An Efficient Representation of Edge Shapes in Topological Maps

Nakju Lett Doh and Wan Kyun Chung

There are nodes and edges in a topological map. Node data has been used as a main source of information for the localization of mobile robots. In contrast, edge data is regarded as a minor source of information, and it has been used in an intuitive and heuristic way. However, edge data also can be used as a good source of information and provide a way to use edge data efficiently. For that purpose, we define a data format which describes the shape of an edge. This format is called local generalized Voronoi graph's angle (LGA). However, the LGA is constituted of too many samples; therefore, real time localization cannot be performed. To reduce the number of samples, we propose a compression method which utilizes wavelet transformation. This method abstracts the LGA by key factors using far fewer samples than the LGA. Experiments show that the LGA accurately describes the shape of the edges and that the key factors preserve most information of the LGA while reducing the number of samples.

Keywords: Localization, topological map, mobile robots, wavelet transformation, dynamic time warping.

I. Introduction

The interior of a large well-structured building, such as a hospital, consists of many corridors and junctions. The topological map which is an abstraction of an environment is a good tool for the representation of large buildings as shown in Fig. 1. This map is made up of nodes and edges. Here, nodes represent topologically meaningful places, such as junctions. The edges describe corridors that connect two nodes.

In the topological map, the nodes have distinctive features. Researchers have developed many localization algorithms [1]-[5] that utilize the node information. In contrast, edge data is not rich in comparison to node information. For this reason, there have been relatively few studies that utilize edge information. In the vision research community, significant results [6]-[16] concerning edge information have been reported. However, the foci of these reports are on robust edge detection, an edge matching technique after a shift or rotation of an image, edge data learning, and so on. These topics do not have a meaningful relationship in utilizing edges for robot localization because edge data in localization represents corridor-like environments, while that of the edge in vision research is a series of distinct points in images.

In robotics research, Nagatani and others [17] used edge information for localization in a topological map. They used the length, shape, and area of an edge. Similarly, Kruusmaa [18] utilized the shape of the edge as an aid for navigation in a dynamic environment. Kuipers and others [19] also utilized edge information, but they used it only to acquire the number and the direction of emanating edges.

In a similar approach, we propose an efficient representation method for edge shapes. For that purpose, we consider three subjects. First, we define a data format for edge representation.

Manuscript received Jan. 04, 2007; revised Apr. 18, 2007.

This work was supported by the Korean Institute of Construction & Transportation Technology Evaluation and Planning (KICTEP) and the program number is '06-Unified and Advanced Construction Technology Program-D01'.

Nakju Lett Doh (phone: + 82 2 3290 4822, email: nakju@korea.ac.kr) is with the School of Electrical Engineering, Korea University, Seoul, Rep. of Korea.

Wan Kyun Chung (email: wkchung@postech.ac.kr) is with the Department of Engineering, POSTECH, Pohang, Rep. of Korea.

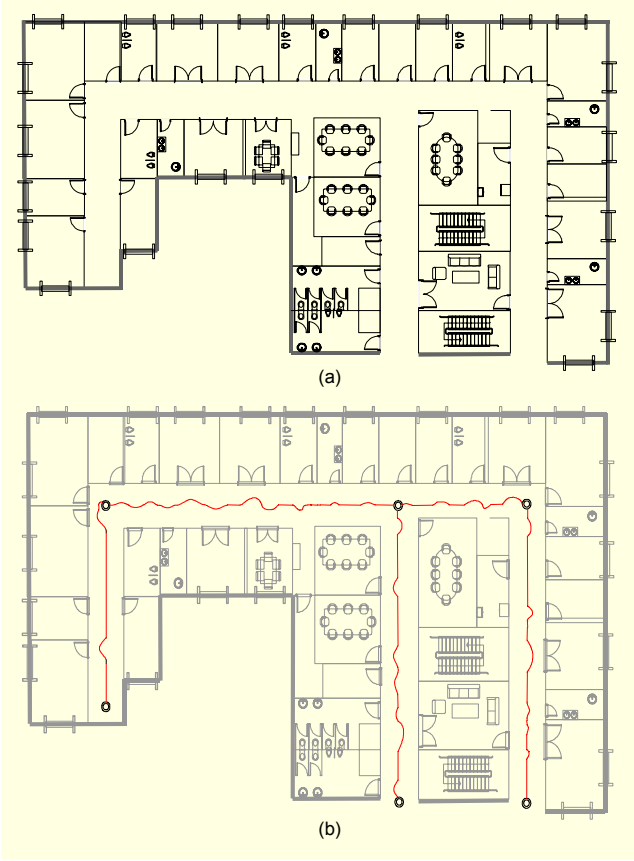

Fig. 1. Maps of the interior of a large building: (a) CAD map and (b) a corresponding topological map. In our approach, we use a hybrid map where only hallways are represented by the topological map. Other spaces such as rooms are modeled by local metric maps to describe the layout of the area in detail.

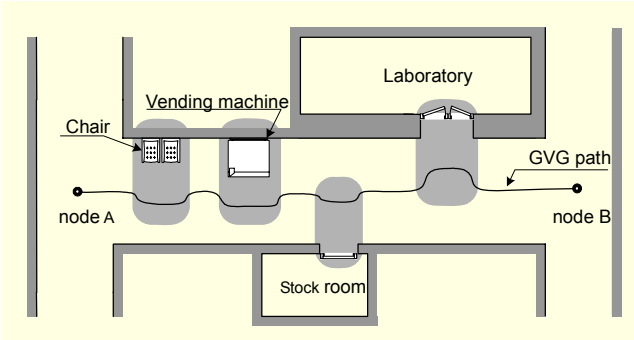

Fig. 2. Example of a GVG path from node A to node B along a corridor. Note that the objects on the side walls (the chair and vending machine) change the shape of the GVG.

In general, a corridor which corresponds to an edge in a topological map can be represented by a 2-dimensional (2D) grid map which is an accumulation of sensor data around a robot. However, it is not efficient to use a 2D grid map for localization. There are many edges in a large building and the use of a 2D grid map itself causes a heavy computational burden.

In this paper, we suggest a method to describe the edge as a series of points (one-dimensional). This method uses the generalized Voronoi graph (GVG) [20]. The nodes of this graph correspond to points that are equidistant to more than three objects. The edges consist of points that are equidistant to two obstacles (see Fig. 2). The GVG can be extracted from sensor data [21] and corresponds to a path through the center of a corridor. The GVG contains meaningful information about the outlines of side walls. For example, the contours of a chair, a vending machine, hollow spaces in front of a stock room, and a laboratory are expressed in the GVG (see Fig. 2). We slightly modify the GVG by differentiation and coordinate transformation. We call this modified GVG a local GVG's angle (LGA).

We also propose an efficient compression method which enables real time calculation for the LGA. For that purpose, we suggest key factors which describe the characteristics of the LGA in a condensed form. For the extraction of key factors, we adopt wavelet transformation [22]-[24], which is one of the most efficient compression tools.

