



# Automatic Mutual Localization of Swarm Robot Using a Particle Filter

Yang-Weon Lee\*, Member, KIICE

Department of Information and Communication Engineering, Honam University, Gwangju 506-714, Korea

## Abstract

This paper describes an implementation of automatic mutual localization of swarm robots using a particle filter. Each robot determines the location of the other robots using wireless sensors. The measured data will be used for determination of the movement method of the robot itself. It also affects the other robots' self-arrangement into formations such as circles and lines. We discuss the problem of a circle formation enclosing a target that moves. This method is the solution for enclosing an invader in a circle formation based on mutual localization of the multi-robot without infrastructure. We use trilateration, which does require knowing the value of the coordinates of the reference points. Therefore, specifying the enclosure point based on the number of robots and their relative positions in the coordinate system. A particle filter is used to improve the accuracy of the robot's location. The particle filter is operates better for mutual location of robots than any other estimation algorithms. Through the experiments, we show that the proposed scheme is stable and works well in real environments.

**Index Terms:** Particle filter, Swarm robot, Tracking

## I. INTRODUCTION

Swarm robotics is an approach to robotics that emphasizes the use of many simple robots instead of a single complex robot. A robot swarm has much in common with an ant colony or swarm of bees. No individual in the group is very intelligent or complex, but combined, they can perform difficult tasks. Swarm robotics has been an experimental field, but many practical applications have been proposed. A traditional robot often needs complex components and significant computer processing power to accomplish its assigned tasks. In swarm robotics, each robot is relatively simple and inexpensive. As a group, these simple machines cooperate to perform advanced tasks that otherwise would require a more powerful, more expensive robot. Using many simple robots has other advantages as

well. Robot swarms have high fault tolerance, meaning that they still will perform well if some of the individual units malfunction or are destroyed. Swarms also are scalable, so the size of the swarm can be increased or decreased as needed.

Before a project using multiple robots is begun, the types of robots that will be used should be identified and classified because this will affect numerous elements such as the control algorithm, obstacle avoidance, communication. To accomplish given specific tasks, the system is concerned with the ability to cooperate, as shown in Fig. 1.

Mobile robot localization is a problem of determining the posture of a robot relative to a given map of an environment, and establishing correspondence between the map coordinate system and the robot's local coordinate system.

Received 07 August 2012, Revised 29 August 2012, Accepted 12 September 2012

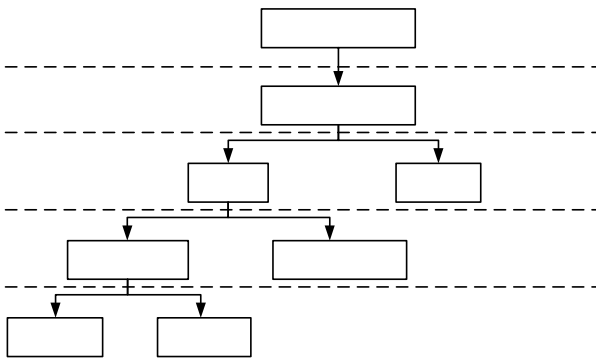
\*Corresponding Author E-mail: [ywlee@honam.ac.kr](mailto:ywlee@honam.ac.kr)

**Open Access** <http://dx.doi.org/10.6109/jicce.2012.10.4.390>

print ISSN: 2234-5973 online ISSN: 2234-8883

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

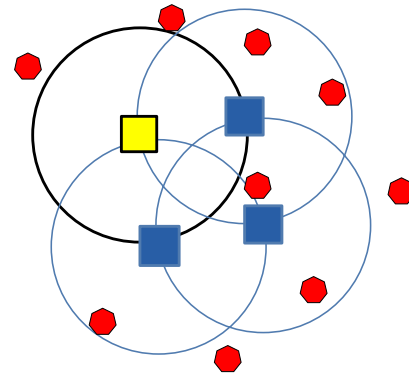
Copyright © The Korea Institute of Information and Communication Engineering



**Fig. 1.** Taxonomy of multi-robot system.

A Bayesian filter is a general method applicable to this problem that recursively estimates the robot's belief about its current position. An extended Kalman Filter [1], an estimation algorithm adopting the Bayesian filter as its foundation, provides an efficient method for representing the positional belief of a robot with a Gaussian distribution. Many robotic applications require that robots work in collaboration [2] in order to perform a certain task [3].

