Sonar Based Position Estimation System for an Autonomous Mobile Robot Operating in an Unknown Environment

Seung Kyun Kang* and Jong Hwan Lim**

(*Received June 23, 1998*)

The paper presents a new method of position estimation for robot navigation in a completely unknown environment. The method is different from conventional ones in that it does not need any kind of a priori reference model or man-made landmarks. A series of local maps is built and updated from sonar data while the robot is exploring the unknown environment. Among the constructed local maps, the robot autonomously selects the ones with distinctive features and memorizes them as reference landmarks. An orientation clustering method is developed which enables the robot to extract the features of the map. The maps selected in such a way are then used to estimate the position and orientation of the robot while undertaking the given task. In doing so correspondence indices are defined to determine the corresponding reference map to the current local one among the numerous stored maps. The two maps are matched so as to minimize the discrepancy between them, thus enabling one to estimate the position and orientation of the robot. The usefulness of all these approaches is illustrated with the results produced by a real robot equipped with ultrasonic sensors.

Key Words: Position Estimation, Mobile Robot, Orientation Chustering, Correspondence Indix, Map Match

1. Introduction

Determining the position of a robot relative to a reference coordinate frame or a special object is an important issue in autonomous mobile robot research. In many cases, dead reckoning alone cannot provide sufficient accuracy because its error tends to accumulate without bound. Dead reckoning error comes from the assumption that a revolution of the axle implies a fixed distance traveled by the wheel. Several factors make this assumption inaccurate: wheel slip on the ground, irregularity of the ground surface, etc.. This class of errors is random in nature. Another class of errors involved in dead reckoning are systematic errors, which are not random in nature: imperfection of the wheel shape, different radii between wheels etc. It is therefore generally required to use additional position estimators, e.g., beacon or landmark based estimators.

Beacon and artificial landmark based estimators require, respectively, the emplacement of beacons and the presence of man-made structures in the environment. Leonard and Durrant-Whyte (Leonard, 1992) used an extended Kalman filter approach to estimate the position of a robot from an a priori map of the environment. Hyppa (Hyppa, 1989) made use of the reflective strips in the environment for a rotating laser mounted on the robot. Kleeman (Kleeman, 1989 and Kleeman, 1992) explored a method that combines ultrasonic beacon and dead reckoning data using an extended Kalman filter to estimate the optimal positioning and heading values for a robot.

On the other hand, some researchers utilized natural landmarks for which information used for position and orientation estimation is also used

* Graduate Student, Department of Mechanical Engineering, Cheju National University, Ara-dong, Cheju 690-756, KOREA

** Department of Mechanical Engineering, Cheju National University, Ara-dong, Cheju 690-756, KOREA, email:jhlim@cheju. cheju. ac. kr

for motion planning. Crowley (Crowley, 1986 and Crowley, 1992) developed a feature-based landmark system in which a set of distinctive features is extracted from the sensed data. The features are then matched with the corresponding world model consisting of line segments to estimate the robot's position and orientation. Chatila (Chatila, 1992) has explored a similar method to Crowley's except that the world model used here consists of polygons. Another kind of natural landmark system, called cell-based or iconic systems, was developed in (Elfes, 1987 etc). It works directly on the raw sensed data, minimizing the discrepancy between the raw data and the reference model. The reference models should be given in advance in all of these methods except the work described in Elfes (1987).

Consequently, conventional position estimation can be divided into two different approaches: artificial landmark or beacon systems, and natural landmark systems. The former is more efficient and convenient in estimating the absolute position of the robot, although its prior placement restricts the autonomous behavior of the robot. The latter method utilizes a whole or partial environment of the work space itself. It thus necessitates a matching (or minimization) procedure between a reference model and the sensed data to estimate the robot's position, which usually requires significant computation. Most of the natural landmark systems also need a reference model given in advance.

This paper addresses a new position estimation method utilizing natural landmarks without any a priori information of the environment. The method is different from conventional ones in that it does not need any kind of a priori reference model. It automatically selects reference models from workspace. The underlying idea of this method is that a human memorizes only very impressive objects along a route, such as an old building, a very high tower, or a street corner, and uses them as landmarks. Likewise, a robot also finds distinctive features from its workspace and uses them to estimate its position when it revisits there.

To accomplish this, a series of local maps is continuously built and updated from sonar range data while the robot is exploring its work space. Among the series of local maps, the ones with distinctive features are selected and stored as reference maps (landmarks). The reference maps thus selected are then used to estimate the position and orientation of the robot while performing the given tasks. We utilize the correspondence between the reference and current local maps together with a cell-based matching method that minimizes the discrepancy between them.

