Experimental Research of Map Building and Localization at Human Co-existing Real Environments

Dongheui Lee, Woojin Chung, and Munsang Kim

Intelligent Robotics Research Center, Korea Institute of Science and Technology,
39-1Hawolgok-dong Sungbuk-gu, Seoul 136-701, Korea
{dhlee96, wjchung, munsang}@kist.re.kr

Abstract: Map building and position estimation capabilities are practically indispensable for a mobile robot to execute its given tasks in its working environments. An autonomous map building method and a smart localization method is proposed in our previous works. The experimental verifications are carried out in this paper. We applied the proposed algorithms to mobile service robots in large-scale indoor buildings. Experimental results show that our strategy is reliable and feasible in tough conditions like non-polygonal and dynamic environments. The advantages of the algorithms are well-illustrated through real experiments.

Keywords: localization, map building, mobile robot, dynamic environment, non-polygonal environment, discrete event control

1. INTRODUCTION

Guide Robot Jinny is developed by Intelligent Robotics Research Center in KIST (Korea Institute of Science and Technology). Jinny provides various services like exhibition guide, public information, guidance of the road in large indoor buildings. Jinny is going to carry out its jobs in real environments, such as a lobby at KIST and an exhibition hall at Hyundae Heavy Industry. In order to perform its tasks successfully, autonomous navigation technology and human robot interaction technology must be embodied. Especially map building and position estimation capabilities are practically indispensable for autonomous navigation.

Fig. 1 KIST guide robot Jinny

In our previous papers [1, 2], a map building method and an intelligent position estimation method of the indoor service robot is introduced conceptually. The feasibility is proven through the experiments in simple (polygonal and structured) environments like corridors.

The contribution of this paper is the experimental verification of the proposed algorithms in various real environments, such as non-polygonal and dynamic places. Map building and localization in non-polygonal environments is well-known as a difficult problem because it is hard to detect geometrical features like walls or columns. In our algorithm, the problem is solved easily in the following way. A mobile robot determines the environmental state through Hough transformation and selects matching methods automatically. For example, if the robot is in a non-polygonal place, it knows that its current environment does not contain features and performs only scan matching. If it is in a polygonal surrounding, its matching method is the combination both scan matching and feature matching. Mapping and localization in dynamic environments is also considered as one of challenging navigation problems. The smart localization is able to solve this problem easily by adopting discrete event control concept. It determines how much sensor data is polluted by dynamic factors. If the sensory information is not considered trustable, the information is not used for position estimation and robot position is updated from pure dead reckoning.

The rest of this paper is as the following. In chapter 2, our previous research works are summarized. Experimental research of map building and localization at human co-existing real environments will be presented in chapter 3. Experimental verifications are presented in chapter 4 and some concluding remarks are given in chapter 5.

2. OUR PREVIOUS WORKS

2.1 Map Building

Our autonomous map building strategy is based on three processes: (1) gathering environmental information, (2) registration of laser scan data, and (3) building grid maps.

When the robot is moving through the environment, it gathers environmental information. Basically, the map can be constructed by using dead reckoning and sensor data if the dead reckoning information is accurate. Unfortunately, it is not accurate enough in most cases due to its accumulated errors. Thus, an accurate map cannot be obtained without compensation of the robot's position.

Using our scan registration, which is a modified method of our previous localization, robot positional error is compensated. In other words, successive range images are connected consistently. The scan registration employs both geometric pattern matching and scan matching. The registration scheme without artificial landmarks can be implemented in a non-polygonal environment by scan matching. In polygonal surroundings, the accuracy is improved by adding the geometric pattern matching.

A grid map is adopted for environmental description instead of sensor data because of the efficiency and compactness of the grid map. After registration, the grid map is constructed by the Histogramic In-Motion Mapping (HIMM) method in [5]. The HIMM provides efficient accumulation of various sensor data.

Only scan matching is carried out in non-polygonal surroundings. Both feature matching and scan matching is carried out in polygonal ones because more reliable results can be obtained by using geometrical features. In our algorithm, Hough transform is used for feature detection. A geometric pattern is represented as the voting number in a Hough domain. The use of geometric pattern matching is decided autonomously, which depends on the voting number.