In addition, we present a comparison method for two edge shapes, the results of which can be used for the localization. In the comparison, however, there is a problem because the numbers of samples from the two shapes are different. To alleviate this problem, the dynamic time warping algorithm [25], [26], which is fairly effective and widely used as a tool for comparison of two different time series, is adopted.

Two kinds of experiments were performed. First, a performance analysis was conducted for a total of 30 edges with obstacles within them. In this experiment, we demonstrated that the LGA accurately describes the shape of edges. However, it took 516.9 seconds for the calculation because of the large number of samples in the LGA. The key factors reduced the number of samples by 96.1% while preserving more than 95% of the information of the LGA. The comparison with the key factors was accomplished within a second. However, the discrimination performance was degraded by 21.4%. Second, the proposed method was tested in a real environment within the hierarchical atlas framework [27]. This experiment demonstrated that the additional use of the LGA increases the performance by 1.7 times and that the key factor is a very efficient compression method for the LGA.

This paper is organized as follows. The LGA is defined in section II. A compression method using the wavelet transformation is given in section III. A method for comparison between two edges is explained in section IV. Two sets of experiments are presented in section V and section VI,

respectively and the conclusion follows.

II. Edge Representation

The edge data consists of the length, area, and shape of an edge. Note that the length, area, and shape of an edge correspond to those of a corridor in the real environment. These three parts can be assumed to be independent. With this assumption, we define an edge probability (P_{edge}) as the probability of two edge representations belonging to the same edge. It can be written as

$$P_{edge}(E_\alpha, E_\beta) = P_{length}(L_\alpha, L_\beta) \times P_{area}(A_\alpha, A_\beta) \times P_{shape}(S_\alpha, S_\beta), \quad (1)$$

where $P_{length}(a, b)$, $P_{area}(a, b)$, and $P_{shape}(a, b)$ are probabilities that the length, the area, and the shape of a and b will be the same. The variables L, A, and S denote the length, area, and shape of an edge E, respectively.

Here, P_{length} and P_{area} are simply calculated [17] by the differences in length and area for two edges as

$$P_{length}(L_\alpha, L_B) = \boldsymbol{N}\left(\frac{1}{|L_\alpha - L_B|}\right), \quad (2)$$

$$P_{area}(A_\alpha, A_B) = \boldsymbol{N}\left(\frac{1}{|A_\alpha - A_B|}\right), \quad (3)$$

where $\boldsymbol{N}(\cdot)$ is a normalization function.

In contrast, P_{shape} cannot be acquired by simple algebraic calculations. In our studies, we focus on the shape and suggest a systematic and efficient way to calculate P_{shape}.

For that purpose, a data format for the representation of the shape needs to be defined. The general form of an edge is a 2D grid map. It is not efficient to use the grid map itself. One good choice is the GVG [20] which contains the unique features of an edge (see Fig. 2). This graph can be acquired by storing odometry data of a robot that moves along the GVG.

We modified the GVG for the following reasons. First, the GVG is a set of points which is also 2-dimensional data (position x and y). To reduce the dimensions, we use the tangential angles of the GVG. If an angle is acquired for the same time intervals, the shape of the GVG can be fully recovered because all information of the shape is contained in the angle. Second, the angle should not be described with respect to a global coordinate system. Let us assume that a robot started at point S and reached node α (N_α) as in Fig. 3(a). Also, let us assume that the starting angle of the GVG from N to N is 90° with respect to the global coordinate system. Here, the real angle should be 90° at the start node N. However, the

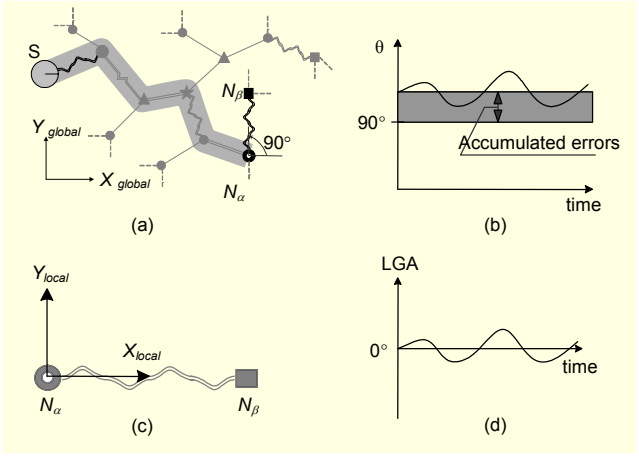

Fig. 3. Illustration of the LGA: (a) A robot had navigated from S to node N= through the shaded path. (b) The angle of the GVG (denoted by θ) is affected by accumulated errors. Thus, the angle is represented with respect to (c) a local coordinate system whose direction of the x-coordinate is parallel to the heading angle of the robot at the start node, and this angle is called (d) an LGA.

robot does not record it as 90° because of accumulated errors during its travel from S to N (see Fig. 3(b)). Thus, it is good to describe the angle of the GVG not with respect to the global coordinate system, but with respect to a local coordinate system (see Fig. 3(c)) whose direction of the x-coordinate is parallel to the heading angle of the robot at the start node.

Therefore, we describe an edge by using a set of angles of a GVG with respect to the local coordinate system and call it the LGA. Note that the initial value of the LGA is always zero as shown in Fig. 3(d).

III. Edge Data Compression

In the previous section, we proposed the LGA for the expression of the shape of edges. The LGA, however, may contain a large number of samples in itself because, in general, the GVG data is recorded for every control step during the movement of a robot. For example, if a robot is controlled with a frequency of 100 Hz and with a speed of 0.25 m/sec, an edge with a length of 10 m is described by an LGA with 4,000 samples.

With the large number of samples, the calculation of the edge probability cannot be performed in real time. Thus, we suggest an LGA compression method which adopts wavelet transformation [22], [23], a powerful condensation tool. The wavelet transformation decomposes an original signal into two parts, a high frequency part and a low frequency part. This procedure is repeatedly applied to an original signal ($A^{[0]}$), and a hierarchical decomposition, as in (4) and Fig. 4, is generated.

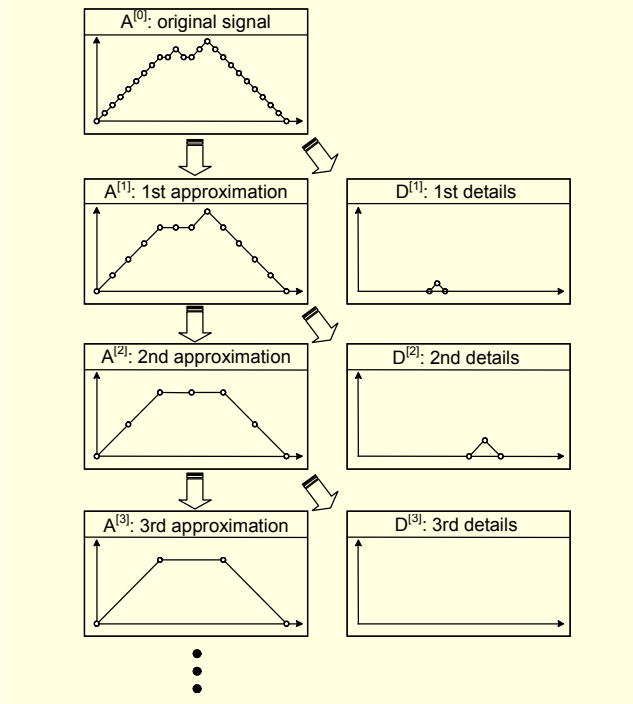

Fig. 4. Hierarchical decomposition of the wavelet transformation. An original signal ($A^{[0]}$) is decomposed into its low ($A^{[1]}$) and high ($D^{[1]}$) frequency part.

