A Navigation Method using the Mutual Feedback of Waypoints and Self-positions

Kosei Demura Yuichi Komoriya

Kanazawa Institute of Technology, Ishikawa, Japan

demura@his.kanazawa-it.ac.jp

Abstract

The purpose of this paper is to solve a significant problem faced by waypoint-based robot navigation: the positions of the waypoints can be located in unreachable areas due to errors in self-localization, the map, or in the waypoint data.

Much research has focused on self-localization, but few studies have focused on estimating waypoints. The method presented in this paper estimates waypoints and self-positions using information from LIDAR (Light Detection and Ranging) processed using particle filters. If waypoints are placed in unreachable areas, the weights of the particles around the waypoint are decreased according to a geometric constraint, and the weights of other particles in reachable areas are increased. Because the waypoint is the expectation value of all particles, this process relocates the waypoint at a reachable location. Another characteristic of this method is the mutual feedback between the probability density functions (PDFs) of the waypoint and the self-position. This method first alters the PDFs of the self-positions using the PDFs of the waypoints, and then changes the PDFs of the waypoints using the PDFs of the self-positions, thus implementing mutual feedback and improving the accuracy of self-localization and the robustness of the waypoint navigation.

This method was used for a trial run of the Tsukuba Challenge 2010, and a 240 [m] run was successfully finished using only dead reckoning information. Other experiments show that this simple, unique method is very useful in waypoint-based robot navigation.

keywords: waypoint-based navigation, waypoint estimation, self-localization, mutual feedback, The Tsukuba Challenge

1 INTRODUCTION

Waypoint-based navigation is widely used for autonomous robot, human, vehicular, vessel and aircraft navigation because of its simplicity and compact representation [1, 2, 4]. The navigation scheme sequentially follows sets of ordered coordinates called waypoints.

The main problem of this navigation method is that errors in self-localization, inaccurate maps, or errors in waypoint placement can move waypoints to real-world locations that robots cannot navigate into, e.g., ponds, walls or hedges. This behavior becomes increasingly problematic as self-localization errors increase. In the Tsukuba Challenge [5], an open experiment of autonomous robot navigation through the pedestrian streets of Tsukuba, many robots failed because of this problem.

Thus, the purpose of this study is to solve this drawback of waypoint-based navigation by simultaneously estimating self-positions and waypoints using a particle filter. The motivation for this study is that our team, the demura.net team, retired during the Tsukuba Challenge 2009 because of this problem. In the Tsukuba Challenge 2010, our proposed method was used in the qualifying run, and successfully finished a 240 [m] run using only dead-reckoning information for the navigation (Figure 1).

This paper is organized as followings. Section 2 describes the related research. Section 3 introduces the proposed method. Some experimental results are presented in Section 4. Finally, Section 5 provides concluding remarks.

——— Insert Figure 1 ————-

2 RELATED RESEARCHES

Numerous studies on waypoint-based navigation have been presented[1, 2, 4, 12, 13]. The subgoal method represents one of related method of solving the problem of waypoint-based navigation. Krogh, Thorpe, and Feng proposed subgoal-based method that interpolates between preselected subgoals and dynamically avoids obstacles using data from a range finder [6, 7]. Barraquand et al. proposed a path planning technique using a sophisticated artificial potential field [8] , but this technique is difficult to use for real-time navigation. To solve this problem, Lagoudakis et al. have proposed a potential field method that sequentially uses an artificial neural network [9] . Maida et al. extended this method to automatically generate local waypoints or subgoals [2] .

Lavalle proposed rapidly-exploring random trees (RRTs) that can address the problem of real-time path planning [10]. Bruce and Veloso developed a path planner based on RRTs. Their work improved the efficiency of replanning and the quality of the generated paths, and they successfully applied this method to the navigation of real robots [11]. Saunders et al. applied RRTs to miniature air vehicles. The method generates waypoint paths around known obstacles using RRTs and generates paths around detected obstacles using a dynamic geometric algorithm [12].

Furthermore, Wang et al. proposed a waypoint-based navigation method that combines reactive and deliberative exploration. The reactive exploration scheme generates waypoints using an incremental decision tree. This method is able to navigate unknown, changing, and time-constrained environments [13].

Most of these studies consider methods of navigating between known waypoints or generating new waypoints between already known waypoints. However, little work has focused on estimating the known waypoints, including the goal point.

In the Tsukuba Challenge, Eguchi and Mizutani proposed a sub-waypoint method [14]. This approach aims to facilitate continuous navigation and obstacle avoidance. When self-localization errors increase, the estimated waypoint positions can be located in unreachable areas. For this reason, sub-waypoints, which are dynamic target points, are set at reachable points along the course in the following two ways:

1. Regions between lined landmarks: sub-waypoints are set in line with lined landmarks.

2. Regions between the two sides of the course, which is surrounded by obstacles such as promenades or corridors: sub-waypoints are set in the middle of the course.

The sub-waypoint method is limited to the above two cases. Thus, its versatility is problem because the method cannot set sub-waypoints in other cases. Furthermore, the technique does not correct for errors in waypoint estimation, it only interpolates sub-waypoints between known waypoints to avoid obstacles.

Our proposed method is quite different from previous studies in that it estimates waypoint positions in reachable areas using external sensors such as LIDAR sensors. Furthermore, the shifts in the waypoints are assumed to be due solely to self-localization errors, and the method improves the estimation accuracy by implementing mutual feedback between the PDFs of the waypoints and the self-positions. Therefore, the problem of waypoint-based navigation is resolved.

3 MUTUAL FEEDBACK

3.1 Outline

The proposed method comprises the following 3 steps:

STEP 1: Self-localization based on a particle filter

STEP 2: Waypoints estimation based on a particle filter

STEP 3: Mutual feedback between self-positions and waypoints. Return to Step 1.