As a scan matching method, range image based gradient method is developed. The gradient method corresponds to the Markov localization using the evenly-spaced grids in 3D parametric space. During this procedure, the estimated robot position gets closer to the real position and error is minimized gradually. The certainties of robot's existence at 27 positions around the expected position are calculated. The estimated robot position is to be the position with the highest certainty value. The certainty values are estimated by range image matching RISF which is a similarity measure function for PSR localization [1]. More details about scan registration method can be found in our previous paper [2]. The proposed registration technique has the following advantages.

- An accurate map can be built without artificial landmarks.
- The scheme is applicable in non-polygonal environments by scan matching.
- In polygonal surroundings, the accuracy is improved by the geometric pattern matching.
- Each scan is registered accurately by comparing it with all previous scans in the local region rather than only two successive scans.
- An accurate map is built by checking the reliance of the each registered scan. If certainty value of the scan is lower than a pre-defined threshold, the scan is discarded.
- The computational cost is acceptably low.

2.2 Particle Filter Based Localization

Our localization method is a map-matching scheme using range scan data, without using any artificial landmarks. A probabilistic position estimation scheme is designed based on Monte Carlo localization (MCL). Probabilistic localization consists of two steps: prediction phase and update phase. Prediction phase forecasts the robot's position at the current time step with respect to the previous robot states and control input. Positional probabilities are newly updated according to the result of map matching computation. Similarity measure functions calculate the probability under conditions of the robot's current state and a given map. The probability is computed by comparing scan data and reference data. Two similarity functions, RISF and ASF, are developed for computing positional probabilities. The robot automatically decides whether or not to use geometric pattern matching (i.e. extracting walls, pillars) by means of the Hough transform. The details of the reliable position estimation method of the PSR are introduced in [1].

2.3 Smart Localization

There are a great number of situations where a robot should behave intelligently. For example, when a man gets lost, he returns back to his original place and starts traveling again. Also, if a man does not know where he is, he tries to find a big obvious building. Human can act intelligently like these cases. Due to the necessity of intelligence in navigation, an integrated localization strategy which is called smart localization is developed.

In smart localization, a robot can take appropriate actions like human beings by introducing discrete event control concept. Not only position estimation but also behaviors for better position estimation and localization error handling methods are considered. Petri net (PN) is used for the discrete event control, since robot activities can be event-driven and the state of navigation can be divided into the possible discrete states. When the robot is unable to compute its position, discrete event based error handling logic is activated according to the predetermined behavior configuration.

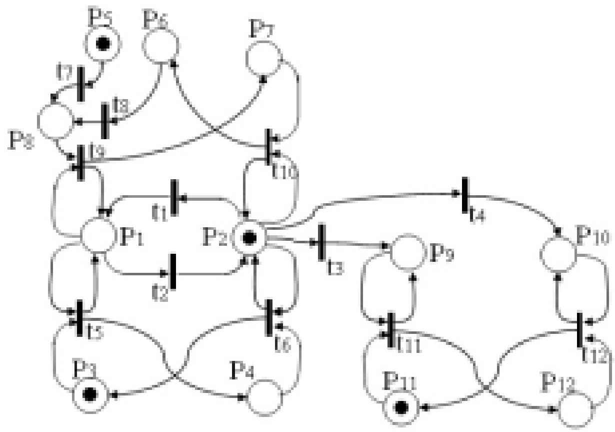


Fig. 2 Petri Net (PN) of Smart Localization

Table 1 Description of places and transitions of PN (Fig. 2)

