2005 American Control Conference
June 8-10, 2005. Portland, OR, USA

Uniform Clustered Particle Filtering for Robot Localization

Tun Yang, and Victor Aitken
Department of Systems and Computer Engineering,
Carleton University,
1125 Colonel By Drive, Ottawa, Ontario, K1S 5B6, Canada
tunyang@sce.carleton.ca, vaitken@sce.carleton.ca

Abstract— **Localization is a fundamental ability for an autonomous mobile robot. Different particle filter based solutions to the problem are simulated in a software simulator, and the results are compared and discussed. The weighted bootstrap filter, clustering particle filter, and the uniform particle filter are examined. A new method based on the clustered particle filter and the uniform particle filter, the uniform clustering particle filter, is also proposed and evaluated.**

I. INTRODUCTION

Autonomous mobile robots must act intelligently without external control. To achieve a level of intelligence, a robot must have some capacity in reasoning about its environment given mounted sensors and actuators. Some regard robot localization as "the most fundamental problem to providing a mobile robot with autonomous capabilities" [1].

Succinctly defined, localization is the act of estimating the location of a robot in its environment, given that the robot has an approximate map of its surrounding environment. There has been much research into this problem, and localization may be divided into more specific categories: local position tracking, global localization, and relocalization.

Local position tracking assumes that the initial location of the robot is relatively well known. Localization methods that solve the tracking problem continue to provide good estimations to robot location as the robot moves through the environment despite uncertain motion and error filled sensor observations.

The global localization problem assumes that the initial location of the robot is unknown. Localization methods that solve this issue typically need to deal with parts of the environment that are indistinguishable from other parts of the environment due to structural similarity.

The relocalization problem occurs when a robot may be certain of its location, but given further observations and motion, realizes that it is completely lost. This is the case when a localization method estimates the incorrect robot location, which may be due to environment similarities. This is also an issue when a robot is moved by external intervention, such as a referee in a competition. The relocalization problem is also known as the *kidnapped robot problem*.

The family of methods known as Monte Carlo localization (MCL) combines the use of particle filtering methods, and robot sensor and action models to solve all three categories of robot localization. Also, MCL is easily implemented, as compared to other localization methods. It is due to these characteristics that MCL enjoys widespread popularity in recent times. First introduced by Dellaert, Burgard, Fox, and Thrun [2], the original MCL method has been subsequently modified, augmented, and implemented by other researchers including [3], [4], [5].

In this paper, we propose to modify the standard MCL solution by changing the particle filter used. The new MCL solution proposed is based on the Uniform MCL (U-MCL) [4], and the Clustered Particle Filtering MCL (CPF-MCL) [5]. The resulting MCL solution retains the benefits of computational efficiency and ease of implementation from the U-MCL, while inheriting its robustness in global localization from the CPF-MCL method. The MCL method is therefore named the Uniform Clustered Particle Filtering MCL (UCPF-MCL), and is a good solution to the localization problems with improvements over the standard MCL method.

II. BACKGROUND

Bayesian state estimation is the basis of successful robot localization techniques such as the Kalman filter [6], Markov localization [7], and Monte Carlo localization (MCL) [2]. Bayesian state estimation is so named for its use of Bayes' theorem to incorporate new evidence from robot sensors to condition the current estimated state.

For robot localization, the quantities x, y, and θ are of interest since they describe the position and orientation of the robot in the environment. Let the state of the robot at time t be $s_t = [\ x \quad y \quad \theta\ ]$. We are interested in estimating the state based on general events $d_{0\ldots t}$, defined as

$$d_{0\ldots t} = o_0 a_0 o_1 a_1 \ldots o_{t-1} a_{t-1} o_t,$$

where o are sensor observations, and a are robot actions. The events of interest occur in alternating order following the paradigm of a sense-plan-act architecture that is common in robotics. Note that the sequence ends in an observation reading due to the fact that localization occurs as part of the planning process when o_t is available, but a_t has yet to occur.

With these definitions, the probability density distribution of the state of the robot is given by

$$Bel(s_t) = p(s_t | o_0 a_0 o_1 a_1 \ldots o_t). \tag{1}$$

Supported by the National Science and Engineering Council of Canada

This conditional distribution is a general definition that is fundamental to all Bayesian filters, and the notation $Bel(s_t)$ emphasizes that the conditional distribution is the robot's *belief* of its location. Using the theorem of total probability,

$$p(x) = \int p(x|y)p(y)dy,$$

Bayes' theorem,

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)},$$

and the Markov assumption,

$$p(o_t|d_{0\ldots t}s_t) = p(o_t|s_t),$$

a recursive estimation equation can be derived from (1) to be

$$Bel(s_t) = \eta p(o_t|s_t) \int p(s_t|a_{t-1}s_{t-1})Bel(s_{t-1})ds_{t-1}, \tag{2}$$

where η is a normalizing constant.