2. Local Map Representation and Classification of Its Shapes

The position estimation system in this paper is based on a sonar map. This section describes the concept of a local map, how to update it, and how to classify its features.

2.1 Local map

In this paper, we consider ultrasonic sensors that are cheap and easy to use. Sensed data are processed into a sonar probability map, which is composed of 2-D occupancy girds updated from the well-known Bayesian Updating Model(Lim, 1992 etc). We, however, modified slightly the original model to avoid high computational cost and enable real time operation(Lim, 1994b); cells in an empty region where a sonar beam passes through are not updated (provided they have never been updated before) because the occupied cells can only be the boundary of an object. On the other hand, cells in an empty region that have already been updated from the previous data are still updated because they could possibly be a phantom object due to false readings. The method will lower somewhat the quality of the resulting map ccmpared to the original methed, but it can provide an effective means for the robot to distinguish the boundary of objects from the free space.

At the initial stage of operation, a robot explores its work space to collect information on the area. A factory, office area or corridor is generally too spacious for the robot to construct a fine resolution map of the entire area because the resolution of a map depends largely on the grid

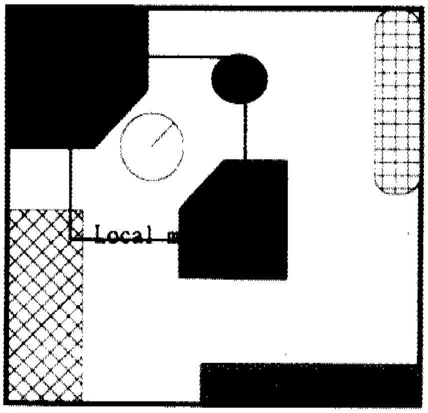


Fig. 1 Configuration of a local map.

size. Therefore, a local map is used to construct a high resolution map in real time with limited memory capacity and computing speed. Another advantage to using a local map is that it is easy and fast to classify the shapes of objects in the map, which is important for position estimation.

Figure 1 shows the configuration of a local map. The size of the local map is 48×48 cells and each cell represents a real-world square of size $0.05 \times 0.05 m^2$. The center of a local map is identical to that of a robot, and the map is continuously translated with the movement of the robot. The orientation of the local map is fixed with respect to the frame of reference.

2.2 Classification of shapes in a local map

The shape of an object in a local map plays an important role in our position estimation system. Different kinds of shapes give different kinds of position information. That is, a linear shape can only provide information on orientation, and a complete circular shape only on position, and a cornered object on both. Accordingly, the robot should have the ability to classify the shapes of objects in reference maps and to memorize what kinds of information they can provide in order to use them for position and orientation estimation. In addition, at the position estimation stage, the robot should check if the shape of the current local map corresponds to that of the reference map to be matched in order to avoid situations in which maps of different shapes are matched together.

The the probability map from our map construction model also provides information on orientation of each cell. The method makes use of the specular reflection property of a sonar sensor, i,e., the sensor cannot detect an object whose surface is not almost normal to the beam path. The orientation of each occupied cell is also updated while the occupancy probability is updated (Lim, 1996). We develop a method that can classify the shapes of objects in a local map using this orientation information in real time.

To classify the shape of objects from the orientation information, the orientations of the occupied cells are clustered into groups according to their values. Then the center of each cluster is estimater of a which is the average angle of the orientations of the cells in each cluster. For example, a corner of a wall that forms a right angle would be clustered into two groups, and the relative angle between the two centers would be about 90°. The number of clusters, therefore, means the number of line segments of objects in the local map, and the value of each center itself represents the orientation of the line segment with respect to the reference frame. We set the minimum number of cells, N_{min}, for the line segment and discard the line segments under N_{min} assuming these are either small objects or line segments that are not sufficiently identified. The orientation of each line segment itself, however, has no meaning because there can always be an angle error of the robot with respect to the reference frame. Only the relative angle between line segments can give information on the shape of the object. These two pieces of information, the relative angle and the number of line segments, can completely characterize the shape of the object in the local map.