	Description
P1	State: Local Success
P2	State: Not Local Success
P3	Behavior: Random Motion (Robot explores randomly.)
P4	Behavior: Planned Motion (Robot tracks the planned path.)
P5	Robot position is unknown.
P6	Relative Localization (robot position is known from odometry.)
P7	Absolute Localization (known from sensor based localization.)
P8	Robot position is unknown or known from odometry.
P9	State: Many dynamic obstacles surround the robot. (Crowded)
P10	State: ~P9 (Not crowded)
P11	Robot says nothing. Normal localization is executed.
P12	"Please move back." Update phase is not executed.
t1	Event: LocSucc (local localization and successful match)

t2	Event: NotLocSucc (~LocSucc)
t3	Event: Obstacles (Many dynamic obstacles surround the robot.)
t4	Event: NotObstacles (~Obstacles)
t5	State: Local Success and Random Motion
t6	State: Not Local Success and Planned Motion
t7	State: Robot position is unknown.
t8	Relative Localization (robot position is known from odometry.)
t9	State: LocSucc and (unknown position or relative localization)
t10	Absolute Localization
t11	State: Robot says nothing. Normal Localization. Obstacles
t12	State: "Please move back." No Update phase. NotObstacles

PN of localization is defined like figure 2. Table 1 shows the description of the places and transition of the PN. Every time localization algorithm iterates, smart localization checks the distribution of samples, reliance of the localization result, and existence of obstacles. It is decided whether localization state is local tracking problem or global positioning problem through the calculated sample distribution. Four events are defined in localization: *LocSucc*, *NotLocSucc*, *Obstacles*, and *NotObstacles*. *LocSucc* means that samples are converged into a local region and the estimated results from localization are successfully matched. *NotLocSucc* is the complementary event to *LocSucc*. *Obstacles* implies that too many moving obstacles near the robot caused an unsuccessful localization result. *NotObstacles* is the complementary event to Obstacles. Petri net enables the robot to take appropriate motion behavior, sound behavior, position update, and localizer behavior with respect to the localization events.

When a robot's navigation starts, tokens in the Petri net are placed in P2, P3, P5, P11 as in figure 1. At this status, the robot does not know its position and it moves randomly.

Because the robot does not know its initial location, initial samples are distributed over the whole region. Because it is a global localization problem, the event *NotLocSucc* occurs. However, t2 is not enabled so the transition cannot be fired. Therefore, the robot still does not know its absolute location, keeps exploring the environment randomly.

According to the result of localization, the globally distributed samples are converged into a local region. If also the estimated results from localization are successfully matched, the event *LocSucc* occurs and transition t1 is fired. Tokens move to P1, P4, P7 and P11. The robot knows its absolute position from localization. Path planner plans the traveling path to navigate and robot tracks the planned path.

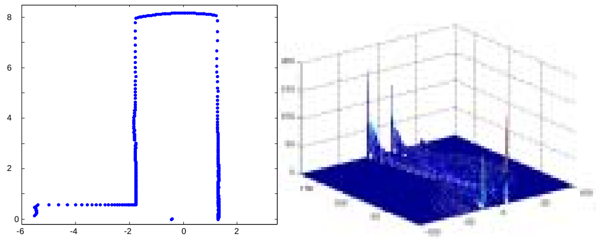
3. EXPERIMENTAL RESEARCH

3.1 Polygonal vs. Non-polygonal Environments

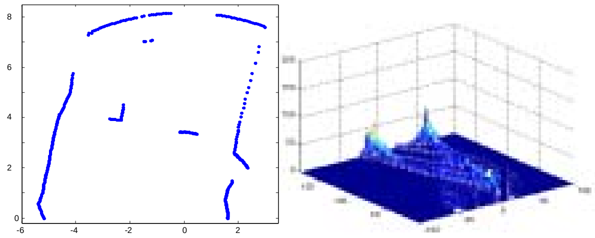
In our method, both scan matching and geometric feature matching is applied in polygonal environments and only scan matching is used in non-polygonal environments. By using Hough Transform results, the mobile robot can automatically determine whether to use geometric features and the number of geometrical features to extract.

If a voting number of a candidate feature is more than a pre-defined threshold, the candidate is considered as a feature. There are examples below. Figure 3 shows the laser scan data and voting number in Hough domain of the sensor data in a polygonal environment. Two main peaks are shown clearly in Hough domain (Fig. 3 (b)). The main feature is [ρ, Θ] = [-1.3m, 180°] and its voting number is 156. The second main feature is [ρ, Θ] = [1.8m, 180°] with voting number 119.

