A Robust Localization Algorithm in Topological Maps with Dynamic Noises


* Robotics and Bio-mechatronics Lab., Mechanical Engineering, POSTECH, Pohang, Korea. Tel: 82-54-279-2844, Fax: 82-54-279-5994, Email: {lekomin, wkchung}@postech.ac.kr

** Robotics Lab., School of Electrical Engineering, Korea University, Seoul, Korea. Tel: 82-3290-4822, Fax: 82-3290-3666, Email: {nakju,dltmdrbekt}@korea.ac.kr

*** Department of Information & Communication, Kyungmoon College, Kyeongki, Korea
Email: r13337@unitel.co.kr

Abstract

Purpose: We propose a localization algorithm for topological maps constituted by nodes and edges in a graph form. Our focus is to develop a robust localization algorithm that works well even under various dynamic noises.

Design/Methodology/Approach: For robust localization, we propose an algorithm which utilizes all available data such as node information, sensor measurements at the current time step (which are used in previous algorithms) and edge information, and sensor measurements at previous time steps (which have not been considered in other papers). Also, the algorithm estimates a robot's location in a multi-modal manner which increases its robustness.

Findings: We show that the proposed algorithm works well in topological maps with various dynamics which are induced by the moving objects in the map and measurement noises from cheap sensors.

Originality/value of the paper: Unlike previous approaches, the proposed algorithm has three key features: 1) usage of edge data, 2) inclusion of history information, and 3) a multi-modal based approach. By virtue of these features, we have developed an algorithm that enables robust localization performance.

Keywords: Topological Map, Mobile Robot, Localization, Kidnapping, Edge Data.

Classifications: Research paper.

1 To whom all correspondence should be addressed.
A Robust Localization Algorithm in Topological Maps with Dynamic Noises

I. Introduction

Topological maps which abstract an environment as nodes and edges in a graph form are very widely used representations.

However, there is a significant defect that a detection of a false node causes a serious problem. In a topological map, there are many nodes with different identities (id). At the same time, there are various noises such as a human walking, door opening, and so on. It is essential to discriminate the node's id even under various noises and we call this discrimination, “robust localization”.

In this paper, we propose a systematic framework that enables robust localization in topological maps. For robust performance, the framework is designed with two basic philosophies. The first philosophy is to utilize as much of the data as possible. The proposed framework uses both the node and edge data while previous research mainly utilized node information only (Abbattista and Dalbis, 1998; Bailey et al., 2000; Choset and Nagatani, 2001; Piaggio et al., 1999; Nagatani et al., 1998; and Tomatis et al., 2002). Also, the framework considers not only current observations but also history data that the robot has collected so far. The second philosophy is to presume a robot's location in a multi-modal manner. In other words, the framework does not restrict the current node position to a node where the robot is supposed to stop (which is evidently extracted from the given map). There are similar approaches that aim for robust localization in topological maps. These approaches can be categorized into three veins. The first approach is to enhance the performance of feature detections on a node. There are several methods which propose robust feature detection from a visual image (Blaer and Allen, 2002; Bradley et al., 2005; Radhakrishnan and Nourbakhsh, 1999; Wang et al., 2006a;
Wang et al., 2006b; Ulrich and Nourbakhsh, 2000). In 2004, Tapus (Tapus et al., 2004) suggested a concept of a fingerprint by combining various sensor data using a Bayesian formalism to detect features of a node in a robust manner. For a topological map in subterranean environments, Bradley (Bradley et al., 2004) proposed a regional point descriptor for localization. These approaches consider a feature description of nodes while our method considers a framework in a higher level. In other words, the feature description is a sub-problem of the framework and it can be integrated into our proposed framework.

The second approach is to reduce the searching zone of the topological map. Choi and Kown (Choi et al., 2002; Kwon et al., 2006) implemented a particle filter algorithm just into the Voronoi diagram to increase computational efficiency. Similarly, Yan (Yan et al., 2006) improved particle filters by a selective distribution for a large-scale environment. However, their main goal is the localization of a robot in an entire space while our aim is localization at the node.

The third approach is to combine the topological map with a metric map. This combined map is called a hybrid map where coarse localization is done in the topological map. Fine localization is performed in the metric map (Blanco et al., 2006; Tomatis et al., 2001; and Zhuang, 2006). In our approach, however, we only consider a topological map. This technique can be integrated into the hybrid map.