3. Error Propagation Model of Robot's Position and Orientation

The error propagation model describes how the robot's position and orientation change with time in response to a control input and noise disturbance. Let $\underline{X}_k (=[x_k, y_k, \theta_k]^T)$ be the robot's position and orientation state vector, u_k the control input, and $\underline{w}_k$ be the noise disturbance at some time k, The general discrete form of the state vector is

$$\underline{X}_{k+1}\Phi(\underline{X}_k, \underline{u}_k)+\underline{w}_k, \quad \underline{w}_k \sim N(0, Q_k) \tag{1}$$

where $\Phi(\underline{X}_k, \underline{u}_k)$ is the nonlinear state transition function. The notation $\underline{w}_k \sim N(0, Q_k)$ indicates that this noise source is assumed to be zero-mean Gaussian with covariance Q_k (Gellb, 1973)

The model we have used is based on point kinematics (Smith, 1986). The control input $\underline{u}_k = [d_k, \Delta\theta_k]^T$ is a forward translation by a distance d_k and a rotation by an angle $\Delta\theta_k$. Then the state transition $\Phi(\underline{X}_k, \underline{u}_k)$ has the form

$$\Phi(\underline{X}_k, \underline{u}_k)=\begin{bmatrix} x_k+d_k\cos\theta_k \\ y_k+d_k\sin\theta_k \\ \theta_k+\Delta\theta_k \end{bmatrix} \tag{2}$$

The error covariance matrix P_k at time k is defined as

$$P_k=E[\underline{\tilde{X}}_k \quad \underline{\tilde{X}}_k^T] \tag{3}$$

where the error vector $\underline{\tilde{X}}_k$ is the difference between the true state $\underline{X}_k$ and the estimated one $\underline{\hat{X}}_k$ from dead reckoning, i. e., $\underline{\tilde{X}}_k=\underline{\hat{X}}_k-\underline{X}_k$. We then can write the estimated state $\underline{\hat{X}}_{k+1}$ at time k +1 such that

$$\underline{\hat{X}}_{k+1}=\Phi(\underline{\hat{X}}_k, \underline{u}_k) \tag{4}$$

By linearizing Eq. (1) about the estimated $\underline{\hat{X}}_k$ we have

$$\underline{X}_{k+1}=\Phi(\underline{\hat{X}}_k, \underline{u}_k)+J(\underline{X}_k-\underline{\hat{X}}_k)+\underline{w}_k \tag{5}$$

where J is the Jacobian of the state transition function $\Phi(\underline{\hat{X}}_k, \underline{u}_k)$ whice has the form

$$J=\begin{bmatrix} 1 & 0 & -d_k\sin\hat{\theta}_k \\ 0 & 1 & d_k\cos\hat{\theta}_k \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

Subtracting Eq. (5) from Eq. (4), we have the error vector at time k+1,

$$\underline{\tilde{X}}_{k+1}=J(\underline{\hat{X}}_k-\underline{X}_k)-\underline{w}_k \tag{7}$$

The error covariance matrix P_{k+1} at time k+1 can be found by squaring Eq. (7), and taking expectations to yield

$$\begin{aligned} P_{k+1}&=E[\underline{\tilde{X}}_{k+1} \quad \underline{\tilde{X}}_{k+1}^T] \\ &=JE[\underline{\tilde{X}}_k \quad \underline{\tilde{X}}_k^T]J^T+E[\underline{w}_k \quad \underline{w}_k^T] \\ &\quad -JE[\underline{\tilde{X}}_k \quad \underline{w}_k^T]-E[\underline{w}_k \quad \underline{\tilde{X}}_k^T]J^T \end{aligned} \tag{8}$$

Since the last two terms of the right hand side in Eq. (8) are zero because $\underline{\tilde{X}}_k$ and $\underline{w}_k$ are uncorrected ($\underline{w}_k$ is zero-mean Gaussian random variable), Eq. (8) is

$$P_{k+1}=JP_kJ^T+Q_k \tag{9}$$

By using Eq. (9) we can estimate the error range of the robot's current position from the multidimensional Gaussian probability distribution (Smith, 1986).

4. Autonomous Selection of Reference Maps

Selection of landmarks is a key issue for position estimation in an autonomous mobile robot system. Memorizing all the local maps as reference landmarks seems to be redundant and an excessive waste of time and memory. In addition, a different map can give different kinds of information according to the shape of object, and some of them cannot give sufficient information for position estimation. This section develops a method that can select and store distinctive maps among a series of local maps.

4.1 Strategies for the selection of reference maps

Among successive local maps, only the one that can give as much information on position (x, y, θ) as possible is considered to be qualified for a reference map. Furthermore, the method should be able to be applied in real time execution. Therefore, considering that our map is composed of occupancy grids, we set up the following strategy for the selection of reference maps:

1. Select a distinctive feature such as a cornered shape. A local map composed of one line segment does not provide any information on position (x, y), it only gives orientation information relative to the frame of reference. In cases where the robot operates near a long straight wall, however, the orientation information itself could be important because the heading error of the robot will result in a large position error over the traveling distance. A local map composed of one line segment is also

selected in a regular interval , D_{max}, as well as those with cornered shapes.