$$\begin{aligned} A^{[0]} &= A^{[1]} + D^{[1]} \\ &= A^{[2]} + D^{[2]} + +D^{[1]} \\ &= A^{[3]} + D^{[3]} + D^{[2]} + +D^{[1]} \\ &\vdots \end{aligned} \tag{4}$$

Here, $A^{[\alpha]}$ is the low frequency part of $A^{[\alpha-1]}$ (the approximation of level α) and $D^{[\alpha]}$ is the high frequency part of $A^{[\alpha-1]}$ (the details of level α).

In a conventional compression, the coefficients of the details are cut off by a global threshold or by the balance sparsity norm [28]. This conventional approach, however, cannot be used in our case because this method needs a decoding procedure, namely, reconstructing the original signal using the approximations and details. This decoding procedure adds another calculation burden.

In this paper, we propose a method that uses the wavelet transformation without the decoding procedure. Before going further, let us define a retained energy (Er) [28] for the approximation of level α ($A^{[\alpha]}$) with respect to the original signal ($A^{[0]}$). It denotes for the remaining energy after an approximation and is expressed by a ratio of summations as

$$E_r(\alpha) = 100 \times \frac{\sum_{i=1}^{\#(A^{[\alpha]})} \left| A_i^{[\alpha]} \right|^2}{\sum_{i=1}^{\#(A^{[0]})} \left| A_i^{[0]} \right|^2}, \tag{5}$$

where $\#(A)$ is the number of samples in A, and A_i is an i-th element in A.

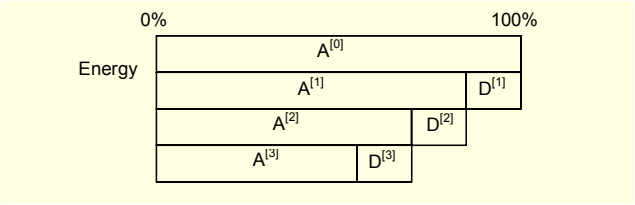

Fig. 5. Retained energy of $A^{[\alpha]}$ is decreased as the level of decomposition (α) increases. The amount of energy distribution is exaggerated in this figure for the sake of readability.

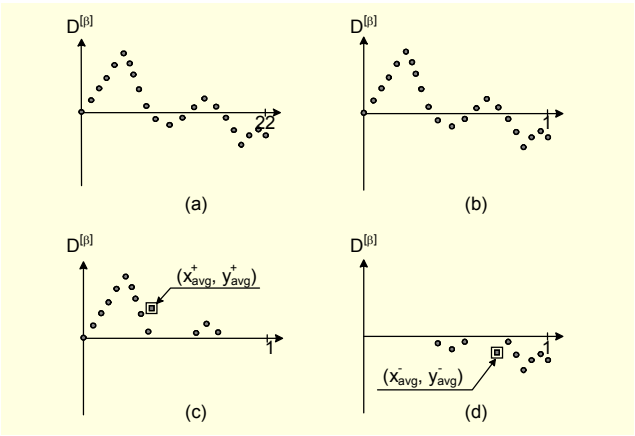

Fig. 6. Procedures of extract micro key factors: (a) all values of $D^{[\beta]}$ are represented as points and (b) we normalize them along the x-direction. Then, we calculate (c) the positive and (d) the negative micro key factors by averaging them.

The retained energy of $A^{[\alpha]}$ decreases as the level of decomposition (α) increases as shown in Fig. 5. If we select $A^{[\alpha]}$ whose retained energy is larger than 95%, we can preserve most of the information of the original signal while reducing the number of samples by 2^{α} times. To reduce as many samples as possible, it is good to take the largest α (denote it as α*) whose approximation of that level ($A^{[\alpha*]}$) contains more than an energy of 95%. Here, $A^{[\alpha*]}$ represents the overall shape of an edge called the *macro key factors*.

Now, an energy less than 5% is contained in $D^{[1]}, \cdots, D^{[\alpha*]}$ (denoted as $D^{[1:\alpha*]}$). There are many samples in $D^{[1:\alpha*]}$ even though the contained energy is less than 5%. Therefore, it is efficient to abstract $D^{[1:\alpha*]}$ in a simple form. For that purpose, we define the *micro key factors* whose extraction procedures are as follows:

Step 1. Select a level β between 1 and α*.
Step 2. Represent all values of $D^{[\beta]}$ as points (see Fig. 6(a)).
Step 3. Normalize the values along the x-coordinate with respect to the largest one (see Fig. 6(b)). This normalization eliminates the effect of the edge length.

Step 4. Calculate the positive average $(x^{+}_{avg}, y^{+}_{avg})$ for points whose y values are larger than 0 (see Fig. 6(c)).

Step 5. Calculate the negative average $(x^{-}_{avg}, y^{-}_{avg})$ for points whose y values are smaller than 0 (see Fig. 6(d)).

Step 6. Repeat steps 1 to 5 for all βs between 1 and α^*.

Physically, the micro key factors are the averages for positive and negative values of $D^{[\beta]}$. The magnitude, the direction, and the location of details are implicitly contained in the micro key factors with an intensive abstraction. For example, Fig. 7 shows that the micro key factors indicate the differences in magnitude, direction, and location of the details. In summary, we have defined the macro and the micro key factors to compress the LGA. One LGA is expressed by macro and micro key factors. The macro key factors are sample values in an approximation of the level $\alpha^*(A^{[\alpha^*]})$. These macro key factors reveal the overall shape of the original signal. The number of samples of the macro key factors is $\#(A^{[0]})/2^{\alpha^*}$.

The micro key factors abstract the magnitude, direction, and location of the details of $D^{[1:\alpha^*]}$. The number of samples in the micro key factors is $4\times\alpha^*$. Therefore, the total number of samples in the key factors is given as

$$\frac{\#(A^{[0]})}{2^{\alpha^*}} + 4\alpha^* \approx \frac{\#(A^{[0]})}{2^{\alpha^*}}. \quad (6)$$

Thus, the number of samples in $A^{[0]}$ is condensed by approximately 2^{α^*} times.

Note that this method does not hold if the retained energy of $A^{[1]}$ is less than 95% because we cannot define the macro key factor. However, a time series from the navigation of mobile robots contains many high frequency parts with low energy, and thus, in general, α^* is set as a value larger than 4.

