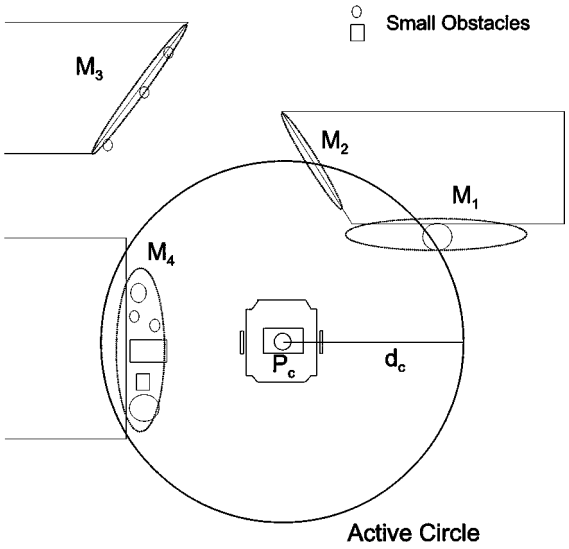

Figure 5. Local map and active circle.

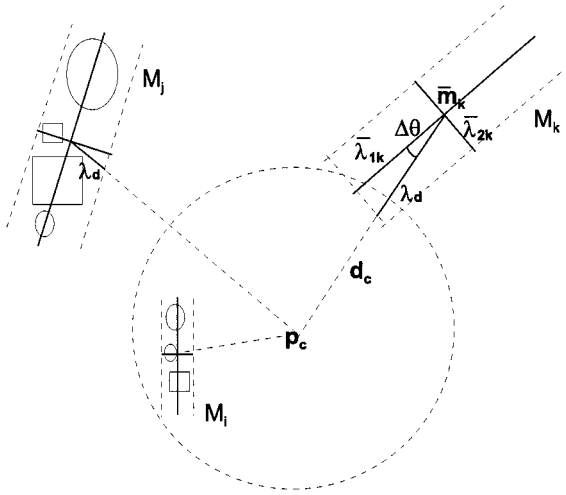

Figure 6. Determination of the local map.

- *Step 5.* Creating new obstacle regions from R_i's which are not associated with any of the M_k's in the present world map.

In Step 1, a set of clustered regions R_i's are first determined from the laser range finder data. In Step 2, obstacle regions included in the local map are selected. Each obstacle region M_k is included in the present local map, if the following condition is satisfied:

$$|\mathbf{p}_c - \bar{\mathbf{m}}_k| < d_c + \lambda_d,$$

where $\mathbf{p}_c$ is the center position of the mobile robot,

$$\lambda_d = \begin{cases} \dfrac{\bar{\lambda}_{1k}}{\cos\Delta\theta} & \text{if } \Delta\theta < \tan^{-1}\dfrac{\bar{\lambda}_{2k}}{\bar{\lambda}_{1k}} \\ \dfrac{\bar{\lambda}_{2k}}{\sin\Delta\theta} & \text{otherwise} \end{cases},$$

and

$$\Delta\theta = |\bar{\theta}_k - \angle(\mathbf{m}_k - \mathbf{p}_c)|.$$

λ_d here represents the span of the obstacle region from $\bar{\mathbf{m}}_k$ along the direction of the vector $\mathbf{p}_c - \bar{\mathbf{m}}_k$ and $\Delta\theta$ represents the orientation difference between the major axis of M_k and the vector $\mathbf{p}_c - \bar{\mathbf{m}}_k$. Consequently, if $\Delta\theta$ is zero degree, then the obstacle region is completely aligned with the vector $\mathbf{p}_c - \bar{\mathbf{m}}_k$ and λ_d becomes equal to $\bar{\lambda}_{1k}$. On the other hand, if $\Delta\theta$ is 90°, then λ_d becomes equal to $\bar{\lambda}_{2k}$. As shown in Fig. 6, if the above condition is satisfied, then all or part of the obstacles included in M_k may be located inside the active circle. Then, each obstacle region in the local map is put to the test in Step 3 to determine which of M_k's are associated with the present clustered regions. Whereas the clustered regions are newly defined every sampling time, the obstacle regions are created, updated or deleted depending on whether each obstacle region is associated with any of the clustered regions. Consequently, Step 3 determines which obstacle regions are associated with the current clustered regions and update their parameters accordingly. In Step 4, we delete the obstacle regions from the world map which are not associated with any of the clustered region. Also, if a clustered region R_i is not associated with any of M_k's, then it is assigned as a new obstacle region in Step 5. Step 1, Step 2, and Step 5 are straightforward, but Step 3 and Step 4 are more involved and explained in more detail in the following subsection.

3.3. Updating and Deleting Obstacle Regions

Since the concept of association is frequently used in updating and deleting the obstacle regions, we define its term more precisely. A clustered region R_i is said to be associated with an obstacle region M_k if R_i and M_k are overlapped with each other. For example, if the obstacles in M_k are not changed at all, then all of the obstacles included in M_k are detected again as R_i and its parameters are nearly the same as those of M_k. Also, if some obstacles included in M_k are moved or some obstacles are added to M_k, then the reduced or enlarged

part of M_k may be detected as a clustered region R_i. For all these cases, M_k and R_i are said to be associated.

More precisely, a clustered region R_i is said to be associated with an obstacle region M_k if the following three conditions hold.