This paper is organized as follows. In Section 2, we suggest a framework for robust localization based on Bayesian formalism which includes the usage of edge and history data. In Section 3, we propose a dual-map that enables a multi-modal approach. In Section 4, we describe the localization framework. We give experimental results in Section 5. Then, we conclude.
2. A Framework for Robust Localization

2.1. Basic terminologies

A topological map consists of nodes and edges whose definitions are:

- **Nodes**: $N_1, N_2, \ldots, N_\eta$ where $\eta$ is the number of nodes.

- **Edges**: $E_{ij}$ is an edge from $N_i$ to $N_j$ for all $i = 1, 2, \cdot \cdot \cdot, \eta$ and $j = 1, 2, \cdot \cdot \cdot, \eta$. $E_{ij} = \emptyset$ if there is no connection between $N_i$ and $N_j$. $E_{ii}$ is a self loop if $i = j$.

Each node contains meaningful data such as odometry, sensor scans, number of edges, and visual scenes from a camera. Let us define $\#A$ as the number of samples in a set $A$ and use a superscript $1:#B$ in $B^{1:#B}$ to express all the elements in set $B$ (i.e. $B = \{B^1, B^2, \cdot \cdot \cdot, B^{\#B}\}$).

Explicitly, the data in a node is:

- $(x_i, y_i)$: the position of the robot at the $i$-th node.
- $\Sigma_i$: the covariance of odometry at the $i$-th node.
- $S_i^{1:#S}$: sensor scan from $S_i^1$ to $S_i^{\#S}$ at the $i$-th node.
- $\#E_i$: number of emanating edges at the $i$-th node.
- $V_i^{1:#E}$: visual scenes along the edge’s directions at the $i$-th node.

In localization, we assume that a map which contains all of the information of the nodes is provided in advance (denoted “a prior map”). Thus, localization is a process to estimate a current node where the robot is located using a prior map and observations. The observations consist of a node observation at a current time step $T$ (We denote it as $NO_T$) and the accumulated information of visited nodes and edges (Denote these as $His(NO)$ and $His(EO)$ for nodes and edges, where “His” denotes history). Explicitly,

- $NO_T = \{(x_T, y_T), \Sigma_T, S_T^{1:#S}, \#E_T, V_T^{1:#E}\}$,
- $His(NO) = \{NO_{T-1}, NO_{T-2}, \cdot \cdot \cdot, NO\}$, and
- $His(EO) = \{EO_{T-1}, EO_{T-2}, \cdot \cdot \cdot, EO\}$. 


2.2. Basic equations for the probability calculation

Let us denote $N$ as the current node. A probability of $N = N_i$ for all $i = 1, \cdots, \eta$ can be derived by the Bayesian rule as,

$$P(N = N_i \mid NO_T, His(NO), His(EO)) = \frac{P(NO_T \mid N_i, His(NO), His(EO)) \times P(N_i \mid His(NO), His(EO))}{P(NO_T \mid His(NO), His(EO))} = \xi \times P(NO_T \mid N_i, His(NO), His(EO)) \times P(N_i \mid His(NO), His(EO)) = \xi \times P_{\text{present}} \times P_{\text{history}},$$

where $\xi = 1/P(NO_T \mid His(NO), His(EO))$ is a scaling factor. Here, the probability of present and history is given by $P_{\text{present}} = P(NO_T \mid N_i, His(NO), His(EO))$ and $P_{\text{history}} = P(N_i \mid His(NO), His(EO))$, respectively. The subscripts $\text{present}$ and $\text{history}$ denote that the probability depends on the information of present and history observations, respectively.

2.3. The present probability

The present probability, $P_{\text{present}}$, can be written as,

$$P_{\text{present}} = P(NO_T \mid N_i, His(NO), His(EO)) = P(NO_T \mid N_i), \quad (1)$$

because $\text{His(NO)}$ and $\text{His(EO)}$ are independent of $NO_T$. Here, $P_{\text{present}}$ can be decomposed into,

$$P(NO_T \mid N_i) = P((x_T, y_T), \Sigma_T \mid (x_i, y_i), \Sigma_i) \times P(S_T^{1\#S} \mid S_1^{1\#S}) \times P(#E_T \mid #E_i) \times P(V_T^{1\#E} \mid V_1^{1\#E}), \quad (2)-(5)$$

where (2)-(5) denote probabilities based on the measurements of odometry, sensor scans, number of edges, and the visual scene, respectively.