Figure 4 shows the laser data and its Hough transform results in a partially-not-polygonal environment. The maximum voting number is 70, which is less than the threshold. No candidate is considered as geometric features.



(a) Laser scan data (b) Hough domain
Fig. 3 Polygonal environment

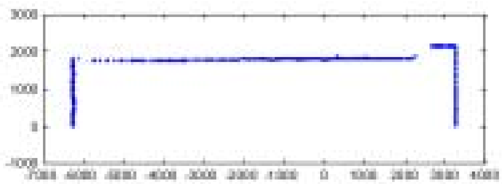


(a) Laser scan data (b) Hough domain
Fig. 4 Non-polygonal environment

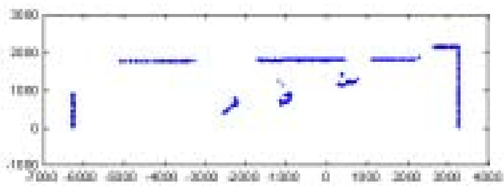
3.2 Static vs. Dynamic Environments

There are two important issues concerning dynamic surroundings: the reliability of map matching performance under dynamic obstacles and the degree of sensor corruption.

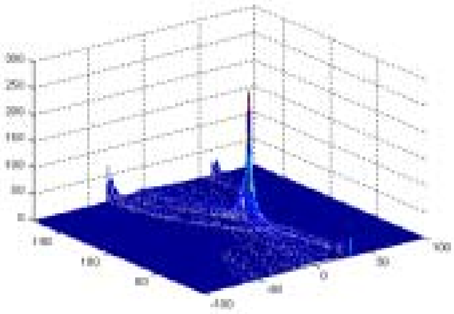
(1) Our map matching is reliable in a dynamic environment. Hough transform is employed in a dynamic environment, for example, only the voting number of the geometric patterns decreases and the main geometric patterns are still obtained accurately. Figure 5 shows laser data in a static and dynamic environment and its Hough transform results. In both cases, the same geometric feature, which is [ρ, Θ] = [1.8m, 91°], is found. The voting number was 254 and 178 in the static case and the dynamic case respectively.



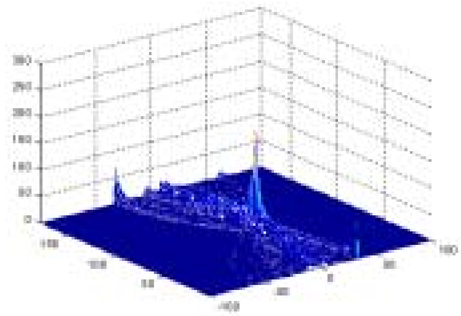
(a) Laser scan data



(b) Laser scan data



(c) Hough domain



(d) Hough domain
Fig. 5 Static vs. Dynamic environment

(2) When a mobile robot is surrounded by very many people and most parts of its sensor data are corrupted, the localization result with the corrupted sensor data is not trustable. For convenience of explanation, corruption ratio is used and it means the percentage of corrupted sensor data out of whole sensor data. If a robot can determine the sensor corruption ratio and the corruption ratio is larger than a threshold, it is better to skip the update phase because the similarity function with the corrupted sensor readings will incorrectly calculate samples' similarities. In other words, only the prediction phase is executed.

In order to calculate the corruption ratio, the reference distance and scan distance for the reference position is compared. Reference position is referred as expected position from odomtry. Reference position is used because the odometry information will be more accurate than the localization result with unreliable sensor data.

3.3 Reliability

An algorithm to check the reliability of the localization result is developed. If a robot is able to judge the reliability, it can decide whether to update the estimated position or not. A robot checks the reliability by comparing the reference distance and scan distance with respect to the estimated position. The method is similar to the way to check the corruption ratio in a dynamic case. The difference is which the reference position is, localization result or odometry. Reliability implies the percentage of the laser data whose error is less than 10 percent in this paper.