1. The difference between the orientations of the major axes of R_i and M_k is smaller than some uncertainty bound
2. The mean vector $\mathbf{m}_i$ of R_i is located within the distance $\bar{\lambda}_{2k}$ from the major axis of M_k
3. R_i is located within the distance $\bar{\lambda}_{1k}$ from the mean vector $\bar{\mathbf{m}}_k$ of M_k

Condition (1) checks if the angle difference between the major axes of M_k and R_i is sufficiently small. If the difference is larger than the uncertainty bound σ_θ, then some obstacles that used to be in M_k may have been moved outside of M_k. σ_θ here is defined as $\sigma_\theta = \tan^{-1}\frac{\bar{\lambda}_{2k}}{\bar{\lambda}_{1k}}$. As shown in Fig. 7, σ_θ represent the upper limit of the angle variation of the major axis of M_k assuming that the obstacles are fixed. Condition (1) is then equivalent to $|\bar{\theta}_k - \theta_i| < \sigma_\theta$. Condition (2) checks whether M_k and R_i are closely located by examining the distance from the mean position of R_i to the major axis of M_k. Once the condition (1) is satisfied and if the distance is smaller than $\bar{\lambda}_{2k}$, then M_k and R_i are aligned and closely located with each other. If we define $\bar{\mathbf{m}}_k$ as the mean vector of M_k and $\bar{\phi}_k$ as $(\cos\bar{\theta}_k, \sin\bar{\theta}_k)$, then any point $\mathbf{p}$ on the major axis of M_k can be formulated as

$$\bar{\phi}_k \cdot \mathbf{Q}_1 \cdot (\mathbf{p} - \bar{\mathbf{m}}_k)^T = 0,$$

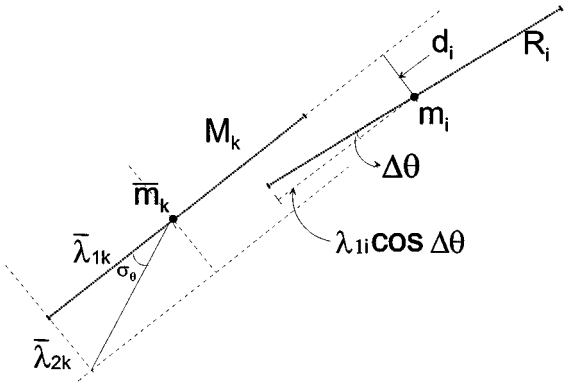

Figure 7. Example of associated case.

where

$$\mathbf{Q}_1 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

The distance d_i from the major axis of M_k to the mean vector $\mathbf{m}_i$ of R_i is then computed as

$$d_i = |\bar{\phi}_k \cdot \mathbf{Q}_1 \cdot \boldsymbol{\Delta}\mathbf{m}^T|,$$

where $\boldsymbol{\Delta}\mathbf{m} = \mathbf{m}_i - \bar{\mathbf{m}}_k$. Condition (2) is then equivalent to $d_i < \bar{\lambda}_{2k}$, where the minor eigenvalue $\bar{\lambda}_{2k}$ is used as an uncertainty bound of M_k along the direction of its minor eigenvector. On the other hand, condition (3) checks whether the associated R_i is roughly overlapped with M_k. It is then equivalent to

$$|\bar{\phi}_k \cdot \Delta\mathbf{m}^T| < \bar{\lambda}_{1k} + \lambda_{1i}$$

where $|\bar{\phi}_k \cdot \Delta\mathbf{m}^T|$ is the projection if $\Delta\mathbf{m}^T$ on the major axis of M_k and $\bar{\lambda}_{1k}$ is used here as an uncertainty bound of M_k along the direction of its major eigenvector. As shown in Fig. 7, if conditions (1), (2) and (3) hold, then R_i is associated with M_k and vice versa. Now, condition (3) is further divided according to whether the obstacles in M_k are fixed or moved. If all of the obstacles are static and fixed, then $|\bar{\phi}_k \cdot \Delta\mathbf{m}^T|$ will be close to zero and the obstacles previously identified as M_k may be detected again as R_i with small disturbances in its parameters. However, if the obstacles in M_k is moved to another location or some obstacles are added to M_k or subtracted from it, then the parameters of M_k are changed. The presented technique is able to detect these moves by comparing the size of the eigenvalues and the difference between the mean positions. In fact, $|\bar{\phi}_k \cdot \Delta\mathbf{m}^T|$ represents the difference between the mean positions of R_i and M_k and the size of the major eigenvectors $\bar{\lambda}_{1k}$ and λ_{1i} represent the approximate sizes of the regions. Consequently, by monitoring these parameters, we can detect the change of obstacles such as movement, addition or subtraction. For example, if $|\bar{\phi}_k \cdot \Delta\mathbf{m}^T|$ and $|\bar{\lambda}_{1k} - \lambda_{1i}|$ are both nearly zero, then the obstacle region M_k and the clustered region R_i are closely aligned as shown in Fig. 8(a).