STEP 1 uses a conventional particle filter as described in references [15, 16]. External sensors used to estimate the self-position may be either LIDARs or GPSs. In the Tsukuba Challenge, we used LIDARs for external sensors because the course includes sections in which robots cannot receive GPS signals.

STEP 2 also uses a conventional particle filter. However, the waypoint motion model is discrete in contrast to the model of the robot's self-positions. The measurement model of STEP 2 is also different from that of STEP 1. Because waypoints are usually set to points where there are no physical entities, it is not easy to estimate waypoints. We devised a method that will be described in detail in Section 2.2.

STEP 3 assumes that there are no errors in the map or waypoint data, and all errors thus arise in self-localization. Therefore, the shifts in waypoint positions are due to self-localization errors. In the proposed method, the waypoint positions and self-positions are represented by probability density functions (PDFs). The shapes of the two PDFs are basically the same because the waypoint estimation errors are determined by the self-localization errors. However, constraints on the physical locations of waypoints, e.g., waypoints must be 3 [m] away from a wall or in the middle of an aisle, can constrain the PDFs of the waypoints. The reduced PDFs are fed back to the PDFs of the self-positions, and thus the self-position PDFs are also reduced. In the same way, the self-position PDFs are then fed back to the waypoint PDFs. This iterative computation can improve the accuracy the positions of the robot and the waypoints.

Obstacle avoidance and navigation are based on an artificial potential field (APF) method. In this method, waypoints exert attractive forces, while obstacles exert repulsive forces. Robots navigate to the waypoints and avoid obstacles. This method is simple and effective, but exhibits one major problem. If a local minimum exists, then robots can become trapped in this minimum and are then unable to reach waypoints. Waypoint estimation errors are one of the causes of this problem.

The proposed method can reduce the errors in waypoint and self-position estimation and can thus partially solve the major drawback of waypoint-based navigation.

3.2 Waypoint Estimation Using Particle Filter

3.2.1 Outline

This section describes the waypoint estimation method using the particle filter. The particle filter method is well-known and widely used for self-localization. The framework of the particle filter is the same as that of conventional filters [16]. However, the motion models and measurement models are quite different from those conventional filters. The particle filter represents the positions of the robot and the waypoints using a set of particles, which represent candidate positions. The i-th particle

$\boldsymbol{s}_t^{(i)} = (x_t^{(i)}, y_t^{(i)}, w_t^{(i)})$ at time t is represented by a position $(x_t^{(i)}, y_t^{(i)})$ and a weight $w_t^{(i)}$. The particle filter algorithm is presented below.

STEP 1: Initialization

Set the initial position $(x_0^{(i)}, y_0^{(i)})$ and weight $w_0^{(i)}$ of a particle $\boldsymbol{s}_0^{(i)}$ at time $t = 0$.

$$x_0^{(i)} = wp_x^1 + rand(0, {\sigma_x}^2) \tag{1}$$

$$y_0^{(i)} = wp_y^1 + rand(0, {\sigma_y}^2) \tag{2}$$

$$w_0^{(i)} = 1/N \tag{3}$$

where (wp_x^1, wp_y^1) is the first waypoint, $rand(0, \sigma^2)$ is a zero-mean random number with variance σ^2 and N is the total number of particles.

STEP 2: Sampling

Update the position $(x_t^{(i)}, y_t^{(i)})$ of a current particle $\boldsymbol{s}_t^{(i)}$ based on the motion model.

The motion model of the waypoint is represented by the following equation. The waypoint does not move sufficiently far to switch to the next waypoint.

$$x_t^{(i)} = wp_x^{next} + rand(0, {\sigma_x}^2) \tag{4}$$

$$y_t^{(i)} = wp_y^{next} + rand(0, {\sigma_y}^2) \tag{5}$$

where $(wp_x^{next}, wp_y^{next})$ is the next waypoint position.

STEP 3: Update weights

Calculate the weight $w_t^{(i)}$ of a particle $\boldsymbol{s}_t^{(i)}$ using Equation (6).

$$w_t^{(i)} = \frac{p\left(x_t^s, y_t^s \mid x_t^{(i)}, y_t^{(i)}\right)}{\sum_{j=1}^{N} p\left(x_t^s, y_t^s \mid x_t^{(j)}, y_t^{(j)}\right)} \tag{6}$$

Where (x_t^s, y_t^s) is the estimated waypoint position at time t, $p(x_t^s, y_t^s \mid x_t^{(i)}, y_t^{(i)})$ is the measurement model described in detail in Section 3.2.2.

STEP 4: Localization and resampling

The position (x_t, y_t) of the waypoint is the expected value of all particles as shown in the Equations (7) and (8).

$$x_t = \sum_{i=1}^{N} w_t^{(i)} x_t^{(i)} \quad (7)$$

$$y_t = \sum_{i=1}^{N} w_t^{(i)} y_t^{(i)} \quad (8)$$

In the resampling, low-variance sampling is used to reduce the sampling error [16]. New particles are randomly placed using a uniform distribution around the center of the estimated waypoint position (x_t^s, y_t^s) using Equations (9) and (10) . Return to STEP 2.

$$x_t^{(j)} = x_t^s + rand(0, {\sigma_x}^2) \quad (9)$$

$$y_t^{(j)} = y_t^s + rand(0, {\sigma_y}^2) \quad (10)$$

3.2.2 Measurement model

In many cases, waypoints do not correspond to physical entities so that robots can pass through the waypoints. Therefore, measurement models must be devised.

This method assumes that waypoints are near physical obstacles such as walls, buildings, or trees. In this paper, a waypoint WP_i is set $\boldsymbol{d}^i = \{d_x^{(i)}, d_y^{(i)}\}[m]$ away from obstacles.