In MCL, $p(o_t|s_t)$ is called the sensor model, while $p(s_t|a_{t-1}s_{t-1})$ is called the action model. Sensors and actions with such models defined can be used as part of the MCL process. These continuous probability density functions are approximated using discrete samples called *particles*, and the algorithm to update the state estimate is called a particle filter.

Particle filters stem from developments in Monte Carlo sampling, importance sampling, sequential importance sampling, and sampling importance resampling. A standard particle filter is the weighted bootstrap filter [8], and the algorithm with reference to its application towards robot localization is given in Table I. Using terms defined in sampling importance resampling, the proposal distribution used in the algorithm is given by

$$p(s_t|s_{t-1}a_{t-1})Bel(s_{t-1}),$$

and the weight w of each particle i is given by

$$w_t^{(i)} = p(o_t|s_t^{(i)}).$$

With these settings, the particle filter approximates (2).

TABLE I
WEIGHTED BOOTSTRAP FILTER ALGORITHM

N= number of samples used to approximate probability density distribution
S= set of particles representing a continuous probability density distribution of robot state
w_i= set of weights for each particle i, where $i = 1 \ldots N$
for i = 0 to N do
sample random s from S according to $w_1, \ldots, w_N$
sample random $s^{'} \sim p(s^{'}|a_{t-1}, s)$
$w^{'} = p(o_t^{1\cdots m}|s^{'})$ for different observations 1 …m
add $\left(s^{'}, w^{'}\right)$ to $S^{'}$
endfor
Set S=$S^{'}$

A. Clustered Particle Filtering MCL

It is observed in [5] that the standard particle filtering technique sometimes incorrectly converges to a unimodal distribution. For robot localization, this occurs in the case where there are *similar* locations in the robot's environment. Similar locations are indistinguishable from each other given the sensor observations. Generally, however, such locations are distinguishable given some amount of robot motion. Milstein *et al* observed that standard particle filters are unable to maintain multimodal belief distributions that contain indistinguishable locations. Such particle filters settle prematurely to an arbitrary unimodal belief distribution before distinctive sensor observations arrive to correctly update the location estimate.

Milstein *et al* proceed to develop the clustered particle filter for use in the MCL process. Developments in [5] show that the idea of clustering spatially similar particles together is mathematically sound, and solves the problem of a premature collapse to a unimodal belief. The clustered particle filter makes a number of assumptions:

- there are K clusters
- every particle is assigned to a single cluster. Let $c(s_t^{(i)})$ be a function that returns the cluster number that a particle $s_t^{(i)}$ belongs to
- $f(s_t)$ gives the same value for each particle that is in the same cluster
- the cumulative weight over all particles in a cluster is equal to the cumulative weight in all other clusters:

$$\sum_{\substack{s_t^{(i)} \in S_t \\ c(s_t^{(i)})=k}} f(s_t^{(i)})p(s_t^{(i)}|d_{0\ldots t}) = \sum_{\substack{s_t^{(i)} \in S_t \\ c(s_t^{(i)})=k'}} f(s_t^{(i)})p(s_t^{(i)}|d_{0\ldots t})$$

- a particle, once assigned to a cluster, cannot change to another cluster.

With these assumptions, an algorithm for a clustered particle filter is given in Table II. The algorithm performs well in environments with similar locations, and some results are independently reproduced in [9].

B. Uniform MCL

Typical mobile robots have many components that vie for computational power. As a result, computational resources are scarce and algorithms that require real-time response need to be aware of these limitations. In [4], Ueda *et al* make modifications to the standard MCL solution to reduce computational requirements, while maintaining performance. Their motivation was a localization algorithm that runs successfully on the Aibo robot dog for Robocup.