First, to get the probability of odometry in (2), let us define,
\[ P_{odo}(i) = \left[ \left( \frac{x_T - x_i}{\sum_{j=1}^{T} \left( \frac{x_T - x_j}{\sum_{j=1}^{T} (x_T - x_j)^2} \right) \sum_{j=1}^{T} (y_T - y_j) \right) \right] \times \left( \sum_{j=1}^{T} \right)^{-1}. \]  

(6)

where \( \|\cdot\|_2 \) denotes the Euclidean norm of \( \cdot \). The first term is the Mahalanobis distance (Duda et al., 2000) as shown in Figure 1. The second term is multiplied to the Mahalanobis distance to assign a higher probability for less \( \|\Sigma_T\|_2 \).

The term \( P_{odo}(i) \) is calculated for a candidate node \( N_i \). For many nodes, a set of \( P_{odo}(i) \) is acquired which should be normalized. A normalizer (denoted by \( \mathbb{N} \)) is used for the probability of odometry as,

\[ P((x_T, y_T), \Sigma_T \mid (x_i, y_i), \Sigma_i) = \mathbb{N}(P_{odo}(i)). \]  

(7)

Figure 1: The Euclidean distance from \( (x_i, y_i) \) to \( A \) is larger than that of \( B \). However, the Mahalanobis distance from \( (x_i, y_i) \) to \( A \) is closer than that of \( B \).

Figure 2. The procedure of calculating \( P_{sen}(i) \). (a) and (b) correspond to the situations of \( S_T^{1:5} \) and \( S_i^{1:5} \), respectively. To get \( P_{sen}(i) \), \( S_i^{1:5} \) is rotated in a manner that minimizes the errors between two sensor scans as in (c).
Second, to calculate the probability of a sensor scan, let us define,

\[
P_{\text{sen}}(i) = \max_{\alpha=1, \ldots, \#S} \left\{ \sum_{\beta=1}^{\#S} \left| NO_T(S_\beta) - N_i(S_{(\beta-\alpha)_{\text{mod}}}) \right| \right\}^{-1},
\]

where \((\cdot)_{\text{mod}}\) denotes a modulation from 1 to \#S. For example, let us assume that 36 distance sensors are used (i.e. \#S=36). Then \((\beta-\alpha) = -1\) corresponds to \((\beta-\alpha)_{\text{mod}} = 36\) and so on. The physical meaning of (8) is that a sensor scan at the \(i\)-th node is rotated so that errors between \(S_i^{1:\#S}\) and \(S_T^{1:\#S}\) are minimized as shown in Figure 2. We use the inverse of the minimum difference to get \(P_{\text{sen}}(i)\). The probability of a sensor for \(N_i\) is given after normalization as,

\[
P(S_T^{1:\#S} | S_i^{1:\#S}) = N(P_{\text{sen}}(i)).
\]

Third, to get the probability of the number of edges in (4), let us define,

\[
P_{\text{edge}}(i) = \begin{cases} 
1 & \text{if } \#E_T = \#E_i, \\
0.1 & \text{otherwise}.
\end{cases}
\]

Simply the probability of the number of edges for \(N_i\) is,

\[
P(\#E_T | \#E_i) = N(P_{\text{edge}}(i)).
\]

Finally, the probability for vision is calculated based on the Scale-Invariant Feature Transform (SIFT) algorithm (Se et al., 2002) which returns a matching number of features between the two scenes. Simply, \(P(V_T^{1:\#E} | V_i^{1:\#E})\) is the returned value from the SIFT algorithm after a normalization.

Two remarks should be made. First, the node information is constituted by the sensors that a robot uses. In this study, we assumed that the robot is equipped with a sensor combination of odometry, distance sensors, and camera which is the most prevalent case. Second, the number of edges is assumed to be uncorrelated to the sensor scan which enabled the decompositions in (2)-(5). Strictly, this is not true because the number of edges is extracted from the sensor scan. But this is a reasonable approximation if two sensor scans are different because the sensor scan
reflects the arbitrary shape of a node while the number of edges reveals core topological information. If two sensor scans are the same, the equation of the sensor scan and the number of edges should be combined into one equation by neglecting one of them.