During the course of robot navigation, self-localization errors can move waypoints inside the obstacles. LIDAR is used to examine the relationship between the waypoint and the position of the obstacles. If the waypoint is inside an obstacle as shown in Figure 2, then the PDFs of the waypoints are changed using the following procedure. The geometric constraints are often set by the presence of entities such as hedges, fences, or walls.

(1) Obtain the point clouds of the obstacles using LIDAR. Extract the two points (Obstacle 1, Obstacle 2) nearest the waypoint from the point clouds. The Obstacle 1 and Obstacle 2 are the closest points to the waypoint in the point clouds.

(2) Determine the line l_a that passes through the two points and the line l_b that passes through the waypoints and is perpendicular to the line l_a.

(3) Determine the intersection $q(x_q, y_q)$ between the lines l_a and l_b. The estimation point D(x_t^s, y_t^s) is set at a pre-defined correction distance $d[m]$ from the intersection.

——— Insert Figure 2 ————

———— Insert Figure 3 ————-

The measurement model is a normal distribution $f(x, y; \mu_x, \mu_y, \sigma_x^2, \sigma_y^2, \rho)$, which is defined as follows.

$$p(x_E, y_E | x_t^{(i)}, y_t^{(i)}) = f(x_t^{(i)}, y_t^{(i)}; x_t^s, y_t^s, \sigma_x^2, \sigma_y^2, \rho) \tag{11}$$

where (μ_x, μ_y) are the expected values, (σ_x^2, σ_y^2) are variances, and ρ is the correlation coefficient.

The weight $\omega_t^{(i)}$ of each particle $s_t^{(i)}$ is obtained using Equation (6) in this measurement model.

This procedure is easy to apply to other geometric constraints, as shown in Figure 3. The estimation point is d_x away from line l_a and d_y away from line l_c. The other procedures are the same.

3.3 Mutual Feedback

The estimation accuracy of the self-positions and waypoints is improved by mutual feedback between the PDFs of the estimated self-position and waypoint positions.

Dead reckoning errors grow as the robot moves, and errors in the PDFs of the waypoint positions similarly increases, as shown in Figure 5. However, the centers of the PDFs do not changed significantly. The authors rewrote the section 3.2.2 Measurement model. The explanation of Figure 2 and Figure 3 are added in the section. When the self-localization errors grow too large, waypoints can move into the interiors of obstacles. In this case, the relative positions of the obstacles and the waypoints can be determined using sensors, such as LIDAR sensors. This information constrains the PDFs of the waypoints because the waypoints cannot exist inside of the obstacles. That is, the PDF of the estimated waypoint is equal to the joint distribution of the newly estimated PDF A of the waypoint and the initial PDF B of the waypoint, as shown in Equation (12) and Figure 4. The PDF of the self-position is updated in the same way.

$$f(x,y;\mu_x,\mu_y,\sigma_x^2,\sigma_y^2,\rho) = f(x,y;x_w,y_w,\sigma_x^2,\sigma_y^2,\rho) \times f(x,y;x_E,y_E,\sigma_{x'}^2,\sigma_{y'}^2,\rho') \tag{12}$$

———— Insert Figure 4 ————

———— Insert Figure 5 ————

Figure 6 illustrates the entire algorithm described above. The process after sampling is different from that of the conventional particle filter. In this process, PDFs are calculated using information from sensors, the estimated PDFs of the self-position and the waypoint positions exert mutual feedback on each other. The subsequent processes are the same as those implemented in other particle filters.

———— Insert Figure 6 ————

4 EXPERIMENTS AND DISCUSSION

4.1 Outline

Experiments were conducted in the FMT (Future Machine Technology Lab.) building at Kanazawa Institute of Technology to validate of the proposed method of mutual feedback. We applied this method to navigation and examined the estimation errors. The experiment was performed 10 times using mutual feedback (Feedback ON) and 10 times with mutual feedback disabled (Feedback OFF) . Figure 7 shows the course schematic and the waypoints, and Table 1 presents each waypoint's coordinates $(x[m], y[m])$ and distance $(z[m])$.

——— Insert Figure 7 ———-

——— Insert Table 1 ———-

We used the robot shown in Figure 1, Kensei-Chan 3, which was developed based on an electric wheel chair in our laboratory. Table 2 presents the specifications of the robot. In these experiments, a map coordinate system was adopted for navigation. The robot's starting point was set to the origin $O(0,0)[m]$, as shown in Figure 7. The correction distance d was set to 1.0 $[m]$, and the current waypoint was switched to the next waypoint professional English editing service if a robot came within 0.25 $[m]$ of the current waypoint.

——— Insert Table 2 —————

4.2 Experimental Results

In 10 trials with mutual feedback enabled (Feedback ON), the robot successfully finished the course 10 times. On the other hand, in the 10 experiments without mutual feedback, the robot succeeded 7 times and failed 3 times.

Table 3 shows the mean deviation of the self-position (μ) from to the true value and the standard deviations (σ) of this error. The mutual feedback method improved the accuracy of the self-localization. The mean error improved from 1.57 $[m]$ to 1.20 $[m]$, and the standard deviation of the error improved from 0.86 $[m]$ to 0.62 $[m]$. Table 4 shows the waypoint estimation results. The mutual feedback method also improved the accuracy of the waypoint estimation. The mean error improved from 1.49 $[m]$ to 1.26 $[m]$, and the standard deviation of the error improved from 0.76 $[m]$ to 0.63 $[m]$.

Figure 8 clearly demonstrates that the estimation errors increase as the robot moves with Feedback OFF, whereas the errors do not increase over time with Feedback ON. The position estimation errors are relative large in both cases because positions were estimated using only dead-reckoning.