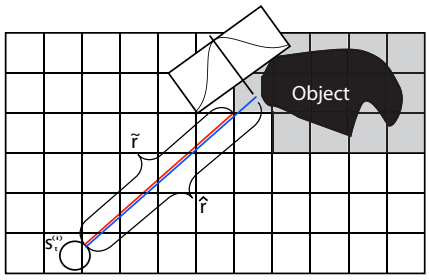
Ueda *et al* made the following key changes to the standard MCL solution:

- action model follows a uniform distribution
- sensor model follows a uniform distribution
- particles are evolved rather than re-sampled
- new particles are sampled from old ones as necessary

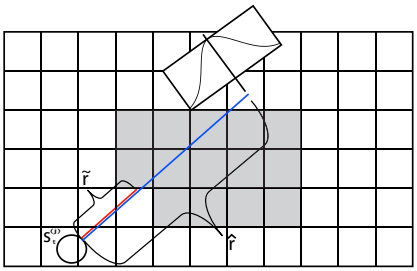
TABLE II

CLUSTERED PARTICLE FILTER ALGORITHM

Initialization
1) seed particle filter with initial seed distribution 2) iterate several steps through the particle filter 3) cluster particles based on spatial similarity 4) create a weighted bootstrap particle filter for each cluster
Normal operation
In normal operation of the CPFMCL after clustering, • update each cluster's particle filter separately • combine clusters that are too similar to each other since different clusters may have evolved to be equivalent
Additional
A separate weighted bootstrap filter is run occasionally in the background to find likely clusters of particles that may not have been chosen by the initial seeding process due to the use of insufficient particles, or inappropriate seeding process. It also allows for the solution of the kidnapped robot problem.



(a) Possible candidate state given high weight



(b) Possible candidate state given low weight

Fig. 1. Laser Range Measurement Scenario for Localisation

It is argued that the motion model for Aibo is well represented by a uniform distribution. Since the robot is a legged robot, the same command for the robot to move forward does not always result in a consistent amount of distance traveled. Furthermore, the robot is prone to collisions with obstacles in the environment, or even with other robots. The distance that the Aibo travels with one step forward is therefore well modeled by the uniform distribution

$$f(x) = \frac{1}{d},$$

where d is the maximum distance traveled when there are no obstructions.

Suppose laser range sensors provide sensor readings to a robot, and the plausibility of the range readings for a specific particle follows that of a Gaussian distribution. In Fig 1(a), the measured range $\hat{r}$ is similar enough to the expected reading $\tilde{r}$ in accordance to the Gaussian distribution, and the specified particle has relatively high weight. In Fig 1(b), however, $\hat{r}$ is not close to $\tilde{r}$, and the corresponding particle is assigned negligent weight. A particle with high weight is expected to be re-selected as part of the sampling process, while a particle with low weight is not expected to be resampled, and is discarded.

Intuitively, however, the sensor model merely conditions the weight of the particles so that likely particles are kept as part of the resampling process, while unlikely particles are discarded. As is demonstrated, likely particles are particles for which readings are within a distance ϵ of the expected distance $\tilde{r}$. Therefore, it is argued that weights for particles may be set based on a discrete uniform distribution. The sensor model is therefore

$$f(x) = \begin{cases} \frac{1}{2\epsilon} & |\tilde{r} - \hat{r}| < \epsilon \\ 0 & \text{otherwise} \end{cases}.$$

Given a simplified sensor model, the normal resampling process of MCL is no longer useful. After an update procedure, particles in the filter will either have zero weight, or a non-zero weight. The particles with zero weight are unlikely particles that are removed from the filter, while the particles with non-zero weight will have a numerically equal weight, and are all kept. In U-MCL, resampling only occurs in the event where there are less particles than desired. To generate new particles, Ueda *et al* inject a mid-point particle from two random particles selected from the filter.

An example of the update procedure is given in the following steps:

1) The U-MCL filter is set to have N particles that approximate the robot's starting position
2) The filter updates the particles based on the robot's actions
3) The filter removes particles with zero weight, leaving n particles in the filter
4) If $n < N$, $N - n$ particles are injected into the filter using the aforementioned re-sampling procedure

These various changes minimize the amount of computation needed by:

- simplifying the probability distributions used
- having a simple re-sampling process on a "as-needed" basis.

Results given in [4] show that U-MCL performs as well as another variant, the sensor resetting localization (SRL) [3] method.

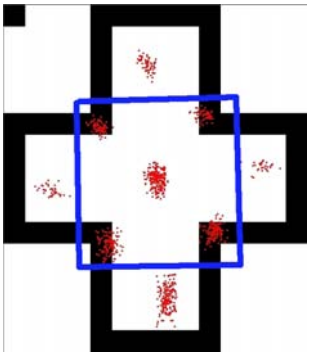


Fig. 2. U-MCL Failing in a General Environment

III. Uniform Clustered Particle Filter MCL

The robustness of CPF-MCL is great for localization in a general structured indoor office environment. Such environments typically have many similar locations, such as indistinct rooms, and being able maintain multimodal belief of the robot's location is useful for faster convergence to the correct robot state.