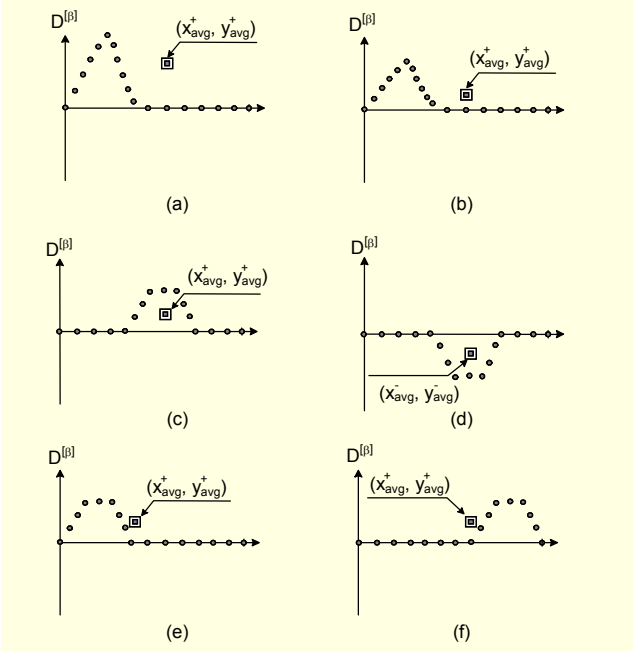

Fig. 7. Abstraction of micro key factors: (a), (b) the magnitude, (c), (d) direction, and (e), (f) location of details.

IV. The Edge Comparison Method

Both the LGA and the key factors are one-dimensional time series. There are two problems in comparing these sets of data. First, the sample numbers of the time series are different. This means that a direct comparison in a one-to-one manner cannot be performed. Second, there are nonlinear distortions in the time series. For example, if a robot moves through the same edge twice, the robot records two slightly different sets of data because of sensor noise, controller sensitivity, and so on (see Fig. 8).

The dynamic time warping (DTW) algorithm [25], [26] is fairly effective in alleviating these problems. The DTW algorithm searches for a path which gives rise to the lowest distortion between two time series while minimizing the amount of computation (see Fig. 9). The overall distortion is acquired by the DTW algorithm. It can be used for the

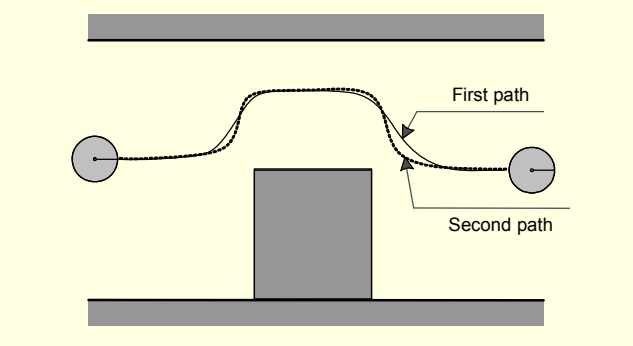

Fig. 8. Robot moves through the same edge twice but records slightly different data because of sensor noise, controller sensitivity, and so on.

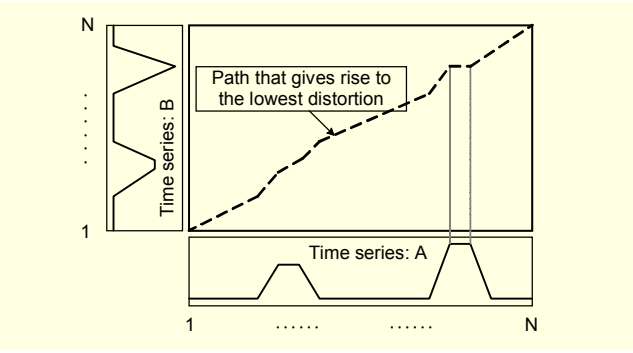

Fig. 9. Path that gives rise to the lowest distortion between two time series whose numbers of samples are N and M for $N > M$.

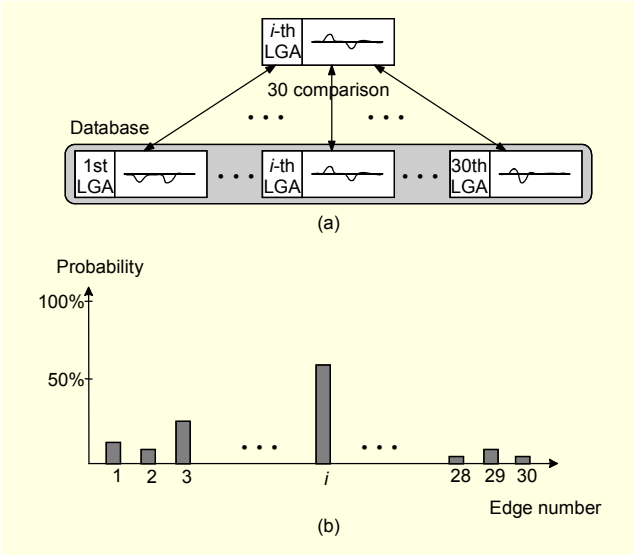

Fig. 10. For the calculation of the probability, (a) an LGA of the *i*-th edge is compared to the LGAs in the database in a one-to-one manner, and (b) this comparison yields a set of probabilities which is called "a probability class."

calculation of the probability from the shape of edges as follows. First, a robot navigates all edges in the topological map to make an LGA database. Then, the robot moves again through an edge, and an LGA is extracted. This LGA is compared to the LGAs in the database in a one-to-one manner using the DTW (see Fig. 10(a)). This comparison yields a set of probabilities called a probability class (see Fig. 10(b)).

The probability for a candidate LGA (LGA_{can}) to be the *i*-th LGA in the database (LGA_i) is

$$P_i(LGA_{can} = LGA_i) = N\left(\frac{1}{DTW(LGW_{can}, LGA_i)}\right), \quad (7)$$

where $DTW(\cdot,\cdot)$ denotes the overall distortion calculated by the DTW.

Unfortunately, (7) cannot be used for the key factors. Recall that the data size of the macro key factors is different while those of the micro key factors are not. Thus, the DTW algorithm and a direct comparison are used for the macro and the micro key factors, respectively. The probability equation of the candidate macro and micro key factors, (Ma_{can}, Mi_{can}), to be the *j*-th macro and micro key factors in the database, (Ma_j, Mi_j), is

$$P_j((Ma_{can}, Mi_{can})=(Ma_j, Mi_j)) = N\left(\frac{0.95}{DTW(Ma_{can}, Ma_j)} + \frac{0.95}{\left\|Mi_{can} - Mi_j\right\|_1^{\alpha^*}}\right), \quad (8)$$

where $\|\cdot\|_1^{\alpha^*}$ denotes the sum of absolute differences of the micro key factors from level 1 to α^*. Note that we scale the values from the macro and the micro key factors by 0.95 and 0.05 because an energy of more than 95% is contained in the macro key factors.

V. Analysis Using Varying Edge Data

Experiments were performed to validate the performance of our proposed algorithm. As mentioned before, our method only considers the probability from the shape of edges. Therefore we selected edges with the same length and the same area in our experiments. For the comparison of edges with varying lengths (or areas), the difference of the lengths (or areas) can be used to get the probability of the length (or area) in (1).