The proposed method of mutual feedback is effective when the self-position estimation errors are large. If the self-position estimation errors are small, the waypoints do not move into unreachable areas, and the proposed method does not affect the results.

The results of these experiments and others indicate that the proposed method is useful. Furthermore, we applied this method to waypoint-based navigation and participated in the Tsukuba Challenge 2010. Our team successfully finished a 240 [m] run using only dead reckoning and mutual feedback.

——— Insert Figure 8 ———————

——— Insert Table 3 —————————

————— Insert Table 4 ——————————-

4.3 Discussion

The proposed method of waypoint estimation using mutual feedback solves the most significant drawback of waypoint-based navigation. Waypoint-based navigation based on the map matching approach sometimes fails because errors in self-position estimation cause the positions of waypoints to move into unreachable. The proposed method can relocate waypoints into reachable areas using the distances around sensed obstacles.

The proposed method presents a number of advantages compared to map matching. The method does not require precise global maps, which are often pregenerated using LIDAR, and the algorithm is simple and easy to implement.

The disadvantage of the proposed method is that the performance depends on the shape of the surrounding environment. The waypoint estimation technique should be modified to address more complicated environment because currently, the method performs better when the waypoints are set near straight lines or right-angled corners as shown in Figures 2 and 3.

In this paper, the waypoint estimation method is only considered in the presence of static obstacles. The method is difficult to simply apply in an environment that contains dynamic obstacles such as pedestrians and vehicles. To solve this problem, dynamic obstacles and static obstacles should be distinguish, and this method should only be applied to the static obstacles. Discrimination between static obstacles and dynamic obstacles is not difficult, but, the authors have not implemented it yet. The topic of dynamic obstacles should be treated in future work.

5 CONCLUSIONS

In this paper, to solve the major problem faced by waypoint-based navigation, we proposed a method of simultaneous estimation of self-positions and waypoint positions using the particle filters.

The waypoint errors were caused by errors in self-position estimation. Waypoint errors, represented as PDFs of the waypoint positions, can be decreased using physical constraints sensed through LIDAR. The improved PDFs are then fed back to the PDFs of the self-positions, which constrains the self-position PDFs. In the same way, the constrained self-position PDFs are fed back to the waypoint position PDFs. This iterative computation improves the accuracy of both the self-positions and waypoint positions.

In the Tsukuba Challenge 2010, this method was used in a trial run in which only dead reckoning information was used for navigation, and the robot successfully finished a 240 [m] run. Other experimental runs demonstrate that this unique method is both simple to implement and useful in robot navigation. Our future work will include methods for dealing with dynamic obstacles.

REFERENCES

[1] J. Van Erp, H. Van Veen, C. Jansen, T. Dobbins, Waypoint navigation with a vibrotactile waist belt, ACM Transactions on Applied Perception, vol.2, pp.106-117 (2005)

[2] A. Maid, P. Mejia, A. Lakhotia and C. Cavanaugh, Subgoal-based local navigation and obstacle avoidance using a grid-distance field, International Journal of Vehicle Autonomous Systems, vol.4, pp.122-142 (2006)

[3] Y. Wang, D. Mulvaney, I. Sillitoe, E. Swere, Robot navigation by waypoints,, vol.52, pp.175-207 (2008)

[4] K. Kendoul, Z. Yu, K.Nonami, Embedded autopilot for accurate waypoint navigation and trajectory tracking: Application to miniature rotorcraft UAVs, IEEE International Conference on Robotics and Automation, pp.2884-2890 (2009)

[5] S. Yuta, M. Mizukawa, H. Hashimoto, H. Tashiro, T. Okubo, Tsukuba Challenge 2009 - Towards Robots Working in the Real World: Records in 2009 -, Journal of Robotics and Mechatronics, vol.23, no.2, pp.201-206 (2011)

[6] B. Krogh and C. Thorpe, Integrated path planning and dynamic steering control for autonomous vehicles, Proceedings of IEEE International Conference on Robotics and Automation (ICRA '86), pp.1664-1669 (1986)

[7] B. Krogh and D. Feng, Dynamic generation of subgoals for autonomous mobile robots using local feedback information, IEEE Transactions on Automatic Control, vol.34, no.5, pp.483-493 (1989)

[8] J. Barraquand, B. Langlois and J.Latombe, Numerical potential field techniques for robot path planning, IEEE Transactions on Systems, Man, and Cybernetics, vol.22, no.2, pp.224-241 (1992)

[9] M. Lagoudakis, Mobile Robot Local Navigation with a Polar Neural Map, Master's thesis, University of Louisiana, Lafayette (1998)

[10] S. LaValle, Rapidly-exploring random trees: A new tool for path planning, Technical Report, no.98-11 (1998)

[11] J. Bruce and M. Veloso, Real-time randomized path planning for robot navigation, IEEE/RSJ International Conference on Intelligent Robots and Systems, vol.3, pp.2383-2388 (2002)

[12] J. Saunders, B. Call, A. Curtis, R. Beard, and T. McLain, Static and Dynamic Obstacle Avoidance in Miniature Air Vehicles, Proceedings of Infotech@Aerospace, paper no. AIAA-2005-6950 (2005)

[13] Y. Wang, D. Mulvaney, I. Sillitoe, and E. Swere, Robot Navigation by Waypoints, Journal of Intelligent and Robotic Systems, vol.52, pp.175-207 (2008)

[14] J. Eguchi and K. Mizutani, Development of an autonomous robot to participate in the Tsukuba Challenge 2009, Proceedings of the Tsukuba Challenge 2009 Symposium, no.01 (2010)

[15] S. Thrun, D. Fox, W. Burgard and F. Dellaert, Robust Monte Carlo Localization for Mobile Robots, Journal of Artificial Intelligence, vol.128, no.1-2, pp.99-141 (2001)

[16] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, The MIT Press (2005)