The computational simplicity of U-MCL is attractive for mobile robots. U-MCL as formulated, however, does not work in a general environment. U-MCL is formulated for the environment used for the Robocup Sony Legged Robot league. In that environment, the robot can distinguish its location to arrive at a unimodal belief distribution; that is to say, all locations within the environment are distinct. In a general environment, however, it is possible that there are many locations in the environment that are indistinguishable from each other leading to a multimodal belief distribution. This belief remains multimodal until new sensor observations and motion refine the belief to be the correct unimodal belief. The U-MCL resampling process does not account for multimodal belief distributions. Fig 2 exhibits the failure of the resampling process in an environment with indistinguishable locations. The robot starts off in a corridor, and the algorithm correctly identifies locations that the robot may exist. The particles that are proposed by the U-MCL resampling process, however, are enclosed by the rectangular marking. These particles accentuate that the resampling process in a multimodal belief distribution does not propose good particles. This problem is identified in [4], and indicates that a different resampling process is needed for a general environment.

It is expected that a resampling process that works in a general environment will increase the computational complexity. The idea of the proposed solution is use the clustering concept of CPF-MCL to reduce the multimodal belief distribution to multiple unimodal distributions that is solvable using the standard U-MCL solution, exhibiting the characteristics of a divide-and-conquer algorithm. The proposed solution is therefore named the Uniform Clustered Particle Filtering MCL (UCPF-MCL). This proposed algorithm combines the advantages of robustness in general environments and computational simplicity. The algorithm is outlined in Table III.

TABLE III
Uniform Clustered Particle Filter Algorithm

Initialisation
1) seed U-MCL filter with initial seed distribution 2) iterate several steps through the particle filter 3) cluster particles based on spatial similarity 4) create a U-MCL particle filter for each cluster
Normal operation
In normal operation of the CPFMCL after clustering, • update each cluster's particle filter separately • combine clusters that are too similar to each other since different clusters may have evolved to be equivalent
Additional
Relocalize the filter if all clusters are considered improbable, and eliminated. First attempt to locally resample from the most likely cluster to attempt and recover from losing track of the robot location. If the local resampling fails, then the filter is re-initialised with the uniform distribution.

IV. Evaluation

All the localization methods described in this paper are implemented and evaluated in a software simulator. This section describes the robot modeled for the experiments, and the virtual test environments.

A. Robot

The robot modeled for the experiments is based on a robot built for a robotics competition, shown in Fig 3 and Fig 4. It is equipped with laser range sensors mounted at strategic locations on the robot, and provides the robot with sparse sensor observations.

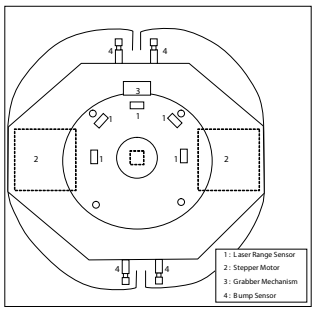


Fig. 3. Blueprint of Robot

Fig. 4. Picture of Realistic Robot

B. Test Environments

There are two test environments that are used for the experiments in this section. The first environment is a small scale symmetric environment. The environment, and the

corresponding path that the robot takes through the environment are shown in Fig 5. The symmetric environment is composed of four corridors labeled as left, right, top, and bottom corridors. It should be noted that the top and bottom corridors are longer than the left and right corridors. The second environment is a larger scale office-like environment. Shown in Fig 6 is the environment with the corresponding path the robot takes through the environment. The environment has many similar areas, such as the square rooms on the left hand edge of the environment.

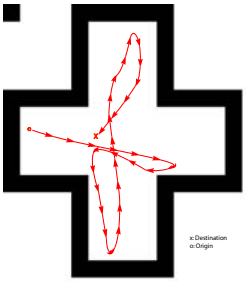


Fig. 5. Symmetric Environment

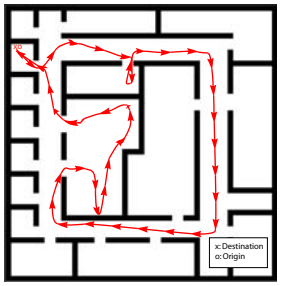


Fig. 6. Office-like Environment

V. Results

Two types of experiments were conducted. The first compares the performance of the localization methods in the symmetric test environment, while the second compares the methods' performance in a more realistic environment.