2.4. The history probability

Previous approaches mainly utilized sensor observations at the current time step (Choset and Nagatani, 2001; and Nagatani and Choset, 1999). However, the probabilities of the previous locations (which we call ‘history probability’) also can be used for the location at the current time step if the robot is not kidnapped. In this section, we provide a systematic formula that uses not only the current observation but also the history probability.

For the systematic approach, let us define the following concepts.

• Near node of level $l$: A node $N_j$ is a near node of $N_i$ with level $l$ if $N_j$ can be reached from $N_i$ by visiting $l$ edges. The set of near nodes of $N_i$ is denoted by $NN^{[l]}(N_i)$. For example, a set of nodes which can be reached from $N_i$ after visiting one edge is $NN^{[1]}(N_i)$ (Figure 3(a)). Similarly, $NN^{[2]}(N_i)$ is a set of nodes that can be arrived at after visiting two nodes (Figure 3(b)). Note that $N_i \in NN^{[2n]}(N_i)$ for all integers $n$ because any node can be reached to itself after visiting an even number of edges.

• Near edge of level $l$: An edge $E_{ij}$ is a near edge of $N_i$ with level $l$ if $N_i \in NN^{[l-1]}(N_i)$ and $N_j \in NN^{[l]}(N_i)$. The set of near edges of $N_i$ with level $l$ is denoted by $NE^{[l]}(N_i)$. Examples of $NE^{[1]}(N_i)$ and $NE^{[2]}(N_i)$ are shown in Figure 4.
Let us assume a topological map in Figure 5(a) and a path history which consists of the current edge observation \((EO_T)\), \(His(NO)\), and \(His(EO)\) as shown in Figure 6. Let us specify a searching limit by defining a depth of inspection for the path history as \(H\). Also, a set of connected paths from \(NE^{[H]}(N_i)\) to \(N_i\) is defined as,

\[
CP = \{i_1 p_1, i_2 p_2, \cdots, i_k p_k\}_{NN(i_{\#}\#i_{\#\#})}.
\]

From here on, \(i\) in the bottom-left corner will be omitted for simplicity. Figures 5(b-d) show three examples of connected paths for the map in Figure 5(a).

Now, let us define two similarity functions. One is a node similarity function \(NSim(N_a, N_b)\) in (13) which returns a high value if two nodes are similar.

\[
NSim(N_a, N_b) = P((x_a, y_a), \Sigma_a | (x_b, y_b), \Sigma_b) \times P(S_a^{\#S} | S_b^{\#S}) \times P(\#E_a | \#E_b) \times P(V_b^{\#E} | V_b^{\#E}).
\]

Note that (13) is the same with the present probability, (2)-(5), with substitutions of \(NO_T = N_a\) and \(N_i = N_b\).
Figure 5. Three examples of connected paths for a map in (a) where the left figure is the path history. In (b-d), similar edges and nodes with the path history are emphasized with thick lines. Among (b-d), (d) shows the best fit with the history and returns the highest $\text{SimCP}(\alpha)$.

The other is an edge similarity function $\text{ESim}(E_a, E_b)$ which returns a high value if two edges are similar. The edge similarity function can be calculated by using equations in (Doh et al., 2007) where the wavelet transformation and the dynamic time warping method were used to compare the similarity of two edges. Using the node and edge similarity functions, a similarity of the path history of an $\alpha$-th connected path can be calculated as,

$$\text{SimCP}(\alpha) = \prod_{\beta=1}^{h} \{ \text{NSim} (N_{O_{\tau-\beta} \cap NN^{\beta}(\beta)), \text{ESim} (EO_{\tau-\beta} \cap NE^{\beta}(\beta)) \} .$$

The physical meaning of (14) is shown in Figure 5. In this figure, similar edges and nodes with the path history are plotted with thick lines. Figures 5(b-d) correspond to $\text{SimCP}(\alpha)$ for $cp1$, $cp2$, and $cp3$. We can observe that $cp3$ is the best fit with the path history which, in return, yields the highest $\text{SimCP}(\alpha)$ value.

Finally, the history probability of a node $N_i$ to be the current node is calculated using the maximum value of $\text{SimCP}(\alpha)$ as,
Here, the history probability is sensitive to dynamic noises. For example, if a node is skipped by the noise, the history probability will yield a wrong value. This problem can be redeemed by a multi-modal approach which is explained in the next section.