Also, if $|\bar{\phi}_k \cdot \Delta\mathbf{m}^T|$ is larger than $\bar{\lambda}_{1k}$ and the $|\bar{\lambda}_{1k} - \lambda_{1i}|$ is nearly zero, then M_k is moved to R_i as shown in Fig. 8(c). Consequently, the condition (3) can be divided into the following three conditions according to the values of $|\bar{\phi}_k \cdot \Delta\mathbf{m}^T|$ and of $|\bar{\lambda}_{1k} - \lambda_{1i}|$:

(3.i) $|\bar{\phi}_k \cdot \Delta\mathbf{m}^T| < \bar{\lambda}_{1k}$ and $|\bar{\lambda}_{1k} - \lambda_{1i}| < \lambda_{\text{th}}$
→ Fixed obstacle region

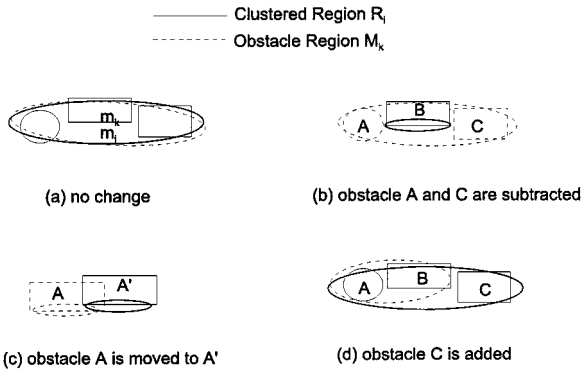

Figure 8. Example of changes in M_k.

(3.ii) $|\bar{\phi}_k \cdot \Delta\mathbf{m}^T| < \bar{\lambda}_{1k} + \lambda_{1i}$ and $|\bar{\lambda}_{1k} - \lambda_{1i}| > \lambda_{\text{th}}$
$\rightarrow$ Reduced or enlarged obstacle region
(3.iii) $\bar{\lambda}_{1k} < |\bar{\phi}_k \cdot \Delta\mathbf{m}^T| < \bar{\lambda}_{1k} + \lambda_{1i}$ and
$|\bar{\lambda}_{1k} - \lambda_{1i}| < \lambda_{\text{th}} \rightarrow$ Moved obstacle region,

where λ_{th} is the predefined threshold value. Figure 8 shows the examples that satisfy each of the above conditions. Dotted ellipses and solid ellipses in Fig. 8 represent respectively the obstacle regions and the clustered regions. Condition (3.i) indicates that the center positions and the sizes of M_k and of R_i are about the same and thus the obstacles in M_k are detected again as the clustered region R_i(Fig. 8(a)). Condition (3.ii) represents the case where some obstacles are added (Fig. 8(d)) to or subtracted (Fig. 8(b)) from M_k. The size of M_k is enlarged or reduced in this case depending on addition or subtraction of the obstacles. Finally, condition (3.iii) indicates that the obstacles in M_k must have been moved to those in R_i(Fig. 8(c)). The size of M_k are not changed in this case but the center position is moved from $\bar{\mathbf{m}}_k$ to $\mathbf{m}_i$.

If the obstacle region M_k and the clustered region R_i satisfy conditions (1), (2) and (3.i), then the obstacles in M_k are fixed and its parameters are updated by using the parameters of R_i. In all other cases, the parameters of M_k are replaced with those of R_i. When the obstacles are fixed, the parameters of M_k are updated as

$$\bar{\mathbf{m}}_k \leftarrow \frac{\bar{\mathbf{m}}_k + \mathbf{m}_i}{2},$$
$$\bar{\lambda}_{1k}^2 \leftarrow \sigma_x^2 + \sigma_y^2 + \|\mathbf{\Lambda}\|,$$
$$\bar{\lambda}_{2k}^2 \leftarrow \sigma_x^2 + \sigma_y^2 - \|\mathbf{\Lambda}\|,$$

where σ_x^2 and σ_y^2 are the variances of the merged region and $\mathbf{\Lambda} = [\sigma_x^2 - \sigma_y^2, 2\sigma_{xy}]^T$, and σ_{xy} is the covariance of the merged region. σ_x^2, σ_y^2 and $\mathbf{\Lambda}$ are all computed from the parameters of M_k and R_i, and $\sigma_x^2 + \sigma_y^2$ is computed as

$$\sigma_x^2 + \sigma_y^2 \leftarrow \frac{1}{2}\|\bar{\boldsymbol{\lambda}}_k\|^2 + \frac{1}{2}\|\boldsymbol{\lambda}_i\|^2 + \frac{1}{4}\|\mathbf{\Delta m}\|^2,$$

and the vector $\mathbf{\Lambda}$ is determined as

$$\mathbf{\Lambda} \leftarrow \frac{1}{4}\begin{bmatrix} \cos 2\bar{\theta}_k & \cos 2\theta_i \\ \sin 2\bar{\theta}_k & \sin 2\theta_i \end{bmatrix}\begin{bmatrix} \bar{\boldsymbol{\lambda}}_k \mathbf{Q}_2 \bar{\boldsymbol{\lambda}}_k^T \\ \boldsymbol{\lambda}_i \mathbf{Q}_2 \boldsymbol{\lambda}_i^T \end{bmatrix} + \frac{1}{4}\begin{bmatrix} \mathbf{\Delta m Q}_2 \mathbf{\Delta m}^T \\ \mathbf{\Delta m Q}_3 \mathbf{\Delta m}^T \end{bmatrix},$$

where

$$\mathbf{Q}_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad \mathbf{Q}_3 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Finally, the slope of the major axis of M_k is updated as

$$\bar{\theta}_k \leftarrow \frac{1}{2}\sin^{-1}\left(\frac{2\sigma_{xy}}{\bar{\boldsymbol{\lambda}}_k \mathbf{Q}_2 \boldsymbol{\lambda}_k^T}\right),$$