2. Select a map with as many occupied cells as possible. The occupied cells can provide information about the configuration of the object in a local map thus providing position information. The more occupied cells the map has, the better the matching quality becomes.
3. Configurations (shapes) of an object in a reference map should be such that it can be easily distinguished from neighboring ones. In the real world there may be many objects whose configurations are identical. This will cause the robot to be confused about which reference map it should select to match the current local map in the position estimation step.
4. Keep a certain distance D_{min} between reference maps. Considering the memory capacity and the time needed to check the correspondence between maps, it is desirable that only one reference map exist within the error bound of the robot's current position.

Items 1 and 2 may seem to be redundant because a map of a cornered shape always has more occupied cells than that of a linear shape does. In the real world, however, the number of occupied cells in a local map is largely dependent on the surface roughness; a smooth surface is seldom detected by a sensor due to multipath effects, so that the number of occupied cells from a smooth surface is much less than that from a rough one. Also, the number of occupied cells in a map can be changed according to the movement of the robot even though the shape of the map remains unchanged, and a cornered shape with two line segments generally has fewer cells than that with more than two line segments. Therefore, item 2 is necessary to select a reference map having as much position information as possible.

There should be no position error in a reference map if it to be used as a global reference. It is, however, impossible to realize in a truly autonomous system to which no a priori reference map is given. Thus we assume that the position error produced in the initial exploration step is negligible compared to that in the task execution step. The assumption is valid for small workspaces such as an indoor environment.

In a wide workspace, the error involved in the reference maps can be significant so that they cannot be used as a global reference. However, if the robot needs only its relative position to a certain object or wall, the reference maps selected under the above assumption can still provide the necessary information for robot's operation. Consequently, the use of a reference map makes it possible for the robot to operate in its workspace within the initial error range produced during the exploration step.

4.2 Selection algorithm

Using the method described in the preceding sections, the current local map is stored as a reference map according to one of the following cases:

1. $D > D_{max}$
2. $D > D_{min}$ and $N_s \geq 2$
3. If $D < D_{min}$, save the current local map according to one of the following cases and remove the nearest reference map saved previously;
   (a) N_s of current map $> N_s$ of the nearest reference map
   (b) N_c of current map $> N_c$ of the nearest reference map
   (c) N_s of the nearest reference map $= 1$ and N_s of current map ≥ 2

where,

D : distance between the centroids of the nearest reference map and current map

N_c : total number of occupied cells in a map

N_s : total number of line segments in a map

Case 1 corresponds to the robot following a long straight wall and the current local map is saved for angle correction even though its shap is linear. Case 3 is necessary to select a local map having more complex shapes than the nearest reference map in order to increase the matching quality. In this case, the previously stored reference map is replaced by the current local map.

Once a local map is selected to be saved for later position estimation, the center of the local map is transformed to the centroid of the occupied cells for the matching procedure (details of this are given in Section 5. 2). The number of line segments and the maximum angle difference between line segments are also saved as the configuration of the reference map as well as the coordinates of the centroid of the transformed map.

5. Position Estimation

Thus far we have described the method of selecting and storing reference maps in the initial exploration step. In this section, the details for position estimation based on cell-based matching are given.

5.1 Correspondence indices

Since there are many stored reference maps, finding the reference map that corresponds to a current local map should be done prior to position estimation. In cases where the position error is very small, one can assume that the closest reference map is most likely to be the corresponding one. However, a significant position error is expected here due to the frequent turns and long traveling distance of a robot while undertaking a given task. Accordingly, we define the following indices to check the correspondence between two maps:

1. Configuration index — This represents the correspondence of configurations in two maps. As stated in Section 2.2, the configuration of a local map is represented by the number of line segments and the maximum angle between them. Thus this index tells the robot whether or not the angle and the number of line segments are the same as those of a reference map.
2. Distance index — The distance between two maps can be an important clue to the correspondence between them because the characteristics of position error can be considered to be random in nature. This index allows one to determin the likely existence of a reference map within the robot's current error range as described in Section 3.
3. Cell index — This represents the correspondence between the number of occupied cells in two maps. Since the number of occupied cells in a map depends on the surface roughness and the shape of an object, this index can also provide information on the correspondence between two maps. Another important feature of this index is the fact that it tells the robot the moment that matching is possible. In other words, the quality of matching mainly depends on the number of occupied cells in two maps and hence it is important to keep more than a certain ratio, R_C, of the number of cells between two maps before matching is done.

If all the indices given above are satisfied, the robot assumes that the correspondence between two maps is satisfied, and starts to match the local map to the reference map at this position to estimate current position.