There are two objectives in this experiment. The first objective is to identify whether the LGA describes the shape of edges well or not. The second objective is to observe whether the key factors effectively compresses the LGA while maintaining the information of the shape.

1. The Performance of the LGA

A. Procedures of the Experiments

As shown in Fig. 11(a), 30 edges with various obstacles were used in this test. The locations and sizes of obstacles were set to be different to provide a unique feature to an edge. A robot with two laser scanners (see Fig. 12) moved along the GVG and generated initial maps shown in Fig. 11(b). The LGAs, extracted from the GVGs, were stored in a database.

Then, the robot moved again through the *i*-th edge, and the

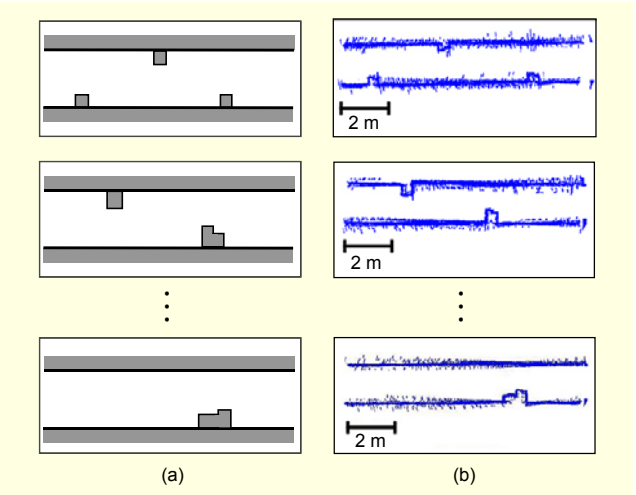

Fig. 11. Edges with various obstacles: (a) schematic diagrams of 30 edges and (b) real maps.

Fig. 12. Mobile robot used for the experiments. This robot has a differential drive type wheel base and is equipped with two laser scanners.

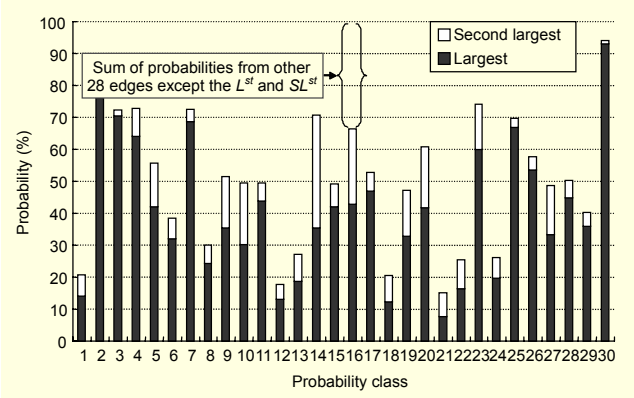

Fig. 13. Largest and second largest probabilities in the probability classes from all LGAs.

i-th LGA was extracted. A probability class for the *i*-th edge was calculated with the *i*-th LGA and the database using (7). These procedures were repeated for all $i = 1, \cdots, 30$. A total of 30 probability classes were acquired.

B. Largest and Second Largest Probability

We selected the largest (L^{st}) and the second largest (SL^{st}) values from each probability class (see Fig. 13). The average of the sum of the L^{st} and the SL^{st} was 50.2%. The other 28 probabilities shared the remaining 49.8% with an average of 1.66% (=49.8/28). Therefore, we only considered the L^{st} and the SL^{st}. Here, we define a decision ratio as a value of the L^{st} over the SL^{st} which can be used to determine whether a probability class has one distinctive value. Intuitively, if a decision ratio is larger than 1.5 (that is, if the L^{st} is higher than the SL^{st} by 1.5 times), we can say that the probability class has one distinctive value.

In our experiments, a total of 28 probability classes showed

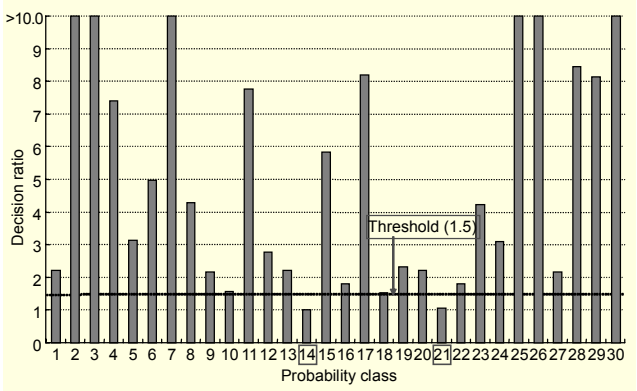

Fig. 14. Decision ratios of the probability classes from the LGAs.

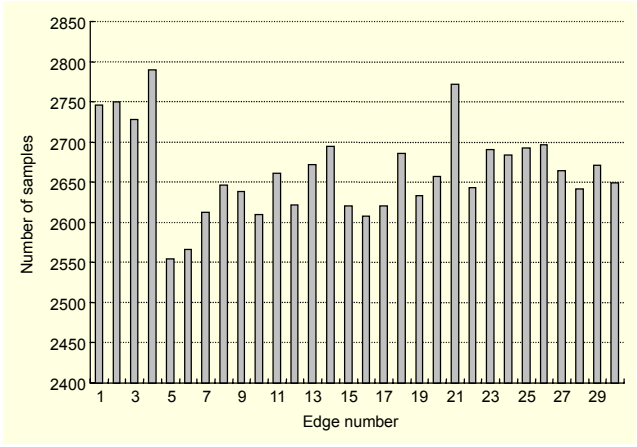

Fig. 15. Number of samples in an LGA.

decision ratios of more than a value of 1.5 (see Fig. 14). The L^{st} values of these classes were acquired from a comparison between a candidate edge E_α and the E_α in the database. The average of the decision ratios was 9.54 which can be interpreted as being very distinctive.

C. Data Size and Calculation Time

For precise navigation along the GVG, the robot was controlled with a control frequency of 50 Hz and with a speed of 0.2 m/sec. The GVG path was recorded for every control step. The average number of samples for an edge is 2,664 (see Fig. 15).

The average time in acquiring one probability class (that is, 30 comparisons using the DTW algorithm) was 516.9 seconds in a computer with 3.2 GHz CPU and 1.0 GB memory (see Fig. 16). It took a long time because of the algorithm complexity of $O(N^2)$ of the DTW. Therefore, the LGA needs to be compressed. The following subsection addresses this problem.