3. Dual-map for Multi-Modal Approach

In the localization, there are three different situations: a) the static case where no noises are included, b) the dynamic case where sensor readings are contaminated by the noise, and c) the kidnapped case where the robot is kidnapped to another place. Because some sensors provide wrong information under dynamic noise, we need to take slightly modified approaches according to the situations. In this section, we provide localization algorithms for 3 different cases based on the probability equations derived in Section 2.

Figure 6. The path history.

Figure 7. The prior map has four nodes, A, B, C, and D, and doors are closed in nodes B, C, and D. The robot starts from A and does not stop at nodes B and C (the passed nodes), and arrives at node D whose sensor scan is different from the prior map (the changed node).
3.1. The probability in the static case

In the static case, all sensor data can be utilized. Thus, no modification is needed and probabilities can be calculated by (2)-(5) and (15).

3.2. The probability in the dynamic case

During the localization, the robot is assumed to stop at nodes such as,

- junctions that have more than 3 emanating edges,
- corners, and
- ends of corridors.

But if dynamic noise occurs, the robot either skips nodes (denoted as “a passed node”) or stops on a node different from the prior map (denoted as “a changed node”) as shown in Figure 7.

First, let us focus on the passed node. Note that a node can be passed under dynamic noises if the arrival and departure angles are similar. For example, if a robot starts from node A to node C (Figures 8(a, b)), then it will stop at node C in any of the cases. However, if a robot moves from node B toward node D, node C will be passed (Figure 8(c)) because the arrival and the departure angles are similar.

Based on this characteristic, a dual map where two nodes are directly connected if in-between nodes can be passed is constructed as shown in Figure 9. The key idea is that if the dual map is used, the history probability can be utilized even with dynamic noise. For example, let us assume a robot which had moved from A to D while skipping two nodes (Figure 10). However, the history probability can be calculated along the shaded path as in Figure 10.

To treat the changed node, some sensor data that is influenced by dynamic noise should be excluded. Those are the sensor scan and the number of edges. Thus the present probability should be modified as,

\[
P(NO_{t} \mid N_{t}) = P(x_{r}, y_{r}, \sum_{r} (x_{r}, y_{r}), \sum_{e}) \times P(V_{t}^{\text{SE}} \mid V_{t}^{\text{SE}}).
\]  

(16)
Figure 8. If a robot starts from node A to node C, (a, b) it will stop at node C in any of the cases. However, (c) if a robot moves from node B toward node D, the node C will be skipped because the arrival and the departure angles are similar.

Figure 9. Topological maps. (a) is the prior map, and (b) is the dual map where two nodes are directly connected if in-between nodes can be skipped.

Figure 10. A robot moved from A to D while skipping two nodes. However, the history probability can be used along the shaded path.

An integration of the present probability (16) with the history probability in the dual map gives the probability equation for the dynamic case.

3.3. The probability in the kidnapped case

The kidnapped case is the worst situation because most of the sensor data cannot be utilized. Dynamic noise also occurs during the kidnap recovery process. Thus, we cannot use the sensor scan and the number of edges.
Figure 11. A robot recognized its kidnapped state at node A in (a). The robot records its relative position for each visit of nodes from node A. The gross shape of the relative odometry is matched with the dual map. When the robot arrives at node I as in (b), the gross shape only can be matched into a path A-B-H-I in the dual map. The robot finally finds its position at node I via this relative odometry matching.

Two pieces of sensor data are available in this case. They are the vision and relative odometry information. Here, the relative odometry is a series of positions of nodes which starts at a node when the robot recognizes its kidnapped state. The relative odometry provides a gross shape of nodes which can be used for a matching with the dual map.

For example, a robot recognized its kidnapped state at node A in Figure 11. The robot records its relative positions for each visit of nodes from node A. The robot tries to match the gross shape of the relative odometry to the dual map. When the robot arrives at node I, the gross shape can be matched only to a path A-B-H-I of the dual map. The robot finally finds its position at node I via this relative odometry matching.

In summary, for the kidnapped case, the present and the history probabilities are acquired using the vision and the relative odometry based on the dual map.