One may question the use of the configuration index because the distance and cell indices can be sufficient for correspondence if the position error is not so large. Without this index, even for the case where there is only one reference map within the possible matching range, the estimation of position can fail. For example, consider a reference map of a right angled corner while a local map represents only a part (only one line segment) of the corner in the reference map. The distance index can tell that the two maps are within the possible matching range because there can be a certain amount of position error. At the same time the cell index can also be satisfied because it checks only the ratio of the total number of occupied cells in two maps. Consequently, the robot would try to match the two maps (one is of cornered shape and the other is of linear shape), and the results may be even worse than before. In actual implementation, this can happen frequently; hence the configuration index is essential for accurate correspondence.

5.2 Matching-minimizing the discrepancy between two maps

Once the correspondence between the reference and local maps is satisfied, then position estimation is performed by minimizing the discrepancy between the two maps. We adopt a method modified slightly from Moravec's (Moravec, 1985) that matches two maps and reports the

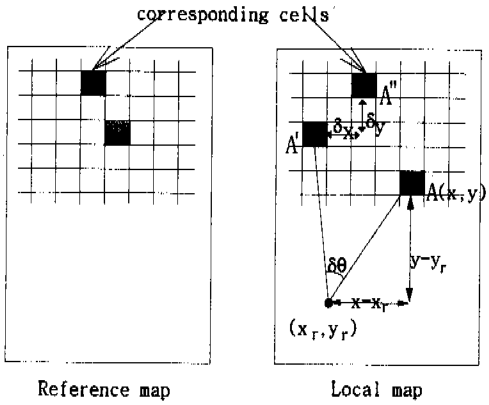


Fig. 2 Schematic diagram of maps for matching(the cell A(x, y) is rotated(A') and transformed into A").

displacement and rotation that best takes one into the other. The goodness of the match between two maps at a trial displacement (δx, δy) and rotation ($\delta\theta$) is measured by computing the sum of products of corresponding cells in the two maps. This procedure is illustrated in Fig. 2. It transforms the coordinates of each occupied cell in the local map through the following equation to find a corresponding cell in the reference map:

$$\begin{Bmatrix} X \\ Y \end{Bmatrix} = \begin{Bmatrix} \cos\delta\theta & -\sin\delta\theta \\ \sin\delta\theta & \cos\delta\theta \end{Bmatrix} \begin{Bmatrix} x-x_r \\ y-y_r \end{Bmatrix} + \begin{Bmatrix} \delta x + x_r \\ \delta y + y_r \end{Bmatrix} \tag{10}$$

where (x_r, y_r) is the current position of a robot, and (x, y) and (X, Y) are the position of an occupied cell in a local map and that in a reference map, respectively. The probabilities of cells at (x, y) and (X, Y) obtained in this way are multiplied and summed.

However, this procedure is very slow because it requires $O(n^3)$ multiplications for each occupied cell when applied to maps with linear dimension of n. Considering that the number of occupied cells in a map is usually more than n, the total cost grows as $O(n^4)$. A speed-up is achieved by generating a hierarchy of reduced resolution versions of each map. A coarser map is produced from a finer one by converting two by two subarrays of cells in the original map into single cells of the reduced one. That is, if the original array has dimension n, the first reduction is of size $n/2$, the second of $n/4$ and so on. A match found at one level can be refined at the next finer level by trying only about three values of $(\delta x, \delta y, \delta\theta)$, in the vicinity of the values found at the coarser level. This method brings the matching cost down to slightly larger than $O(n)$ (Moravec, 1985).

We found, however, that a match found at the second level of size $n/4$ is very poor when applied to a local map, which results in an incorrect final match. There are only a small amount of cells labeled as occupied in the local map, and most of them appear in a particuler quadrant of a map. Thus the reduction produces a coarser map with only two or three occupied cells that are insufficient for matching. The approach starting from the level of size $n/2$ can consume a good fraction of an hour of PC time.

Considerable saving comes from the observation that the difference between the centroids of two separately generated maps of the same area is less than a quarter of the linear size of a map. The trial displacement $(\delta x, \delta y)$ at the level of size $n/2$ can be reduced to 1/16 of the original dimension by starting in the vicinity of the centroids of two maps. With a typical n of 48, this method brings the matching time down to a few seconds. A further speed-up is achieved for a linear shaped map by trying only rotation δ_θ with respect to the centroids of two maps because it cannot give any information on displacement.