A. Symmetric Test

A series of test runs were conducted for the robot in the symmetric test environment varying the level of sensor and motion noise. If the robot is at any of the marked locations in Fig 7, then the ideal result of a localization method is to believe that the robot could be at any of the marked locations. If the symmetric test environment was both horizontally and vertically symmetric, then the ideal result is to maintain all marked locations indefinitely. The top and bottom corridors of the environment, however, are longer than the other two corridors. Therefore, given the proper robot motion, the ideal result of the localization methods is to maintain only two possible locations of the robot as shown in Fig 8. A number of effects are observed for each method, and are discussed below.

1) MCL: The problem that motivated the development of CPF-MCL arose as expected. Although the MCL method is able to initially represent a multimodal belief distribution, it is unable to maintain it. The MCL method tends to settle to a unimodal belief distribution rather quickly, even though there are no distinctive environment features that warrant such a refinement. The resulting unimodal belief does not necessarily represent the robot's true location. This is true regardless of the number of samples used.

TABLE IV
Test Case Settings for Repeated Runs

Test Set	Sample Size	Sensor Noise	Motion Noise
R1-1	10,000	10%	10%
R1-2		50%	50%
R2-1	100,000	10%	10%
R2-2		50%	50%
R3-1	500,000	10%	10%
R3-2		50%	50%

2) CPF-MCL: The CPF-MCL method deals well with the symmetric environment, and is able to find and maintain clusters of particles that represent the robot's true location. At the end of the robot's test runs with varying amounts of noise, the robot tends to maintain all four modes of the belief distribution. The ideal result, however, is to maintain only two modes.

3) UCPF-MCL: The UCPF-MCL presents the best results of the methods compared in this environment when tested under the same conditions. UCPF-MCL typically ended up with the two ideal modes, and in some cases ended up with only one of the ideal modes.

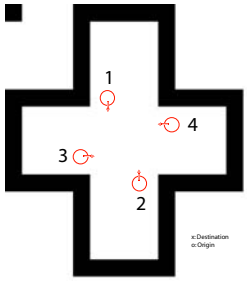


Fig. 7. Similar Locations

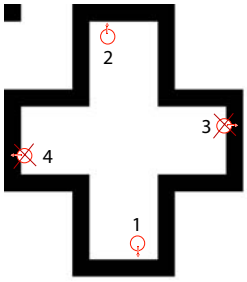


Fig. 8. Distinguished Similar Locations

B. Office-Like Test

The test cases shown in Table IV are conducted for the simulated robot in the office-like test environment. Each test case varies the level of sensor and motion noise, and is executed 50 times each to collect statistics.

As discussed earlier, similarities in the environment reduce the performance of some localization methods. In the office-like map, such similarities exist in the beginning of the robot's path since it starts in a square room, and is advantageous to methods that account for similarities.

Looking at the success rates yielded by the methods (Fig 9, Fig 10) insinuates that both the CPF-MCL and the UCPF-MCL perform better than MCL. With 10% motion noise, the UCPF-MCL outperforms the other methods. With 50% motion noise, the performance of UCPF-MCL is similar to that of CPF-MCL, although the CPF-MCL performs marginally better. Therefore, UCPF-MCL's performance would appear to suffer as the amount of noise is increased. Closer examination indicates that this effect is due to a mismatch in the action model used for UCPF-MCL, and