2. The Performance of the Key Factors

For real time implementation, the macro and micro key

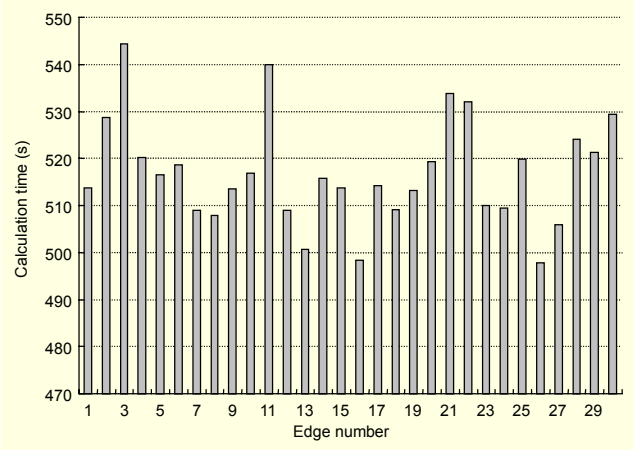

Fig. 16. Calculation times with key factors.

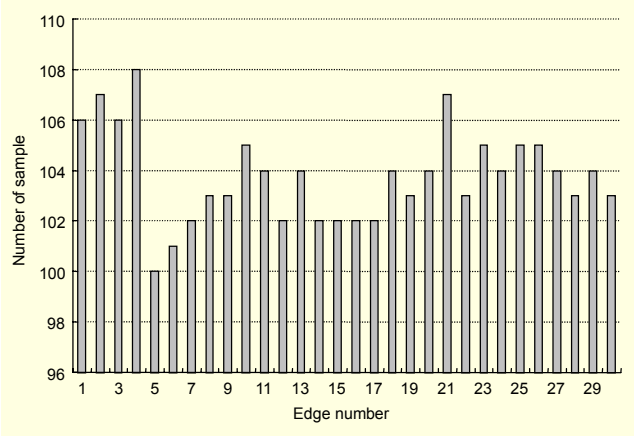

Fig. 17. Number of samples in key factors.

factors are used as condensed forms of the LGAs. Naturally, we expect that there may be advantages in the calculation time and memory requirement, as well as disadvantages in the performance of edge discrimination. These merits and demerits can be addressed in terms of the following questions.

1. How much memory is saved?
2. How much is the calculation time reduced?
3. How are the L^{st} and SL^{st} changed?
4. How are the decision values varied?

A. Reduction in Memory Requirement

The macro and micro key factors are acquired for α^* =5. The average numbers of samples of the macro and micro key factors are 84 and 20 (see Fig. 17) while that of the LGA is 2,664. Thus, the memory requirement is reduced by 96.1%.

B. Reduction in the Calculation Time

With the LGA, it takes 516.9 seconds to calculate a probability class while the same calculation takes only 0.69

Table 1. Probabilities from the LGA and from the key factor.

Probability	From the LGA (%)		From the key factor (%)	
	L^{st}	SL^{st}	L^{st}	SL^{st}
Maximum	93.05	35.49	44.56	23.59
Average	40.62	9.54	23.77	9.19
Minimum	7.68	1.06	9.30	3.55

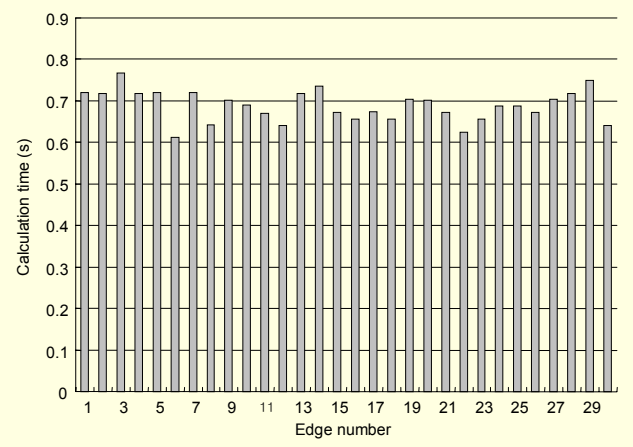

Fig. 18. Calculation times with key factors.

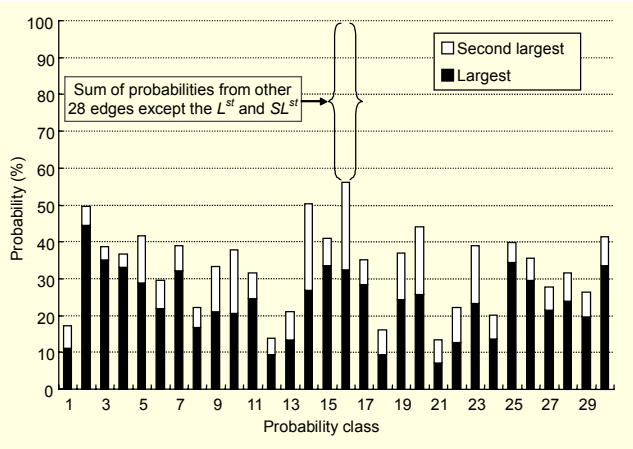

Fig. 19. Largest and the second largest probabilities in the probability classes from all key factors.

seconds with the key factors (see Fig. 18). The reason is that the DTW uses 2,664 samples of the LGA, while it utilizes only 84 samples of the macro key factors.

The calculation time is reduced by 99.9%, which is the same as the reduction ratio of the square of samples ($1\text{-}84^2/2664^2$), because the algorithm complexity is $O(N^2)$.

C. Probabilities for the L^{st} and SL^{st}

Figure 19 shows the L^{st} and SL^{st} of the probability classes

from the key factors. Table 1 compares the L^{st} and SL^{st} values from the LGA and from the key factors. The average of the L^{st} from the key factors is reduced by 41.5%, while that of the SL^{st} is decreased by 3.7%. The decreased portion of the L^{st} is distributed to all other edges by a similar quantity. For example, the probability classes from the LGA and from the key factor for edge 28 (whose L^{st} and SL^{st} are close to the averages) are plotted in Fig. 20. Here, we identify that the decreased portion of the L^{st} is distributed to the other 29 edges. In other words, the compression blurs the probabilities; thus, it decreases the highest one and increases the others.

This phenomenon is observed for all other probability classes. Thus, it does not have a serious effect on selecting the L^{st}. Therefore, even though the L^{st} values are decreased, all of them are acquired from the correctly matched edge.

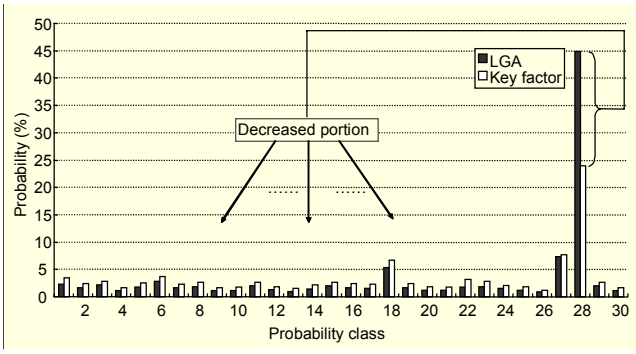

Fig. 20. Probability classes from the LGA and from the key factor for edge 28.