Figure 1: Kensei-Chan 3 at the Tsukuba Challenge 2010

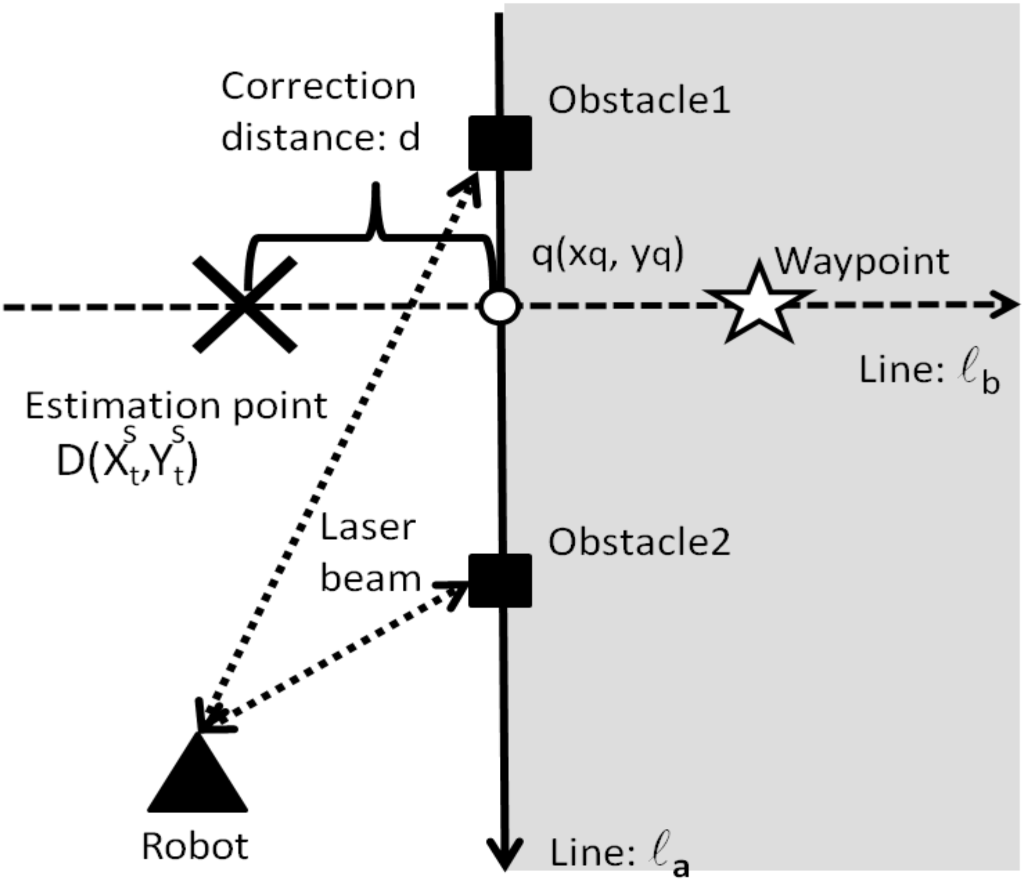

Figure 2: Estimation using geometric constraint 1

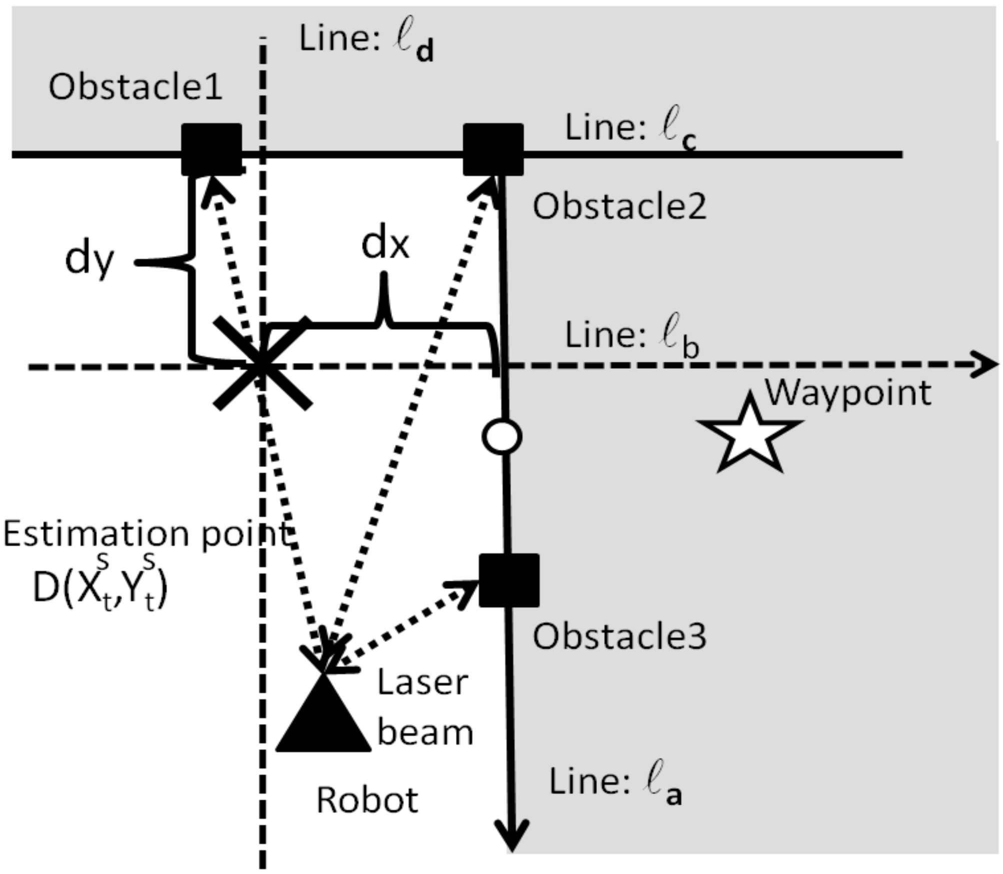

Figure 3: Estimation using geometric constraint 2

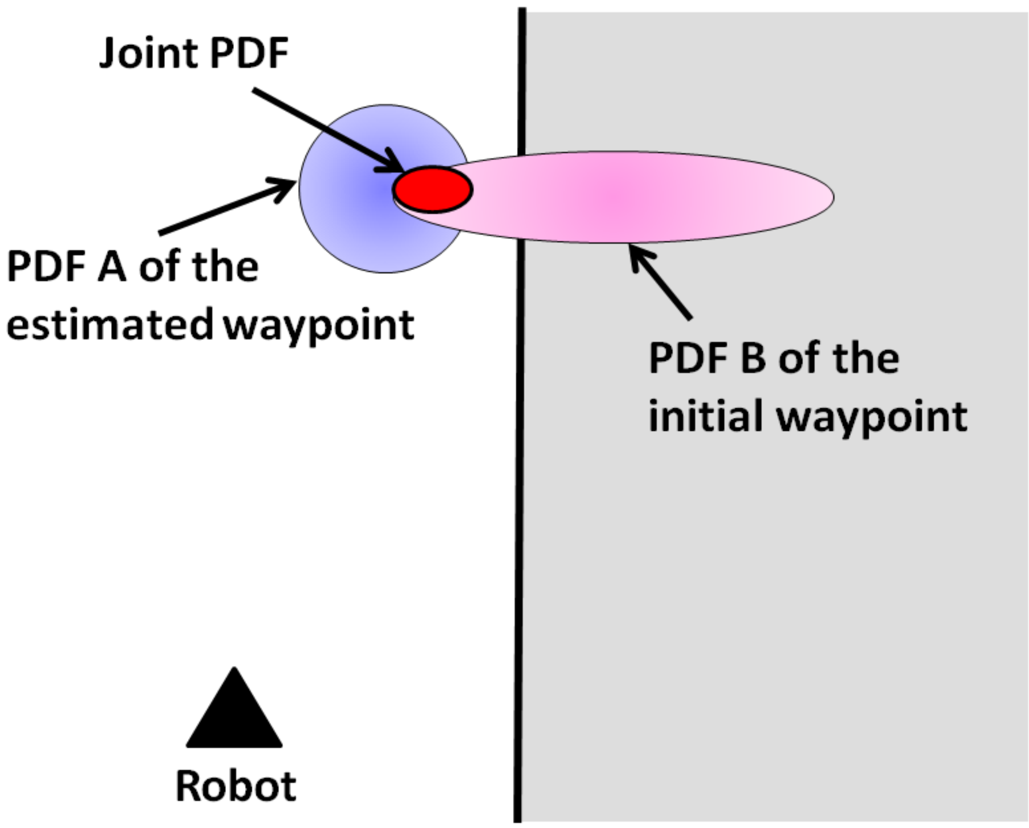

Figure 4: Joint PDFs of waypoints

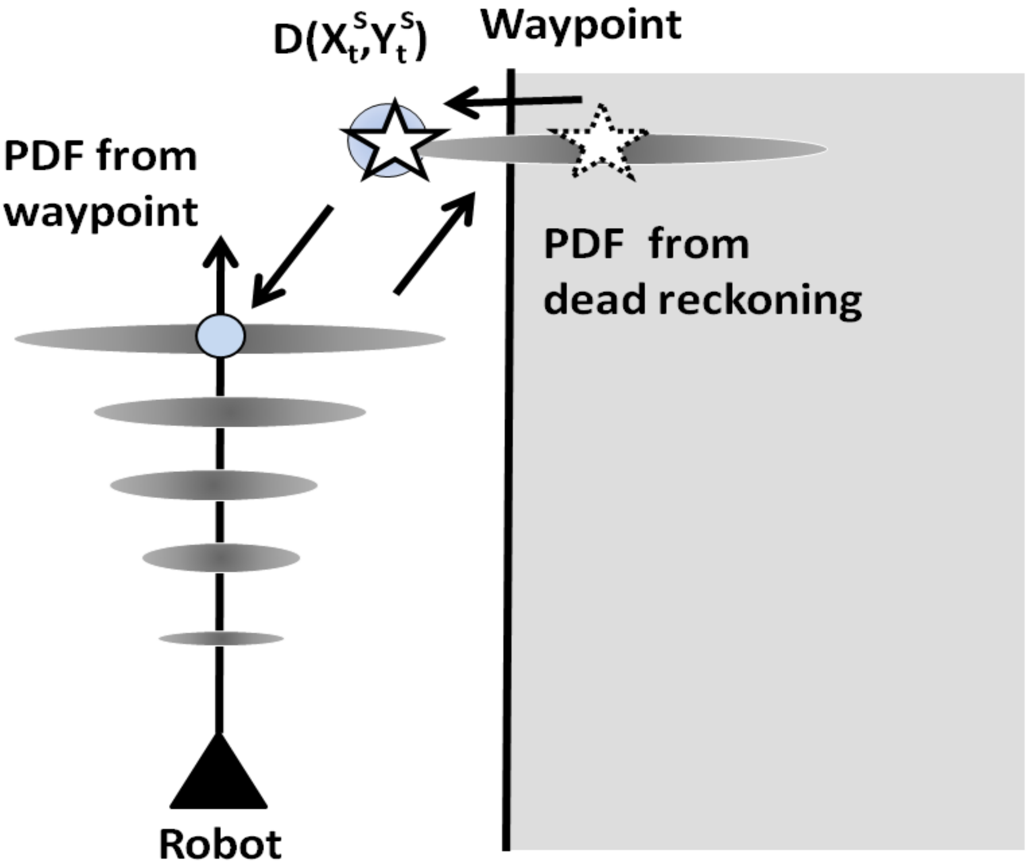

Figure 5: Mutual feedback

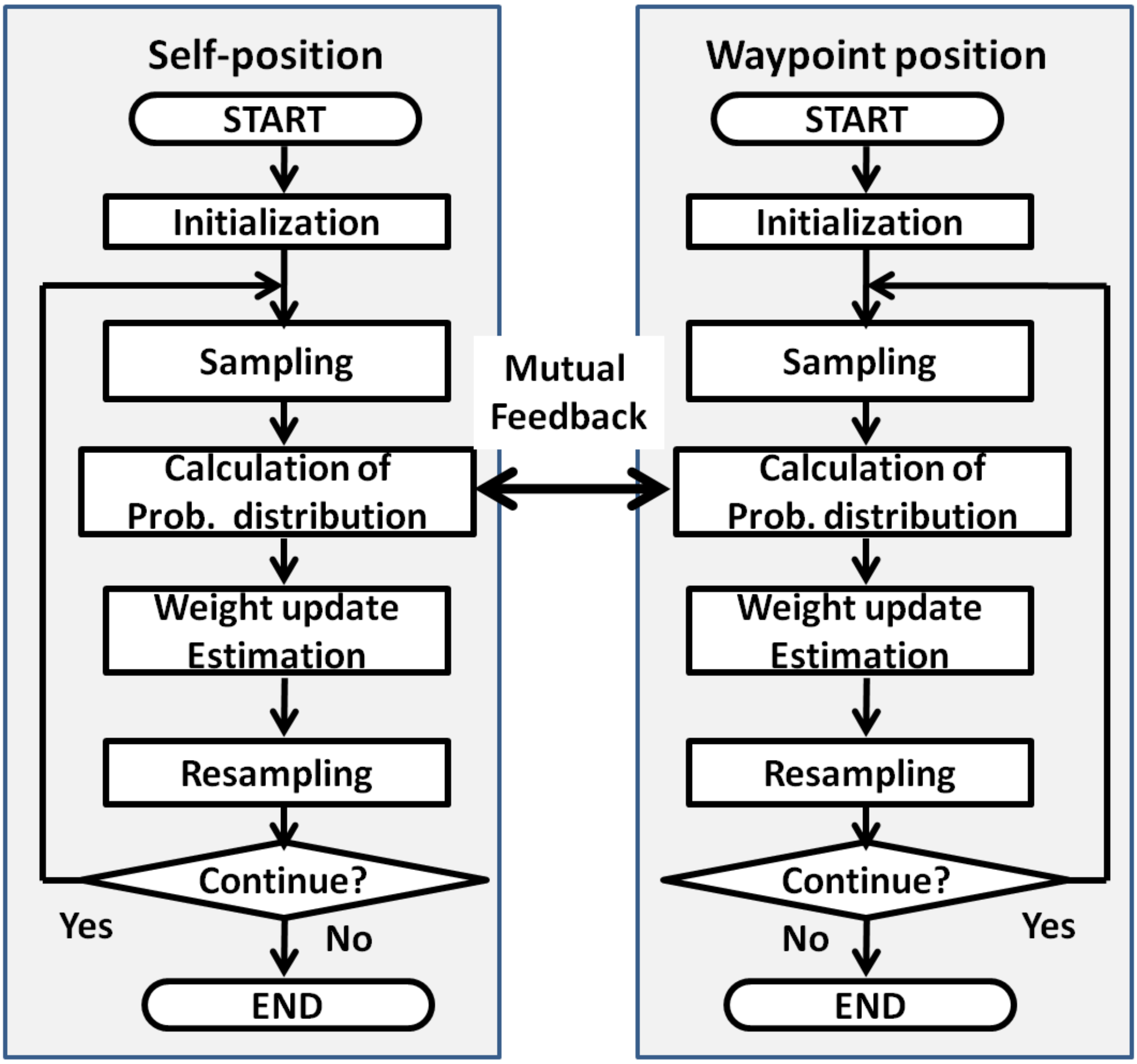

Figure 6: The algorithm of the proposed method