where $\bar{\boldsymbol{\lambda}}_k$ is the eigenvalue vector of the merged region. After updating the parameters of the associated obstacle regions by using the corresponding clustered regions, we now move on to delete the obstacle regions that are in the field of view and not associated with any of the clustered regions. The field of view is the scanning range of the laser scanner where the distance of view depends on whether a clustered region R_i exists in the direction of view. If R_i exists in the direction of view, then the distance of view is the distance up to R_i. Otherwise, it stretches up to the detection limit of the active circle d_c. An example of the field of view is shown in Fig. 9. As shown in Fig. 9, the obstacle regions M_3 and M_4 are located in the field of view and they are not associated with any of R_i's. Consequently, they must have been moved to another location and must be deleted from the map. The obstacle regions M_1 and M_2 are not associated with any of R_i's either, but they are not deleted from the map because they are outside of the field of view. In other words, we do not know whether M_1 or M_2 has been moved or not, because they have not been currently detected. The obstacle region M_7 is associated with R_4, but it is not updated because part of it is located outside of the field of view. In order to check whether the obstacle region that is not associated with any of R_i's

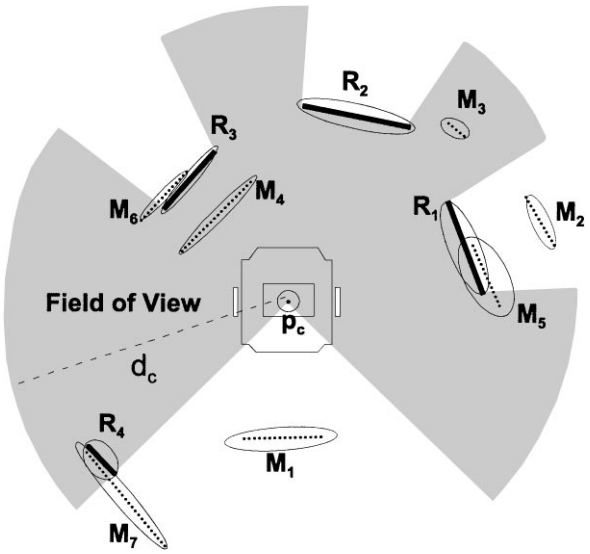

Figure 9. The field of view, the clustered regions and the obstacle regions.

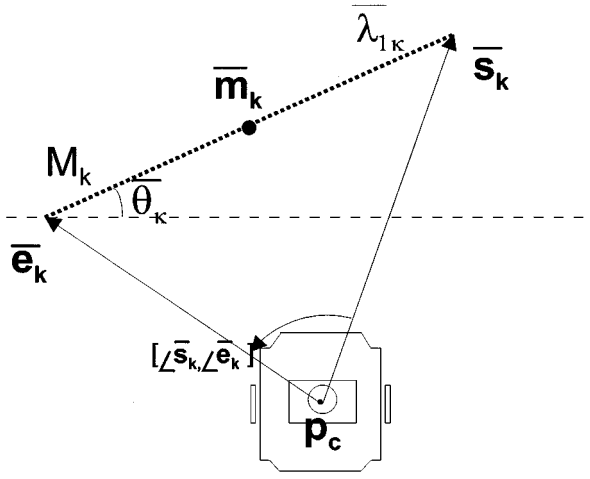

Figure 10. Representation of angle interval of M_k.

is located inside of the field of view, we use the angle interval of the obstacle region. The angle interval of M_k is the interval of directions along which the laser range finder detects the obstacles in M_k. As shown in Fig. 10, the angle interval is determined from the major eigenvector of M_k. Its major eigenvector is described as $\bar{\mathbf{e}}_k - \bar{\mathbf{s}}_k$, where $\bar{\mathbf{e}}_k$ and $\bar{\mathbf{s}}_k$ represent its start and end positions in the sensor coordinate frame. The angle interval of M_k is then determined as $[\angle\bar{\mathbf{s}}_k, \angle\bar{\mathbf{e}}_k]$, where $\bar{\mathbf{s}}_k = \bar{\mathbf{m}}_k + \bar{\lambda}_{1k}\bar{\phi}_k - \mathbf{p}_c$, $\bar{\mathbf{e}}_k = \bar{\mathbf{m}}_k - \bar{\lambda}_{1k}\bar{\phi}_k - \mathbf{p}_c$, and $\mathbf{p}_c$ is the center position of the mobile robot. As shown in Fig. 9, the angle intervals of the obstacle regions such as M_1 and M_7 range outside of $[0, 270]$ and their parameters remain unchanged. On the other hand, the angle interval of R_1 fully covers that of M_2 and the distance of M_2 is longer than the that of R_1. Consequently, M_2 is located behind R_1 and its parameters remain unchanged.

Finally, if R_i is not associated with any of the existing M_k's, then a new obstacle region is created with R_i and is included in the map. For example, R_2 in Fig. 9 is not associated with any of M_k's, and a new obstacle region is created with R_2.

3.4. Uncertainty of the Measurement

The object position $\mathbf{p}_j$ determined from dead reckoning and measured distance value l_j is bound to have uncertainty, which may cause mismatch between the map and the real environment. In general, there are two types of uncertainty sources: one is the noise inherent in the measured data, and the other is the mechanical aspects of the mobile robot such as slippage and upsetting events. The gaussian random noise inherent in the object position $\mathbf{p}_j$ is essentially cancelled out in the presented algorithm because all of the object positions included in the same clustered region R_i are averaged to determine the stochastic parameters of R_i. However, the uncertainty caused by mechanical aspects of the vehicle is relatively large and some localization algorithm may have to be employed to overcome them. The presented method deals only with a map building method and does not address the localization issue. However, as long as the uncertainty bound is not too large, the algorithm can cope with this type of uncertainty by taking its bound into account in the matching procedure.