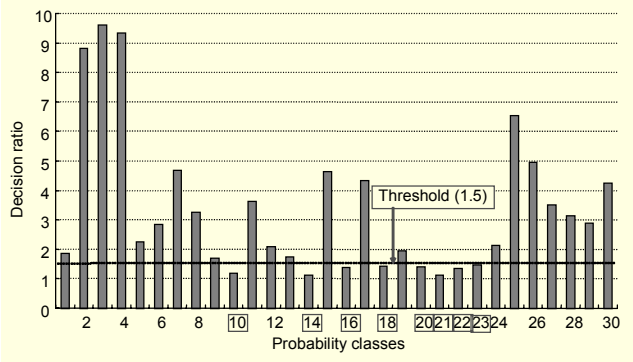

Fig. 21. Decision ratios of the probability classes from the key factors.

Table 2. Decision ratios from the LGA and from the key factors.

Decision ratio	From the LGA	From the key factor
Maximum	87.78	9.61
Average	9.54	3.35
Minimum	1.00	1.13

D. Decision Ratio

Because compression blurs the probabilities, it is natural that the decision ratios are decreased. The ratios from the key factors are shown in Fig. 21. Table 2 shows the ratio values from both the LGA and the key factors.

As shown in Table 2, the average of the decision ratios from the LGA is 9.54, while that from the key factor is 3.35. Also, the number of edges whose decision ratio is larger than 1.5 is 28 for the LGA and 22 for the key factors (see Figs. 14 and 21). Therefore, we can say that the discrimination performance of the key factor is degraded by approximately 21.4% (1-22/28) as opposed to that of the LGA.

VI. Navigation Experiment

For further validation, a navigation experiment was performed in a large building with 6 nodes (see Fig. 22). The proposed method was applied to the hierarchical atlas framework [27] which has two levels. At the highest level, a topological map organizes the free space into submaps at the lower level. The lower-level submaps are a collection of features [27]. In our experiment, we used the LGA and the key factors as the submap.

First, the mobile robot shown in Fig. 12 traveled the entire space to collect reference edge data. A total of 14 reference LGAs and 6 node signatures were acquired. The map generated during this travel is shown in Fig. 23. Note that the map was distorted because of odometry errors. During the collection of the reference LGAs, the departure angle which corresponds to the robot's heading direction when it leaves a node should be carefully selected. For that purpose, one can

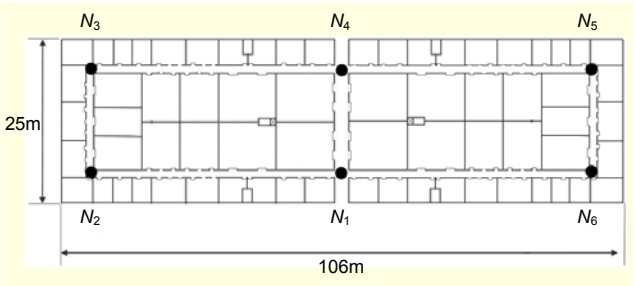

Fig. 22. Schematic map of a large building with 6 nodes (N_1-N_6).

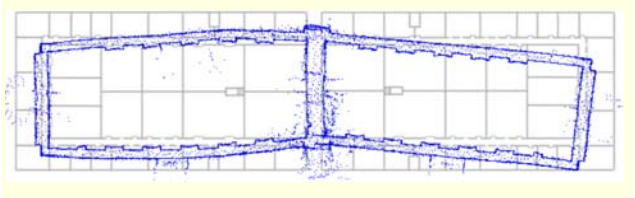

Fig. 23. Map generated during the navigation experiment.

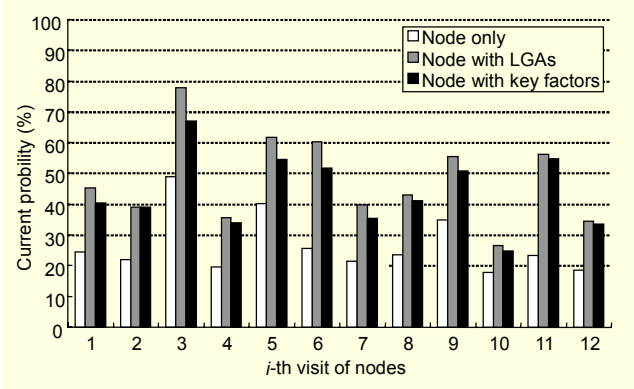

Fig. 24. Current probabilities for three cases: (1) use of node signatures only, (2) node and LGAs, and (3) node and key factors.

use artificial landmarks such as StarLITE [29] or StarGazer [30] which provide accurate heading information. We manually set the departure angle parallel to the corridor in this experiment.

With the reference data, the robot traveled 488.0 m for 45 minutes with a velocity of 0.35 m/sec. During this navigation, the robot visited a total of 12 nodes. Within the hierarchical atlas framework, the current probability, which is the probability of the robot to be located in the current position, was calculated for three cases (see Fig. 24): use of node signatures only, node and LGAs, and node and key factors. In general, the current probability is calculated using all accumulated sensor data. Thus, as the robot travels, the current probability also increases. In this case, it is hard to observe the effect of the addition of the edge information because both the accumulated sensor data and the addition of the edge information increase the current probability.

To alleviate this problem, the current probability was calculated using the node signatures at the current time and the information from the edges that the robot just traveled past. The node signatures that we used were the number of the emanating edges and the distance sensor data for 360° with a resolution of 1°. We intentionally excluded the odometric information because it belongs to the accumulated sensor data.

Note that the current probabilities noticeably increase with the use of nodes and LGAs compared to the usage of node signatures only. The performance is increased by 1.79 times because the average of the current probabilities with LGAs is 47.9% while that with node signatures is 26.7%. In this case, however, the average calculation time for each visit to a node is 826.35 seconds which is too long for real time calculation. If the key factors were used instead of the LGAs, the calculation time was greatly reduced to 0.91 seconds while the current probabilities were slightly decreased.

From these results, we can say that the LGA describes the edge shapes well and that the use of key factors is an efficient compression method for the LGA.

VII. Conclusion

In this paper, we described a method to efficiently represent edge shapes. First, a data format, the LGA, was defined to describe the shape of edges. However, the LGA used a large number of samples; thus, real time localization could not be performed.

To alleviate this problem, a compression method which utilizes wavelet transformation was proposed. The method abstracts the LGA by key factors using far fewer samples than the LGA. Also, dynamic time warping was adopted for the matching of edges whose numbers of samples were different.

Two experiments were performed. In the first experiment, it was shown that the LGA accurately described the shape of edges. However, it took a long time (516.9 seconds) for the calculation because of the large number of samples in the LGA. The key factors reduced the number of samples by 96.1% while preserving more than 95% of the information of the LGA. The comparison with the key factors took less than a second with slightly degraded discrimination performance.

The second experiment was performed by applying the proposed method to a navigation framework. This experiment showed that the localization performance was increased by 1.79 times with the use of LGAs. If key factors were used instead of LGAs, the calculation time was reduced to 0.91 seconds while current probabilities were slightly decreased.

The proposed method presumes a static environment. In the future, we will extend the method so that it can be used for the localization algorithm in dynamic environments.