6. Experimental Results

The autonomous selection of reference maps (natural landmarks) and position estimation methods have been implemented in a real environment with our mobile robot. Figure 3 shows the photograph of the robot. 24 ultrasonic sensors (MURATA, Japan) are mounted at 15-degree angular intervals. The range accuracy of the sensor is about 0.01m. Table 1 displays the statistical properties of the position error for the estimation of error range of the robot's current position. These values were used to evaluate Qk in Section 3.

The robot was run following the walls the composed of straight lines and corners as shown

Fig. 3 Photograph of the robot.

Table 1 Statistical properties of position error.
(unit : meter, radian)

Variable	Move of 2 meter	Turn of 2 radian
Standard deviation of x	0.01229	0.001773
Standard deviation of y	0.02171	0.001709
Standard deviation of angle	0.01515	0.039650
Covariance of x and y	0.00008	0.143136
Covariance of x and angle	0.00002	0.182692
Covariance of y and angle	0.00004	0.148425
Covariance between x and y	0.29326	4028E 07
Covariance between x and angle	0.04622	1.12E 05
Covariance between y and angle	−0.23029	9.93E 06

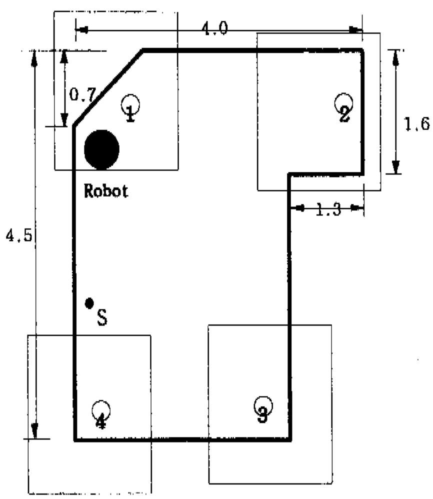
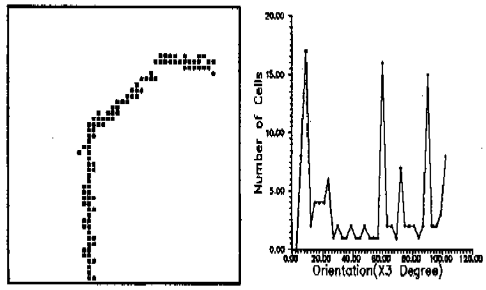
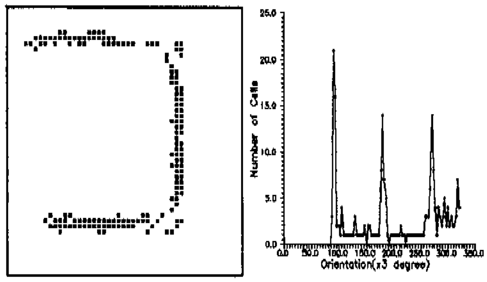


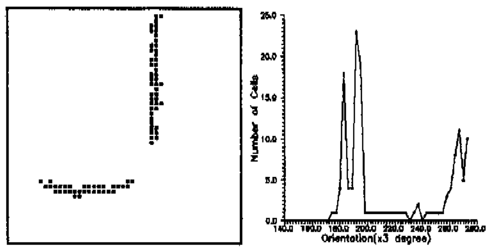
Fig. 4 Experimental environment for position estimation.



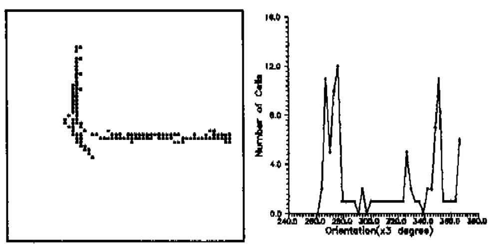
(a) Position 1



(b) Position 2



(c) Position 3



(d) Position 4

Fig. 5 The selected reference maps and distribution of orientations.

in Fig. 4. To do this, the exploration method for an unknown environment(Lim, 1998) was used. Sensor data from the robot were processed on an IBM compatible 80586 PC. At each sampling time (0.1 seconds) the robot executes the following: one translation of a local map, acquisition of 24 range returns, updating the local map, generation of a steering command to follow the wall, clustering the orientations, and position estimation.

When the algorithm to select reference maps was implemented to the environment, the local map at position 1, 2, 3 and 4 in Fig. 4 were selected and memorized as reference maps. The program parameters used in this paper were $N_{min}=7$, $D_{max}=5.0$m, $D_{min}=1.8$m and $R_c=0.7$. N_{min} and R_c were determined through sets of matching experiments. Figure 5 show the selected maps to be used as landmarks, and the distribution of the orientations of cells in each map. The results of the orientation classification are tabulated in Table 2. One can see from the table that the difference between the estimated and true angles is less then half the aperture of the beam ($w/2$). Considering that the effective width of the beam is 30°, the clustering method is accurate enough to use in identifying the shapes of object in a map.