In order to see how the uncertainty is reflected onto the object position, we write the object position $\mathbf{p}_j$ introduced in Section 2.1 as

$$\begin{aligned}\mathbf{p}_j &= \mathbf{p}_c + \mathbf{l}_j \\ &= \mathbf{p}_c^* + \Delta\mathbf{p}_c + \mathbf{l}_j^* + \Delta\mathbf{l}_j \\ &= \mathbf{p}_j^* + \Delta\mathbf{p}_c + \Delta\mathbf{l}_j,\end{aligned}$$

where $\mathbf{p}_j^*$ represents the nominal value of $\mathbf{p}_j$, $\Delta\mathbf{p}_c$ is the position error of the mobile robot and $\Delta\mathbf{l}_j$ is the error of the object position due to the orientation error $\Delta\theta$ of the mobile robot. $\Delta\mathbf{p}_c$ causes a proportional effect on all of the object positions included in R_i and

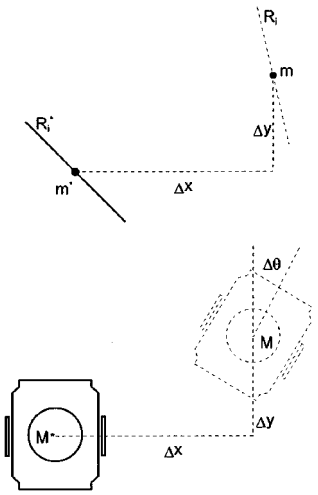

Figure 11. Uncertainties of the robot position.

cause R_i to move by Δx and Δy as shown in Fig. 11. Meanwhile, $\Delta \mathbf{l}_j$ is determined as

$$\begin{aligned}\Delta \mathbf{l}_j &= \mathbf{l}_j - \mathbf{l}_j^* \\ &= l_j[\cos(\theta_j + \Delta\theta_c), \sin(\theta_j + \Delta\theta_c)] \\ &\quad - l_j[\cos\theta_j, \sin\theta_j] \\ &= l_j[\cos(\theta_j + \Delta\theta_c) - \cos\theta_j, \\ &\qquad \sin(\theta_j + \Delta\theta_c) - \sin\theta_j] \\ &= 2l_j \sin\frac{\Delta\theta_c}{2}\left[\sin\left(\theta_j + \frac{\Delta\theta_c}{2}\right),\right. \\ &\qquad \left.\cos\left(\theta_j + \frac{\Delta\theta_c}{2}\right)\right].\end{aligned}$$

As long as $\Delta\theta_c$ is not zero, $\Delta \mathbf{l}_j$ depends on l_j and θ_j, and cause R_i to rotate as shown in Fig. 11.

Consequently, the error in the clustered region R_i due to the error in the robot position can be described as the proportional and rotational errors in the clustered region and they can be reflected on the matching conditions introduced in the previous section. $\Delta \mathbf{l}_j$ can be taken into account in matching condition (1) in Section 3.3 by writing it as $|\bar{\theta}_k - \theta_i| < \sigma_\theta + \Delta L_{\max}$, where $\Delta L_{\max}$ is the threshold constant determined from the maximum bound of $\Delta\theta_c$. $\Delta \mathbf{p}_c$ can also be considered in matching condition (2) in Section 3.3 by writing it as $d_i < \bar{\lambda}_{2k} + \Delta p_{\max}$, where $\Delta p_{\max}$ is the maximum bound of $|\Delta \mathbf{p}_c|$.

4. Experimental Result

The presented algorithm has been implemented on our *ARES-II* mobile robot system (Fig. 12) equipped with a 2D-laser scanner mounted on the mobile platform. The width and the length of the *ARES-II* system are about 36 cm $\times$ 60 cm. The presented algorithm has been tested first in the static environment where all of the obstacles are fixed in their positions during the test. Figure 13 shows part of our laboratory consisting of a number of obstacles such as chairs, boxes, oscilloscopes, power supplies, experiment tables, etc.

These obstacles are in general difficult to represent with the restricted beacons such as line, ellipse or cell based mapping methods. To build a map for the environment, we first move the mobile robot at 30 cm/s from *A* to *B* where the door was left open as in Fig. 13. The result of the first test is given in Fig. 14.

The presented algorithm describes the environment fairly well with a set of stochastic obstacle regions. The dotted lines represent the actual layout of the laboratory, and the solid line represent the obstacles detected during the first move. In order to simplify the description, a line or a circle according to its size is made to represent each obstacle region. If the obstacle size is smaller than 20 cm, it is described as a circle, and otherwise, it is described as a line. When it is described as a line, its length is set to the value of its major eigenvalue. In Fig. 14, *P* represent the large power supply

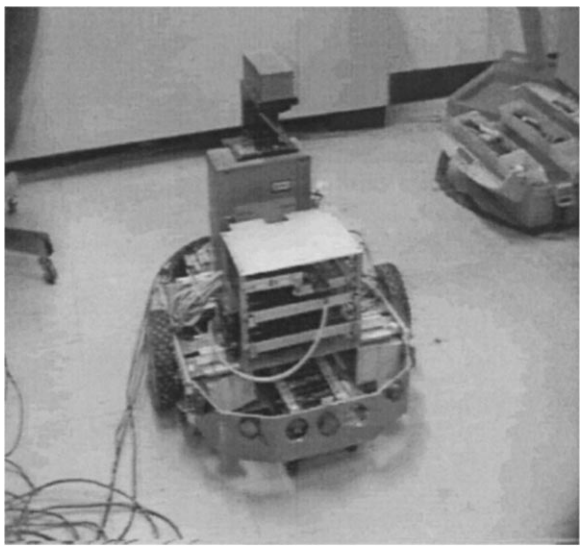

Figure 12. The mobile robot system *ARES-II*.