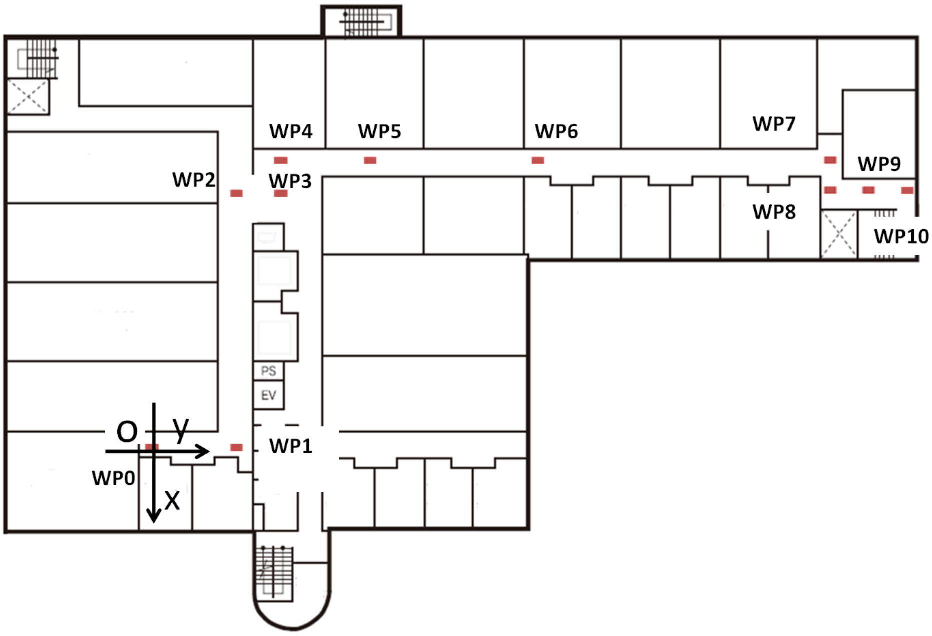

Figure 7: The experiment course

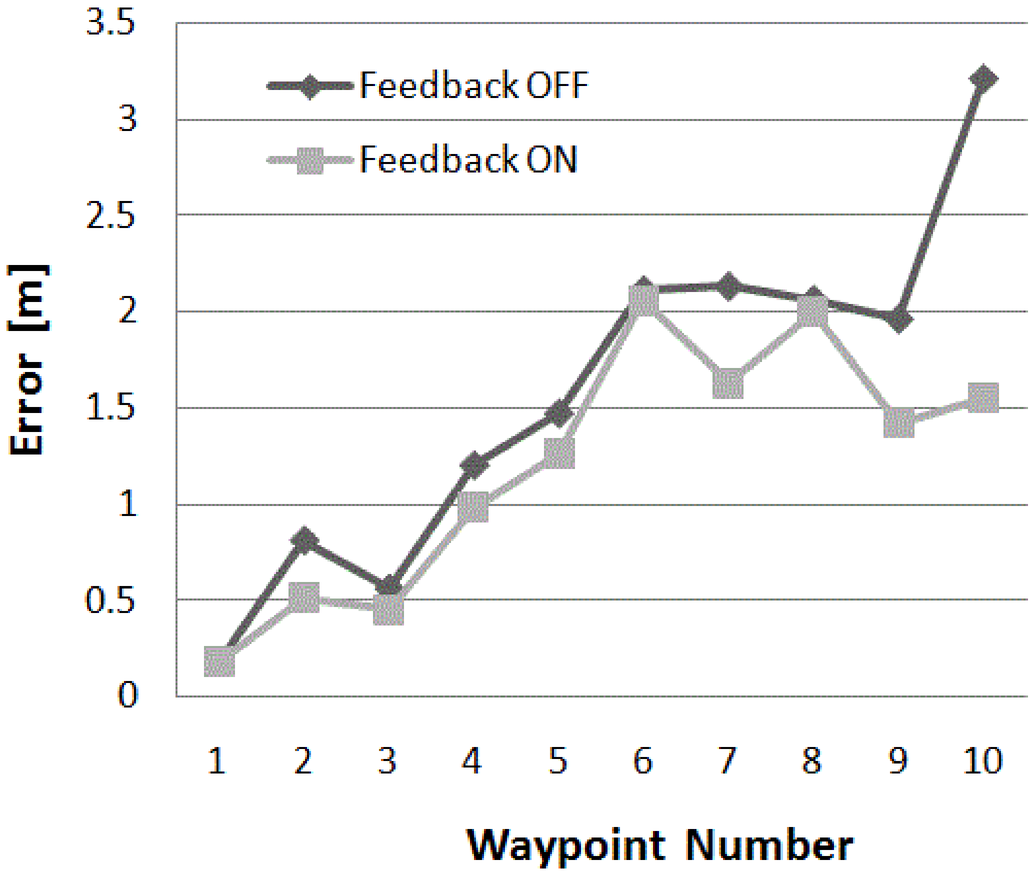

Figure 8: Localization Errors

Table 1: Waypoints

No	x[m]	y [m]	z[m]
1	0.00	7.53	7.53
2	-21.35	7.53	22.64
3	-21.35	11.53	24.26
4	-24.09	11.53	26.71
5	-24.09	19.53	31.01
6	-24.09	34.53	42.10
7	-24.09	59.53	64.22
8	-21.59	59.53	63.32
9	-21.59	62.53	66.15
10	-21.59	67.53	70.90

Table 2: Specifications of the robot

Robot name	Kensei-Chan 3