We moved the robot randomly in the environment until the position and orientation error grew much larger than those produced in the exploration step (reference map selection step). Then the robot was run again following the walls using the

Table 2 The results of orientation classification.

Position	Object shape (the number of line segments)	True angles between line segments (degree)	Estimated angles between line segments (degree)	estimation error (degree)
1	3	45,90	45,96	0,6
2	3	90,180	83,191	−7,11
3	2	90	87	3
4	2	90	73	−17

Table 3 The estimated errors for each position. (unit : meter, radian)

Position		True position	Erroneous position	Corrected position	Error after correction
1	X	0.555	−0.388	0.494	−0.061
	Y	1.990	2.256	2.092	0.102
	Rad	5.068	5.350	5.050	0.018
2	X	3.427	2.441	3.333	−0.097
	Y	1.738	2.813	1.831	0.093
	Rad	3.313	3.595	3.345	−0.032
3	X	1.954	1.825	1.853	−0.101
	Y	−1.134	−0.355	−1.281	−0.147
	Rad	2.757	3.039	2.739	0.018
4	X	0.631	0.616	0.640	0.009
	Y	−1.356	−0.937	−1.442	−0.086
	Rad	1.529	1.811	1.461	−0.068

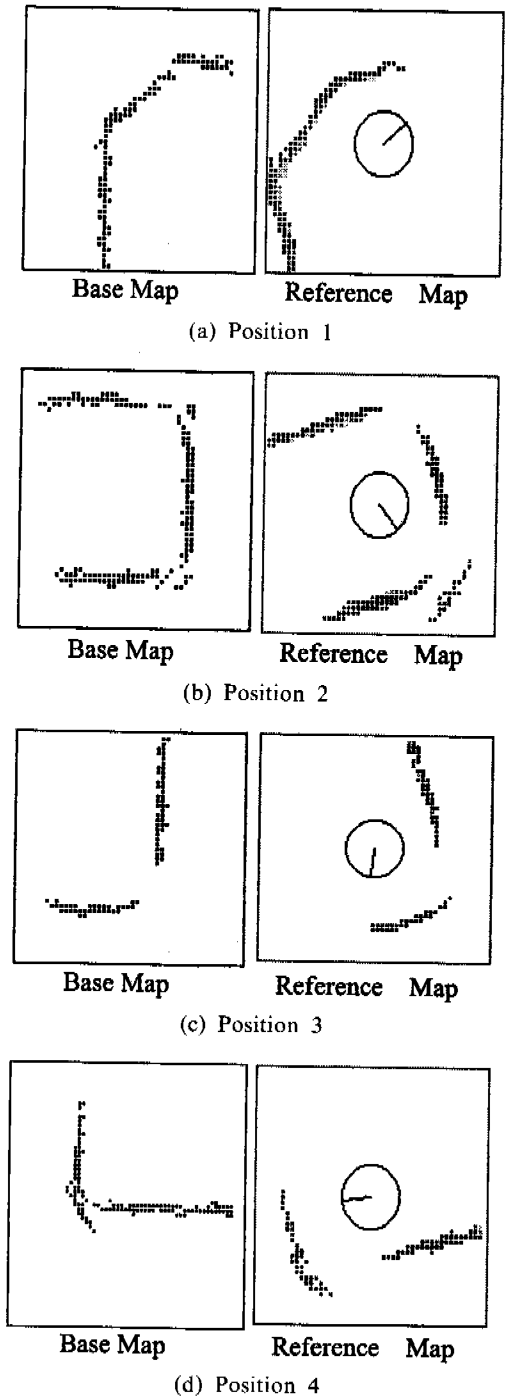


Fig. 6 The local and corresponding base maps for matching.

same algorithms as in the exploration step to test the position estimation algorithms. The results are shown in Fig. 6. The reference map in each figure is the one recalled by the robot itself through checking the correspondence. Table 3 tabulates the estimated error for each position along the path. The maximum estimation errors were 0. 147m and 0.068 rad for position and heading angle respectively. These results are good enough for practical use and surprisingly accurate considering the wide beam aperture of a sonar sensor and the linear dimension of a cell (0.05m).

The run time for reference map selection or correspondence check was negligible, so that the cycle time including local map construction and clustering the orientation was less than 0.1 seconds. The time required for matching is dependent on the number of occupied cells in two maps. The maximum time was 0.16 seconds for Fig. 6 (b) which has more occupied cells than others, and the average was 0.077 seconds.