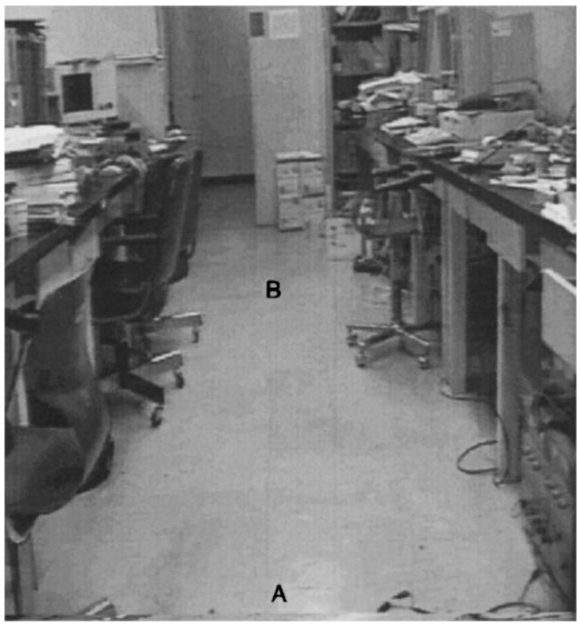

Figure 13. The picture of part of our laboratory.

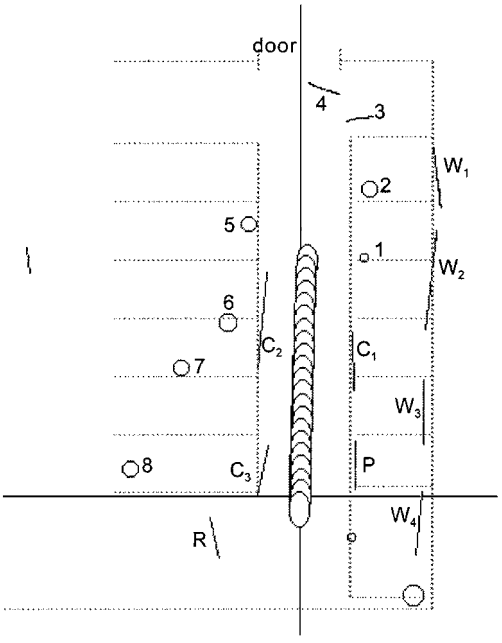

Figure 14. The result of the first test.

whereas R represents the robot controller. W_1,W_2,W_3 and W_4 are parts of the wall and C_1,C_2 and C_3 are the chairs. The obstacles marked with the numbers 1 to 8 indicate the small boxes or sets of small obstacles. Ideally, the wall must have been detected with one obstacle region, but it was divided into 4 regions and was not fully aligned. Some parts of the wall were not detected because of the obstacles in the way such as P and C_1. They were not aligned also to each other because the small obstacles located near the wall were merged to the nearest part of the wall.

Next, the presented algorithm has been tested in the *quasi-static* environment, where the chairs C_5 and C_6 were added, a small box 5 was moved and the door got closed. These changes and the corresponding layout of the laboratory are shown in Fig. 15 and are compared with those for the first test as in Fig. 16.

The presented algorithm reliably detected the changes in the environment as shown in Fig. 17. For example, the laboratory door and the new chair C_6 are shown in Fig. 17. The chair C_5 has also been detected but it is merged with C_1 and P, because it was located closely between C_1 and P.

To represent the small obstacles or differentiate the closely located obstacles, the parameters D_{th} and n_l defined in Section 2.1 should be adjusted accordingly. D_{th} sets the maximum gap between the obstacles belonging to the same clustered region whereas n_l sets the number of data points in the minimum size clustered region. So, if the gap between two obstacles is smaller than D_{th}, they belong one clustered region. Practically, D_{th} is determined to be the width of the mobile robot with some margin of extra space. In other words, if the

Figure 15. The layout of the second test.

Figure 16. The layout for the first test.

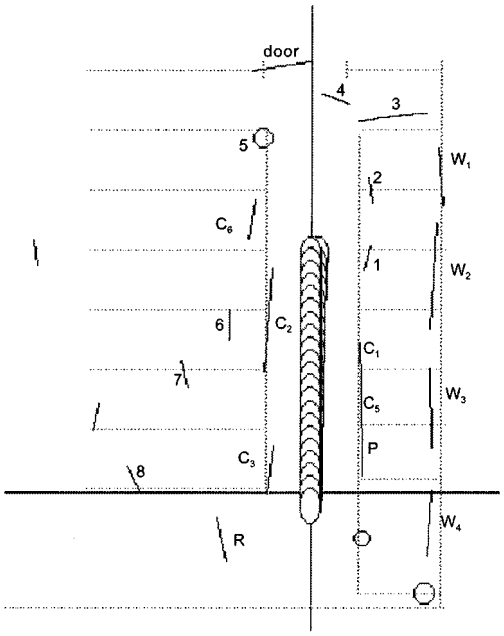

Figure 17. The result of the second test.