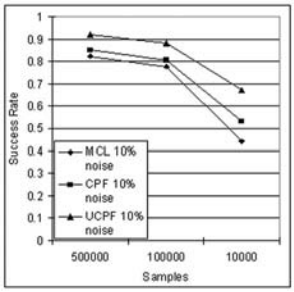


Fig. 9. Sample Size vs Success of Repeated Runs 10% noise

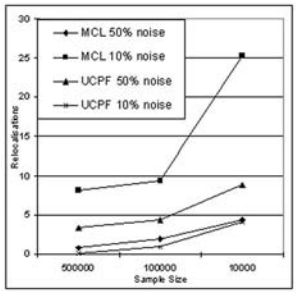


Fig. 10. Sample Size vs Success of Repeated Runs 50% noise

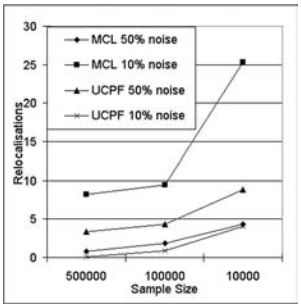


Fig. 11. Sample Size vs Relocalisation

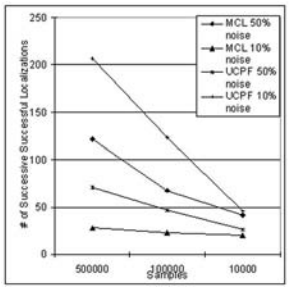


Fig. 12. Avg Longest Run vs Sample Size

the simulated robot movement model. The odometry sensor model employed by UCPF-MCL uses a uniform distribution that assumes that the robot does not travel further than its odometry sensor reading. This is not a bad assumption, since a real odometry measurement usually over estimates the actual motion of the robot due to wheel slip and collisions. The simulator, however, is implemented such that a zero mean Gaussian noise process of variable variance is added to the ideal projected movement distance. This implies that simulated actual movement may be greater than the odometry distance, and in such a case, the action model employed by UCPF-MCL does not suggest any samples in the correct area, and needs to relocalize. Eliminating this mismatch will yield better success rates.

Another statistic of interest is the average number of relocalizations required for MCL and UCPF-MCL. As seen in Fig 11, MCL required less relocalizations as the noise level increased. This effect is documented in [10]. It is advantageous to use the most accurate sensors possible, but the sensor and action models cannot be chosen to be too accurate. Accurate models are unable to robustly inject samples into the filter that counteract effects of non-ideal Monte Carlo sampling and inaccurate localization. UCPF-MCL required more relocalizations as the noise increased, and is again attributed towards the action model mismatched mentioned earlier. The effect of the mismatch is visible with higher amounts of noise, and improving the mismatched model will improve these results. One interesting result is that although the number of relocalizations increases with noise, UCPF-MCL still maintains success rates comparable to the other methods. This indicates that UCPF-MCL is both quick and successful in relocalization.

Lastly, the number of continuous successful localization attempts for the methods are shown in Fig 12. This statistic indicates the ability of the different methods to track the robot under noisy motion. As can be seen, UCPF-MCL consistently produces longer successive correct localization attempts as compared to MCL at the same noise level.

VI. Conclusions and Future Work

Various experiments were performed to compare different particle filtering solutions to the robot localization problem. Results show that MCL does not deal well with similar locations in a general environment, while both the CPF-MCL and the UCPF-MCL perform well in such environments. The UCPF-MCL method appears to perform best because it can eliminate certain similar locations that are distinguishable given further robot observations and actions.

Further experimentation indicates that all methods tested are reasonably successful, although UCPF-MCL tends to perform better under certain conditions with respect to success rate, and length of successive correct localizations.

Given the preliminary simulated results, the newly proposed UCPF-MCL method appears to be a satisfactory solution, with decreased computational requirements. Further work is needed to test the method in a real robot, as well as augmenting the solution with a mapping component to solve the simultaneous localization and mapping (SLAM) problem.

References

[1] J. Gutmann and D. Fox, "An experimental comparison of localization methods continued," in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 1, 2002, pp. 454–459.

[2] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo localization for mobile robots," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, vol. 2, 1999, pp. 1322–1328.

[3] S. Lenser and M. Veloso, "Sensor resetting localization for poorly modelled mobile robots," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, vol. 2, 2000, pp. 1225–1232.

[4] R. Ueda, T. Fukase, Y. Kobayashi, T. Arai, H. Yuasa, and J. Ota, "Uniform Monte Carlo localization - fast and robust self-localization method for mobile robots," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, vol. 2, 2002, pp. 1353–1358.

[5] A. Milstein, J. N. Sánchez, and E. T. Williamson, "Robust global localization using clustered particle filtering," in *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, Edmonton, Alberta, Canada, July 2002.

[6] N. Ayache and O. D. Faugeras, "Building, registrating and fusing noisy visual maps," in *Int. Journal Robot. Res.*, vol. 7, 1988, pp. 804–819.

[7] R. Simmons and S. Koenig, "Probabilistic robot navigation in partially observable environments," in *Proc. of the Int. Joint Conf. on Artificial Intelligence*, 1995.

[8] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *IEE Proc.-F*, vol. 140, 1993, pp. 107–113.

[9] T. Yang, "Mapping and localisation for mobile robots," Master's thesis, Systems and Computer Engineering, Carleton University, Ottawa, Canada, Aug. 2004.

[10] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," in *Artificial Intelligence Journal*, 2001.