4. The Robust Localization Algorithm

Let us denote a set of probabilities for the static, dynamic, and kidnapped cases as $SP_{sta}$, $SP_{dyn}$, and $SP_{kid}$. Note that each set consists of probability values where each value reflects the probability of a candidate node $N_i$ being the current node. The overall scheme of the
topological localization is a decision process using various sets of probabilities as shown in Figure 12. Here, a decision value D which takes an integer value including 0 is calculated in a way that maximizes the Rayleigh coefficient of a set of probabilities (Duda et al., 2000). Here, D = 0 denotes that there is no node with higher probability (i.e. no node to localize the robot). If D = 1, there is one node with a distinctively high probability. The robot is localized at that node. Otherwise (i.e. D ≥ 2), there are more than two nodes that have high probabilities and the robot moves to a near node to collect more information.

The proposed algorithm checks whether or not a decision value is 1 for the static, dynamic, and kidnapped cases. If one of the SPs is 1, the robot is localized with a probability of the maximum value of the corresponding SP. Otherwise, the robot moves to the next node for another chance or terminates the algorithm if it keeps failing.
5. Experimental results

For experiments, a differential drive type mobile robot equipped with two laser scanners (which are categorized as expensive sensors whose price is about $13,000 in total) and 12 sonar sensors (which are categorized as cheap sensors whose price is about $300 in total) was used (Figure 13). The experiments were performed under 3 different conditions such as 1) in a static environment, b) in a dynamic environment with expensive sensors, and c) in a dynamic environment with cheap sensors.

5.1. Experiment in a static environment

The mobile robot traveled a static indoor environment (Figure 14). The travel length was 52.7m and the initial position was given in advance. The robot visited a total of 16 nodes with probabilities higher than 0.99 which can surely be interpreted as a confident localization (Figure 15). These high values were acquired because the proposed algorithm utilized all useful data including history and edge information.
Figure 14. Photo of the static indoor environment.

Figure 15. The maximum probabilities for the static case.

Figure 16. Experimental environment with dynamic noises: a) a photo and b) schematic diagram.
Figure 17. Topological map of the space and the nodes that the robot visited. Initially, the robot position was not known and it recovered from the kidnapped state at the 4th visit of node A. Then the robot kept traveling under various dynamic noises such as door opening (closing) and human passage. By these dynamics, two nodes (D and F) were skipped during the navigation.

Figure 18. The maximum probabilities in the dynamic environment using expensive sensors.

Figure 19. Matching of two sensor scans for the correction of robot’s heading angle.
5.2 Experiment in a dynamic environment with expensive sensors

The robot navigated a dynamic environment shown in Figure 16. The total travel length was 86.28m and the elapsed time was 22 min. Figure 17 shows a topological map of the space and the nodes that the robot visited. Initially, the robot position was not known and it recovered from the kidnapped state at the 4th visit of node. Then the robot kept traveling under various dynamic noises such as door opening (closing) and human passage. By these dynamics, two nodes (D and F) were skipped during the navigation.

The proposed algorithm was used and the highest probabilities were shown in Figure 18. As shown in the figure, the probabilities were distinct and the robot could re-localize itself at the node with the highest value. There are three different ways of re-localization. Firstly, the robot can localize its position \((x,y)\) using a prior map information. Secondly, the robot can localize its states for both position \((x,y)\) and heading angle \((\theta)\). The correction of the heading angle can be conducted by comparing two sensor scans (one from the prior map, and the other from the measurement). In our experiment, we used laser scanners whose angle resolution is 2°. By virtue of the fine resolution, we were able to correct robot’s heading angle within an error bound of 2° by matching two scans (Figure 19). Thirdly, to compare the performance of the proposed algorithm, the Extended Kalman Filter (EKF) (Dissanayake et al., 2001; Guivant and Nebot, 2001; and Huang and Dissanayake, 2006) which is the most popular algorithm in the localization field was used. In this implementation, the information of detected nodes was used as measurement inputs.

The robot paths resulted by these three methods were shown in Figure 20. To provide a physical intuition of the amount of errors, these paths were projected to the schematic map as shown in Figure 21. The root mean square errors\(^2\) were given in table 1 and Figure 22.