4. EXPERIMENTAL RESULTS

4.1 Map Building

This session shows the experimental results of the proposed registration method in a real environment. Since Guide robot Jinny is supposed to perform it tasks in the exhibition hall of Hyundae Heavy Industry (HHI), an empty office at the L7 building in KIST is set like the exhibition hall of HHI for a test environment (figure 6). The exhibition hall is non-polygonal and greatly dynamic. Its crowdedness changes greatly because of a number of visitors. The size of the test environment is 10 meters by 35 meters.

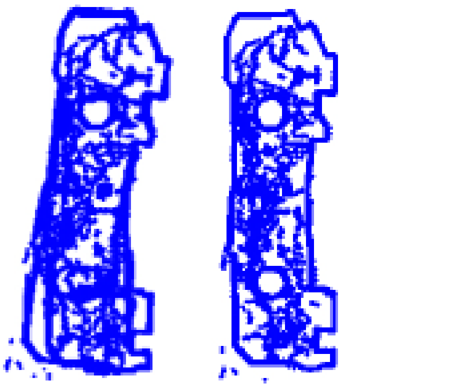
A map of the environment is generated by the proposed mapping algorithm. The results show that the robot can obtain an accurate map even in the dynamic and unstructured environments. While Jinny explored the test environment, several people were walking around Jinny like figure 6 (b).

Figure 7 (a) is laser scans with respect to the robot positions without the position correction. The figure shows that laser data is inconsistent because of two reasons: odometry error and polluted sensor data due to moving people. The laser scans after the proposed registration algorithm are illustrated in figure 7 (b). Polluted laser scans also are registered successfully. In other words, odometry errors disappear. The HIMM method is applied to construct a grid map. Figure 7 (c) and (d) are the maps obtained by the original HIMM and by our modified HIMM respectively.

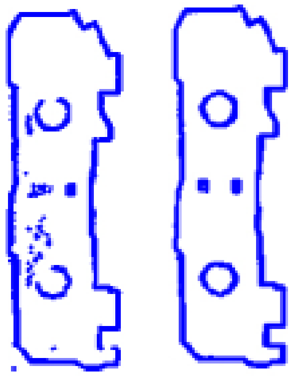
In the original HIMM method, the certainty value of each grid can be between 0 and 15. However, we learned that the conventional way of accumulation is not reliable and depends more on recent sensor data. For instance, if a grid had been empty most time and was occupied for a few seconds in the end, the grid would be considered occupied because of a few recent sensor data. Therefore, we modified the algorithm: each grid value is not limited so that the total accumulation is respected. From the experiments (figure 7 (c) and (d)), it is shown that the modified HIMM outperforms the original algorithm.