References

[1] H. Choset and K. Nagatani, "Topological SLAM: Toward Exact Localization Without Explicit Localization," *IEEE Trans. on Robotics and Automation*, 2001, pp. 125-137.

[2] F. Abbattista and D. Dalbis, "The Scout Algorithm to Explore Unknown Spaces," *IEEE World Congress on Computational Intelligence*, 1998, pp. 705-708.

[3] M. Piaggio, A. Sgorbissa, and R. Zaccaria, "Global Localisation via Sub-graph Isomorphism," *European Workshop on Advanced Mobile Robots*, 1999, pp. 151-158.

[4] N. Tomatis, I. Nourbakhsh, and R. Siegwart, "Hybrid Simultaneous Localization and Map Building: Closing the Loop with Multiple-Hypotheses Tracking," *Int'l Conf. Robotics and Automation*, 2002, pp. 2749-2754.

[5] T. Bailey, E.M. Nebot, J.K. Rosenblatt, and H.F. Durrant-Whyte, "Data Association for Mobile Robot Navigation: A Graph Theoretic Approach," *Int'l Conf. Robotics and Automation*, 2000,

pp. 2512-2517.

[6] A. Tapus, S. Vasudevan, and R. Siegwart, "Toward a Multilevel Cognitive Probabilistic Representation of Space," *Int. Conf. on Human Vision and Electronic Imaging*, 2005, pp. 16-20.

[7] K. Suzuki, I. Horiba, and N. Sugie, "Neural Edge Enhancer for Supervised Edge Enhancement from Noisy Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, 2003, pp. 1582-1596.

[8] S. Konishi, A.L. Yuille, J.M. Coughlan, and S.C. Zhu, "Statistical Edge Detection: Learning and Evaluating Edge Cues," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 255, no. 1, 2003, pp. 57-74.

[9] M.B. Ahmad and T.S. Choi, "Local Threshold and Boolean Function Based Edge Detection," *IEEE Trans. Consumer Electronics*, vol. 45, no. 3, 1999, pp. 674-679.

[10] C. Fong and W. Cham, "An Improved Edge-Model Based Representation and Its Application in Image Postprocessing," *IEEE Int'l Conf.Acoustics, Speech, and Signal Processing*, 2003, pp. III265–III268.

[11] F. Moschetti, K. Sugimoto, S. Kato, and M. Burrini, "A Hybrid Wavelet and Ridgelet Approach for Efficient Edge Representation in Natural Images," *IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, 2004, pp. III677–III680.

[12] T.Y. Kim and J.H. Han, "A Fuzzy Approach to Edge Detection and Representation," *IEEE Int'l Conf. Fuzzy Systems*, 1997, pp. 69-74.

[13] T.F. Cootes and C.J. Taylor, "On Representing Edge Structure for Model Matching," *IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, 2001, pp. I1114-I1119.

[14] O. Kwon and C. Lee, "Edge Representation and Recognition Using Neural Networks," *IEEE Inte'l Symp. Industrial Electronics*, 2001, pp. 110-113.

[15] T. Glauser and H. Bunke, "Edge Length Ratios: An Affine Invariant Shape Representation for Recognition with Occlusions," *IAPR Int'l Conf. Pattern Recognition*, 1992, pp. 437-440.

[16] R.M.F. Ventura, L. Granai, and P. Vandergheynst, "R-D Analysis of Adaptive Edge Representations," *IEEE Workshop on Multimedia Signal Processing*, 2002, pp. 130-133.

[17] K. Nagatani and H. Choset, "Toward Robust Sensor Based Exploration by Constructing Reduced Generalized Voronoi Graph," *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1999, pp. 1687-1692.

[18] M. Kruusmaa, "Global Navigation in Dynamic Environments Using Case-Based Reasoning," *Autonomous Robots*, no. 14, 2003, pp. 71-91.

[19] B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, and F. Savelli, "Local Metrical and Global Topological Maps in the Hybrid Spatial Semantic Hierarchy," *Int'l Conf. Robotics and Automation*, 2004, pp. 4845-4851.

[20] H. Choset and J. Burdick, "Sensor Based Planning I: The Generalized Voronoi Graph," *Int'l Conf. Robotics and Automation*, 1995, pp. 1649-1655.

[21] H. Choset and J. Burdick, "Sensor Based Planning II: Incremental Construction of the Generalized Voronoi Graph," *Int'l Conf. Robotics and Automation*, 1995, pp. 1643-1648.

[22] S.G. Mallat, "A Theory for Multi-resolution Signal Decomposition: The Wavelet Representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, 1989, pp. 674-693.

[23] I. Daubechies, "The Wavelet Transform, Time-Frequency Localization and Signal Analysis," *IEEE Trans. Information Theory*, vol. 36, no. 5, 1990, pp. 961-1005.

[24] A. Boggess and F.J. Narcowich, *A First Course in Wavelets with Fourier Analysis*, Prentice-Hall, 2001.

[25] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algoritm, and System Development*, Prentice-Hall, 2001.

[26] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 26, no. 1, 1978, pp. 43-49.

[27] B. Lisien, D. Morales, D. Silver, G. Kantor, I. Rekleitis, and H. Choset, "The Hierarchical Atlas," *IEEE Trans. on Robotics*, vol. 21, no. 3, 2005, pp. 473-481.

[28] X. Ma, C. Zhou, and I.J. Kemp, "Wavelets for the Analysis and Compression of Partial Discharge Data," *Annual Report Conference on Electrical Insulation and Dielectric Phenomena*, 2001, pp. 329-334.

[29] H. Chae, J. Lee, W. Yu, and N.L. Doh, "StarLITE: A New Artificial Landmark for the Navigation of Mobile Robots," *Japan-Korea Joint Symp. Network Robot Systems*, 2005, pp. 11-14.

[30] http://www.hagisonic.com, 2006.

Nakju Lett Doh (M'02) received his BS, MS, and PhD degrees in mechanical engineering from the Pohang University of Science and Technology (POSTECH), Korea, in 1998, 2000, and 2005, respectively. In 2006, he joined the School of Electrical Engineering, Korea University, as an assistant professor. In 2003, he won the Best Student Paper Award in IEEE International Conference on Robotics and Automation. He also received the gold and the bronze prize in Humantech Thesis Competition hosted by Samsung Electronics in 2005 and 2000, respectively. His research area is the mobile robotics including localization, mapping, and motion planning of robots.

Wan Kyun Chung received his BS degree in mechanical design from Seoul National University in 1981, his MS degree in mechanical engineering from KAIST in 1983, and his PhD in production engineering from KAIST in 1987. He is a professor in the School of Mechanical Engineering, POSTECH (he joined the faculty in 1987). In 1988, he was a visiting professor at the Robotics Institute of Carnegie-Mellon University. In 1995 he was a visiting scholar at the University of California, Berkeley. His research interests include the localization and navigation of mobile robots, underwater robots, and the development of robust controllers for precision motion control. He is a director of the National Research Laboratory for Intelligent Mobile Robot Navigation. He is serving as an associate editor for IEEE Trans. on Robotics, international editorial board for Advanced Robotics.