Size: length $\times$ width $\times$ height	113 $\times$ 64 $\times$ 149 [cm]
Weight	75 [kg]
Max speed	1.1 [m/s]
Platform	YAMAHA JW-Active
Laptop Computer	Lenovo ThinkPad T410
CPU	Intel Core i7 M620
Memory	4GB
OS	Linux Ubuntu 10.04
LIDAR	HOKUYO UTM-30LX
Fiber Optic Gyro	JAE JG-108FD1

Table 3: Localization Errors

	Feedback OFF		Feedback ON	
Waypoint No.	μ [m]	σ[m]	μ [m]	σ[m]
1	0.18	0.14	0.18	0.08
2	0.81	0.46	0.51	0.21
3	0.56	0.37	0.45	0.26
4	1.20	0.49	0.98	0.24
5	1.47	0.65	1.26	0.34
6	2.11	0.98	2.06	0.58
7	2.13	1.30	1.63	0.56
8	2.06	0.59	2.00	0.31
9	1.96	1.36	1.42	0.43
10	3.21	3.27	1.55	0.62
Average	1.57	0.86	1.20	0.62

Table 4: Waypoint Estimation Errors

	Feedback OFF		Feedback ON	
Waypoint No.	μ [m]	σ[m]	μ [m]	σ[m]
1	0.32	0.10	0.30	0.12
2	0.63	0.37	0.45	0.17
3	0.69	0.41	0.52	0.30
4	1.20	0.40	1.05	0.24
5	1.61	0.69	1.32	0.43
6	2.22	1.02	2.22	0.60
7	1.81	0.84	1.55	0.48
8	1.99	0.61	1.96	0.35
9	1.50	0.73	1.40	0.45
10	2.92	2.41	1.79	0.68
Average	1.49	0.76	1.26	0.63