7. Conclusions

This paper has developed a system of position estimation for robot navigation. The system is composed of classification of an object's configuration in a map, autonomous landmark selection, estimation of correspondence, and cell-based matching between maps using the local map.

The classification of object configuration is based on orientations of cells in a local map. These are clustered into several groups to extract the line segments in the map. This information on line segments is then used to classify the configuration of objects in the map for the selection of reference maps and estimation of correspondence between maps. In autonomous landmark selection, the configuration of an object in a map and the distance between reference maps were taken into consideration for selecting distinctive maps among successively constructed local maps. Similar concepts were also used for checking the correspondence between the stored reference map and the current local map. The position of the robot is then estimated through a centroid-based matching procedure. The usefulness of all these approaches was illustrated with the results produced by a real robot equipped with ultrasonic sensors operating in a real world enviranment.

Acknowledgment

This work has been supported by the Korea Science and Engineering Foundation (No. 951-1008-025-2), 1995. 3. 1～1997. 2. 28.

References

Chatila, R. and Laumond, J., 1992, "Position Referencing and Consistent World Modeling for Mobile Robots," *IEEE International Conference on Robotics and Automation*, St. Louis, MO, USA, pp. 138～145.

Crowley, J., 1986, "Representation and Maintenance of a Composite Surface Model," *IEEE International Conference on Robotics and Automation*, San Francisco, CA, USA, pp. 1455～1462.

Crowley, J., 1992, "World Modeling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging," *IEEE International Conference on Robotics and Automation*, USA, pp. 674～681.

Elfes, A., 1987, "Sonar-Based Real-World Mapping and Navigation," *IEEE Transaction on Robotics and Automation*, Vol. RA-3, No. 3, pp. 249～265.

Gellb, A. C., 1973 *Applied Optimal Estimation*, The MIT Press.

Gonzalez, J., Stentz, A. and Ollero, A., 1992, "An Iconic Position Estimator for a 2D Laser Range Finder," *IEEE International Conference on Robotics and Automation*, Nice, France, pp. 2646～2651.

Herbert, M., Kanade, T. and Kweon, I., 1988, "3-D Vision Techniques for Autonomous Vehicles," Tech. Rep. CMU-RI-TR-88-12.

Hyppa, K., 1989, "Lulea Turbo Turtle (LTT)," *IEEE/IROS, '89* Tsukuba, Japan, pp. 620～623.

Kleeman, L., 1989, "Ultrasonic Autonomous Robot Localization System," *IEEE/IROS*, '89

Tsukuba, Japan, pp. 212~219.

Kleeman, L., 1992, "Optimal Estimation of Position and Heading for Mobile Robots Using Ultrasonic Beacons and Dead-Reckoning," *IEEE International Conference on Robotics and Automation*, Nice, France, pp. 2582~2587.

Leonard, J. and Durrent-Whyte, H., 1992, "Mobile Robot Localization by Tracking Geometric Beacons," *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 3, pp. 367~382.

Lim, J. H. and Cho, D. W., 1992 , "Physically Based Sensor Modeling for a Sonar Map in a Specular Environment," *IEEE International Conference on Robotics and Automation*, Nice, France, pp. 1714~1719

Lim, J. H. and Cho, D. W., 1994a, "Specular Reflection Probability in the Certainty Grid Representation," *Transactions of ASME, Journal of Dynamic Systems, Measurements and Control* 116, No. 3, pp. 512~520.

Lim, J. H. and Cho, D. W., 1994b, "Real Time Map Construction and Position Estimation for an Autonomous Mobile using Sonar Sensors," *ISCA Int. Jour. of Computers and Their Applications*, Vol. 1, No. 2, pp. 111~124.

Lim, J. H. and Cho, D. W., 1996, "Multipath Bayesian Map Construction Model from Sonar Data," *ROBOTICA* 14, pp. 527~540.

Lim, J. H. and Cho, D. W., 1998, "Sonar based Systematic Exploration Method for an Autonomous Mobile Robot Operating in an Unknown Environment," To appear *ROBOTICA*, 1998. 11. (accepted and in press).

Moravec, H. P. and Elfes, A., 1985, "High Resolution Maps from Wide Angle Sonar," *IEEE International Conference on Robotics and Automation*, St. Louis, pp. 116~121.

Smith, R. C. and Cheesman, P., 1986 , " On the Representation and Estimation of Spatial Uncertainty," *The International Journal of Robotics Research*, Vol. 5, No. 4, pp. 56~68.