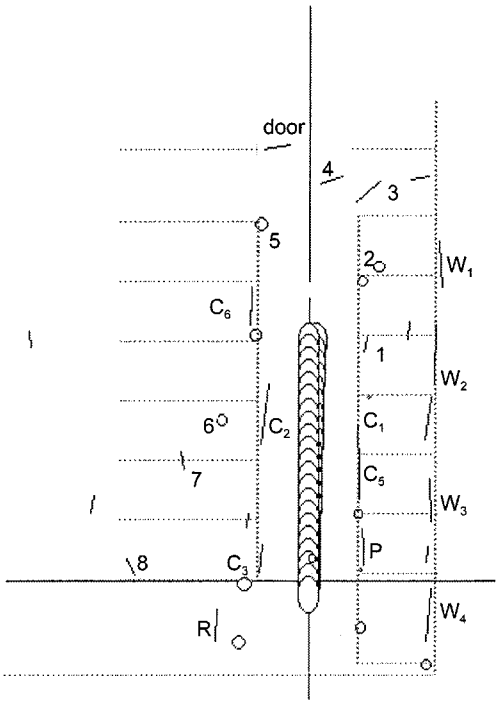

Figure 18. The map with small n_l and D_{th} values.

gap between two obstacles is smaller than D_{th}, then the mobile robot can not pass through the gap. Therefore, the two obstacles are made to belong to the same clustered region to simplify the map. On the other hand, n_l determines the minimum size of the obstacle region that is to be shown in the map. If the number of object positions in R_i is smaller than n_l, R_i is deleted because it is considered too small. If n_l is too small, then even the tiny single objects will be represented as an independent obstacle region. Unless an accurate map is required, n_l and D_{th} is set to be reasonably large to simplify the map. But if necessary, an accurate map can be made by decreasing the values of n_l and D_{th}. Figure 18 is the map with small n_l and D_{th} values whereas Fig. 19 is the map with large n_l and D_{th} values for the layout in Fig. 15.

As you can see, Fig. 18 describes additional details in the map, that is, the obstacles C_1, C_5 and P are separated and the legs of the table between these obstacles are shown. The obstacle 2 in Fig. 17 is now separated into two obstacles here, since it is actually composed of a number of small obstacles. In Fig. 19, however, small obstacles such as 1, 2, 4, 7, 8 in Fig. 14 do not exist anymore. Therefore, by adjusting the parameters n_l and D_{th}, it is possible to determine the number of obstacle regions and the accuracy of the map. For obstacle avoidance, fine map is usually necessary, and n_l

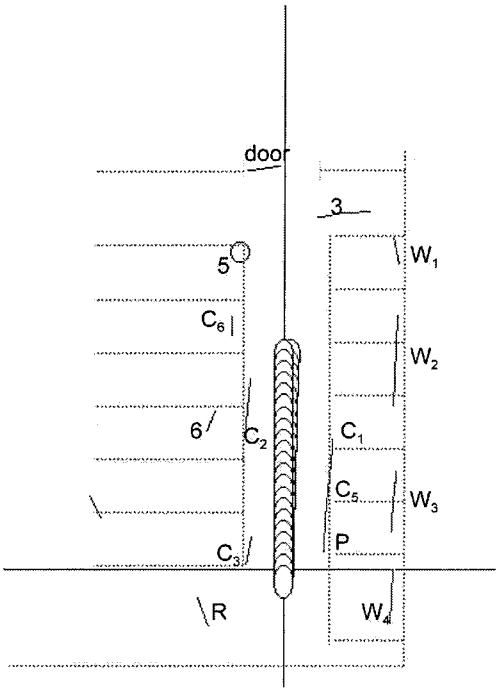

Figure 19. The map with large n_l and D_{th} values.

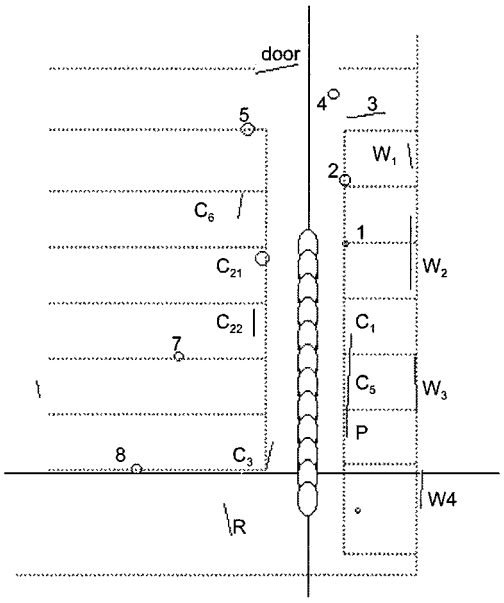

Figure 20. The result of the fast moving situation.

and D_{th} are decreased to describe small obstacles. For global path planning, on the other hand, coarse map is usually sufficient and n_l and D_{th} are increased to reduce processing time and memory consumption.