(a) Static real environment

(b) Dynamic real environment
Fig. 6 a real environment



(a) Sensor data w.r.t odometry (b) after scan registration



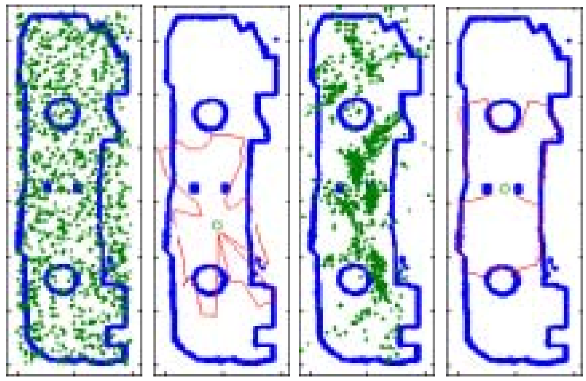
(c) Original HIMM result (d) Grid Map

Fig. 7 Map Building

4.2 Localization

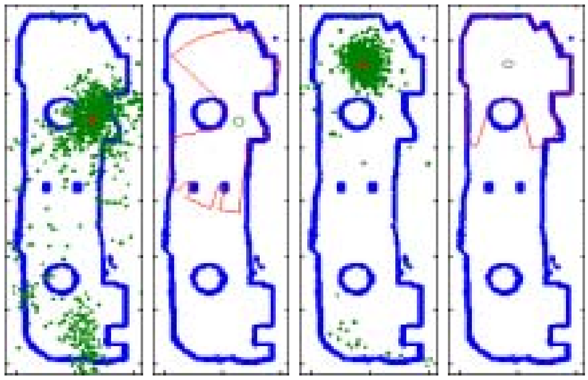
Localization experiments are carried out with the obtained map by the proposed map building strategy. A robot's position is described with a discrete probability sample set, and then the robot's position is computed according to our map matching algorithms. When a robot starts to move, a sample set is updated using the motion control input. At this stage, a pre-defined motion model determines the samples' positions and positional belief. The update phase estimates samples' probabilities to be in the present state with sensor data by using the two similarity functions.

Some localization results are illustrated to point out important characteristics of our algorithm. A procedure from the global positioning problem to the local tracking problem is shown in figure 8. Four steps during the procedure are selected and the sample distribution and scan data is displayed every step. Since the initial robot position is not given, a thousand samples are spread over the whole area. At the beginning, samples are shown in the x-y plane in the left of figure 8 (a). At this stage, the real robot position was [x, y, ϴ] = [3.9, 14.5, -1°] and the position with maximum probability was [4.44, 12.92, 11.25°]. The position with maximum probability is different from the actual position because of many ambiguous locations. The variance of sample distribution v(s) is 91494439 and the estimated position has low reliability. The laser scan data at the estimated position is not matched well to the map (the right figure 8(a)): the reliability is 12%. Therefore the event *NotLocSucc* occurs. The robot still does not know its position and explores the environment randomly according to the PN of smart localization.



Samples and sensor data at

(a) [3.9, 14.5, -1°] (b) [4.01, 16.45, -3.5°]



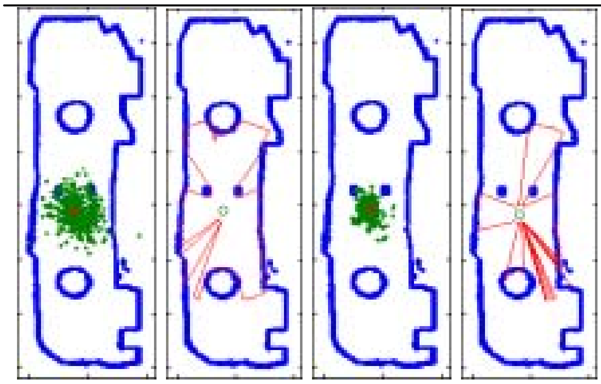
(c) [6.37, 22.49, 4.78°] (d) [4.19, 27.86, 79.4°]

Fig. 8 from global positioning to local tracking

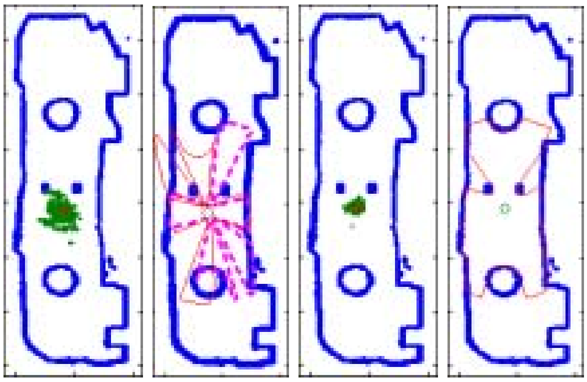
When the robot explores and arrives at position [4.02, 16.51, -3.1°], sample distribution and scan data is represented in figure 8 (b). The maximum probability position [4.01, 16.45, -3.5°] was almost same as the real position. From the right figure in Fig. 8 (b), it is shown that the most sensor data was matched accurately: its reliability is 89 %. Although the sensor based localization is successful, the samples are distributed globally. The variance of sample distance v(s) = 40701845 and the threshold of v(s) is set as 30000000 in our experiments. Since there can be multiple places where looks similar during global positioning problem, we do not take this result as successful. Thus, the event *NotLocSucc* occurs again; the robot does not know its position and keeps exploring the environment randomly.