From these results, we can observe that the localization of position and angle shows the best

\(^2\) After the kidnap recovery, the robot traveled along y-axis. Thus, the real position, which is needed for the calculation of errors, can be estimated by the travel length of the robot.
performance. This performance is acquired by virtue of reset of angle in each visit of nodes. As explained, the angle is reset within an error bound of $2^\circ$ and thus the error growth rate is noticeably decreased. For the localization of position, however, the error goes to 0 at each node. But it soon increases as the robot moves because the angle error was not calibrated. The EKF localization (which is the most popular localization algorithm) showed the worst performance because EKF does not consider dynamic situations.

Figure 20. Paths of (a) localization of position, (b) localization of position and angle, and (c) EKF localization compared to raw odometry in a dynamic environment with expensive sensors.
Figure 21. Projection of paths to the schematic map. This figure gives a physical intuition about the amount of position errors.

Table 1. Root mean square errors for the experiment in a dynamic environment with expensive sensors.

<table>
<thead>
<tr>
<th></th>
<th>Raw odometry</th>
<th>Localization (Position)</th>
<th>Localization (Position + Angle)</th>
<th>EKF Localization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average position error (m)</td>
<td>0.74 m</td>
<td>0.40 m</td>
<td>0.17 m</td>
<td>2.03 m</td>
</tr>
</tbody>
</table>

Figure 22. Root mean square errors for the experiment in the dynamic environment with expensive sensors.
5.3. Experiment in a dynamic environment with cheap sensors

Similar experiment was performed by using cheap sensors (odometry and 12 ultrasonic sensors). The robot traveled a total of 92m for 37 min. By the dynamic noises, 4 nodes were skipped during the navigation. This was the worst situation from the robot's point of view because the robot did not know its initial position and the cheap sensor provided noisy measurements. However, the proposed algorithm was successfully implemented and the highest probabilities were shown in Figure 23. At the fifth visit, the robot recovered from the kidnapped state and robustly dealt with the noises at its seventh and tenth visits.

The robot paths for the three cases (a) localization of position, (b) localization of position and angle, and (c) EKF localization were shown in Figure 24. The root mean square errors were given in table 2 and Figure 25. With cheap sensors, the localization of position showed the best performance. The localization of position and angle (which showed the best performance in the experiment with expensive sensors) showed the worst performance. With cheap sensors, only small number of sensor data is provided with high noises. The angle matching with these data yields large amount of angle errors which, in return, causes high position errors. Also, the performance of the EKF localization was not satisfactory because EKF does not consider the dynamic situation.

![Figure 23. The maximum probabilities in the dynamic environment using cheap sensors.](image)
Figure 24. Paths of (a) localization of position, (b) localization of position and angle, and (c) EKF localization compared to raw odometry in the dynamic environment with cheap sensors.
Table 2. Root mean square errors for the experiment in a dynamic environment with cheap sensors.

<table>
<thead>
<tr>
<th></th>
<th>Raw odometry</th>
<th>Localization (Position)</th>
<th>Localization (Position + Angle)</th>
<th>EKF Localization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average position error (m)</td>
<td>3.50 m</td>
<td>1.00 m</td>
<td>3.19 m</td>
<td>2.25 m</td>
</tr>
</tbody>
</table>

Figure 25. Root mean square errors for the experiment in the dynamic environment with cheap sensors.

6. Conclusion

We proposed a robust localization algorithm in topological maps. Unlike previous approaches, the proposed method has the following three key features. Firstly, the framework used both the node and edge data while previous research mainly utilized node information only. Second, the framework considered not only the current observation but also history data that the robot collected so far. Finally, the framework presumed the robot's location in a multi-modal manner. In other words, the framework did not restrict current node position to a node to which the robot was supposed to stop. Three sets of experiments were performed to validate the performance. It was shown that the proposed algorithm works well in a dynamic environment.
Also, the proposed algorithm performs better than the extended Kalman filter, which is a most popular localization tool, in the dynamic environment.

Two remarks should be made. First, our approach is only good for topological maps. If there are big halls in between topological maps, the proposed method should be modified appropriately. Second, the major application of our approach is guide robots whose major job is navigation in a topological map with dynamics.

Acknowledgement

This work was supported in part by the KOSEF grant of MOST [No. R0A-2003-000-10308-0], by ], by the grant of the Korea Health 21 R&D Project [A020603], by the KICTEP grant of ‘06-Unified and Advanced Construction Technology Program-D01’, and by the grant of Korea University.

References