Finally, we increased the velocity of the mobile robot up to 60 cm/s, leaving all other parameters the same as in Fig. 17. The map is similar to that in Fig. 17, but some parts of the region were left undetected as shown in Fig. 20. Consequently, the map shows that the size of the detected obstacles is a little smaller than those of Fig. 14. In this case, the chair C_2 is separated into two obstacle regions C_{21} and C_{22} since the middle parts of the region have not been detected.

In our experiments, a full cycle of data clustering and map updating took about 200 ms. The LADAR 2-D laser scanner is able to provide 8 scans every second with 0.6° resolution but we used 5 scans every second with 1° resolution.

5. Conclusion

An effective map building method based on stochastic representation technique is presented. The presented algorithm classifies the obstacles in the environment into a number of obstacle regions that are characterized by several stochastic parameters. As shown in the experimental results, our algorithm is able to model the environment successfully with a set of obstacle regions. The stochastic parameters in the obstacle region can represent the obstacles of irregular shapes such as irregular walls and sets of tiny objects. Unlike traditional laser range finder based map building method, our stochastic map building method does not have to specify directly the shapes of the obstacles to be updated since the parameters of each obstacle region fully contains the features of the obstacles. Moreover, the stochastic representation technique updates the world map in real time by detecting the changes of each obstacle region. The presented algorithm is also robust to the uncertainties of the measurement. The uncertainties due to gaussian random noise are cancelled out in stochastic updating procedure, and the uncertainties due to mechanical aspects are readily considered in stochastic matching procedure.

The dynamic environment where the obstacles are constantly moving can also be modeled using this tool once we monitor the center position of each obstacle region. The presented algorithm, however, is essentially an environment modeling tool, and its applications to dynamic environment, path planning, localization and collision avoidance problems remain future works.

Acknowledgment

The authors wish to thank the anonymous reviewers for their valuable comments. Also, financial support from Korea Science and Engineering Foundation through the Automation Research Center at POSTECH is gratefully acknowledged.

References

Borenstein, J. and Koren, Y. 1991(a). The vector field histogram fast obstacle avoidance for mobile robot. *IEEE Transaction on Robotics and Automation*, 7(3):278–288.

Boreinstein, J. and Koren, Y. 1991(b). Histogrammic in motion mapping for mobile robot obstacle avoidance. *IEEE Transaction on Robotics and Automation*, 7(4):535–549.

Crowley, J.L. 1989. World modelling and position estimation for a mobile robot using ultra sonic ranging. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 674–680.

Cho, D.W. 1990. Certainty grid representation for robot navigation by a bayesian method. *Robotics*, 8:159–165.

Cho, D.W. and Lim, J.H. 1995. A new certainty grid based mapping and navigation system for an autonomous mobile robot. *International Journal of Advanced Manufacturing Technology*, 139–148.

Elfes, A. 1986. Sonar based real mapping and navigation system. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1151–1156.

Elfes, A. 1991. Occupancy grids: A stochastic spatial representation for active robot perception. Autonomous Mobile Robot, IEEE Computer Society Press, pp. 60–70.

Gonzalez, J., Stentz, A., and Ollero, A. 1995. A mobile robot iconic position estimator using a radial laser scanner. *Journal of Intelligent Robotics Systems*, 13:161–179.

Gonzalez, R.C. and Woods, R.E. 1992. Digital image processing, Addison-Wesley Publishing.

Kwon, Y.D. and Lee, J.S. 1995. An obstacle avoidance algorithm for mobile robot: The improved weighted safety vector field method. In *Proceedings of the IEEE 10th International Symposium on Intelligent Control*, pp. 441–446.

Kwon, Y.D. and Lee, J.S. 1996. A local path generation method using obstacle vectors and via points. In *Proceeding of WAC 96*, pp. 415–420.

Leonard, J.J., Durrant-Whyte, H.F., and Cox, J.I. 1992. Dynamic map building for an autonomous mobile robot. *The International Journal of Robotics Research*, 11(4):286–298.

"LADAR 2D Controller Software Programmers Manual," IBEO Lasertechnik, 1992.

Taylor, R.M. and Probert, P.J. 1996. Range finding and feature extraction by segmentation of images for mobile robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 95–100.

Vandorpe, J., Van Brussel, H., and Xu, H. 1996. Exact dynamic map building for a mobile robot using geometrical primitives produced by a 2D range finder. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 901–908.

Young D. Kwon is Ph.D candidate student of the department of Electrical Engineering of the Pohang University of Science and Technology (POSTECH). He received the B.S. degree in electronics engineering from Korea Institute of Science and Technology Daejeon, Korea in 1991. He received MS degree in Electrical Engineering from POSTECH. His current research include mobile robot control, multi-agent strategy learning, intelligent control and robot soccer. He is a leader of robot soccer team *BALL BREAKER*, and the team wins a 2nd place in FIRA'98 ASIA-PACIFIC league.

Jin S. Lee received the B.S. degree in electronics engineering from Seoul National University Seoul, Korea in 1975. He received the M.S. degree in electrical engineering and computer science from the University of California, Berkeley in 1980 and the Ph.D. degree in system science from the University of California, Los Angeles in 1984. From 1984 to 1985, he was a member of technical staff in AT & T Bell Laboratories and from 1985 to 1989, he was a senior member of engineering staff in GE Advanced Technology Laboratories. Since 1989, he has been with the Department of Electronics and Electrical Engineering at Pohang University of Science and Technology and he is currently the professor and the chair of the department. His research interests include robotics, adaptive control, intelligent control, and their application to automation systems.