When the robot is located at position [6.37, 22.55, 4.92°], samples are converged. The estimated position is [6.37, 22.49, 4.78°]. Its v(s) = 29988312 and its reliability is 74%. The localization state is local tracking problem and successful matching. The event *LocSucc* occurs. Therefore, the robot knows its position from sensor based localization: the maximum probability position is updated as the robot position. Robot plans how to travel and tracks its planned path.

When the robot is located at position [4.19, 27.86, 79.4°], odometry based position was [4.93, 27.77, 67.28°]. It is shown that dead reckoning has error and the estimated position was same as the real location. Its v(s) is 1438415 and its reliability is 99%. Therefore the *LocSucc* event occurs.



(a) Corruption ratio: 13.4 % (b) 49.8 %



(c) 55 % (d) 1.3 %

Fig. 9 Dynamic Situation

Figure 9 shows how Guide robot Jinny takes actions in a dynamic situation. Jinny stayed in the same spot while many obstacles were moving around it. In figure 9 (a), there was only one person near it. Only 13.4 percent of laser sensor data was corrupted and its reliability was 80.5 percent. Therefore, the localization result was successful. Many people are gathered toward Jinny (figure 9 (b)). At that time, almost 50 percent of sensor data was corrupted and the reliability of the localization result was 51 percent. It was successful. In figure 9 (c), however, the localization result was not successful because 55 percent of the sensor readings were polluted and its reliability was only 21.1 percent. The laser scan data is displayed with respect to the estimated position (solid curve) and the dead reckoning position (dotted curve). Event *NotLocSucc* and *Obstacles* occurs. Therefore, the update phase was skipped and the odometry position was updated rather than the localization result. After all the people go away (figure 9 (d)), the localization result is successful: only 1.3 percent data was corrupted and its reliability was 97.8 %.

Experimental results illustrate that the proposed smart localization technique can deal with various difficulties in the environments due to benefits from the discrete event control. For example, the global positioning and the local tracking problem can be distinguished by using variance of sample distance and the number of samples is changed for efficient localization estimation. Moreover, estimated positional belief is calculated and the robot can decide whether or not to update the estimated pose as the robot pose. It can check the crowdedness and ask people to clear out of its way. The smart localization outperforms the localization without a discrete event control in the various real worlds. Its localization accuracy is better and it is able to perform appropriate behaviors intelligently.

5. CONCLUSIONS

Sample based localization with map matching method is proposed in [1]. Autonomous map building and smart localization is introduced in [2]. They are mentioned conceptually and validated in structured surroundings. The experimental research of the map building and the smart localization algorithm is addressed in this paper. The scheme is implemented into a real robot system Jinny. The advantages of the algorithms are well-illustrated through various real experiments: polygonal vs. non-polygonal cases and both static vs. dynamic cases. Experimental results show that our strategy is reliable and feasible in tough conditions, such as non-polygonal and dynamic environments.

REFERENCES

[1] Dongheui Lee, Woojin Chung, Munsang Kim, "A Reliable Position Estimation Method of the Service Robot by Map Matching," *IEEE International Conf. on Robotics and Automation (ICRA),* 2003.

[2] Dongheui Lee, Woojin Chung, Munsang Kim, "Autonomous Map Building and Smart Localization of the Service Robot," *IEEE/RSJ International Conf. on Intelligent Robots and Systems (IROS)*, 2003.

[3] J.Borenstein & Y. Koren, "Histogramic In-Motion Mapping for Mobile Robot Obstacle Avoidance," *IEEE TRAN,. Robotics and Auto,*. Vol 7, No. 4, 1991.

[4] Gunhee Kim, Woojin Chung, Munsang Kim, Chongwon Lee, "Tripodal Schematic Design of the Control Architecture for the Service Robot PSR," *IEEE International Conf. on Robotics and Automation (ICRA), 2003*

[5] Murata, T., "Petri Nets: Properties, analysis and applications," *Proceedings of the IEEE, Volume: 77 Issue: 4, April 1989*