In this paper, we present a particle filter localization system for multiple cooperative robots in an environment where they can move automatically as a group and extract position information in a given map of the environment from stationary landmarks. For assessment of the system, we consider a scenario where some robots cannot detect landmarks. In this case, uncertainty in position estimates for all robots is expected to increase. We suppose that each mobile robot in the environment not only can detect and calculate the distance to the other robots, but also can share its position information with the other robots. Consequently, each mobile robot acts as an additional mobile landmark. However, this can introduce some degree of error to the position information from stationary landmarks, and localization could become unstable. Still, we desire self-localization performed by each robot to be improved by extra position information from the other robots. For example, in Fig. 2, the localization error will be high when a robot cannot detect landmarks. However, if a robot can detect other robots as new landmarks and utilize their position information, the robot may see increased opportunities for better localization. When a robot determines the location of another robot relative to its own, both robots refine their internal beliefs based on the other robot's estimate, hence improving self-localization accuracy on both sides [4]. In our research, each robot used an omnidirectional camera as a range finder sensor to detect other robots and landmarks. The sensor can measure the range, bearing, and signature of a landmark relative to the robot local coordinate.



**Fig. 2.** Robots and landmarks.

As a characteristic of the omnidirectional camera, objects are displayed radially in its image; a near-to-far layout in real space is transformed into a center layout in the camera image space.

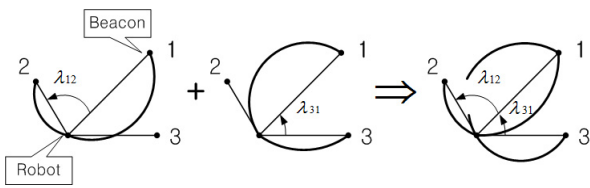
Fig. 2 shows an example omnidirectional image. This characteristic constrains the analytical ability of the camera because the noise distribution in the camera image space distance is transformed nonlinearly and variably from the real space distance. In order to investigate the noise distribution of the omnidirectional camera, we assume the input noise distribution is a Gaussian distribution and try to obtain the output noise distribution according to the geometry of hyperboloid projection.

## II. LOCALIZATION ALGORITHM

Localization is the process of finding both position and orientation of a vehicle in a given referential system [5-7]. Navigation of mobile robots indoors usually requires accurate and reliable methods of localization. Many transportation systems now using wire-guided automated vehicles may benefit from the increased layout design flexibility provided by a wire-free localization method such as triangulation with active beacons.

### A. Triangulation

Triangulation is based on the measurement of the bearings of the robot relative to beacons placed in known positions. It differs from trilateration, which is based on the measurement of the distances between the robot and the beacons. These beacons are also called landmarks by some authors. According to [8], the term beacon is more appropriate for triangulation methods. When navigating on a plane, three distinguishable beacons (at least) are required for the robot to localize itself (Fig. 3).  $\lambda_{12}$  is the oriented angle "seen" by the robot between beacons 1 and 2.



**Fig. 3.** Three-object triangulation.

It defines an arc between these beacons, which is a set of possible positions of the robot. An additional arc between beacons 1 and 3 is defined by  $\lambda_{31}$ . The robot is in the intersection of the two arcs. Usually, the use of more than three beacons results in redundancy.

**B. Trilateration**

Trilateration is a method to determine the position of an object based on simultaneous range measurements from three stations located at known sites. This is a common operation not only in robot localization [9], but also in kinematics, aeronautics, crystallography, and computer graphics. It can be trivially expressed as the problem of finding the intersection of three spheres, that is, finding the solutions to the following system of quadratic equations:

$$\begin{aligned} (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 &= l_1^2 \\ (x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 &= l_2^2 \\ (x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 &= l_3^2 \end{aligned} \quad (1)$$

where,  $x_1, x_2, x_3$  are the coordinates of the station,  $i$  and  $l_i$  is the range measurement associated with it.

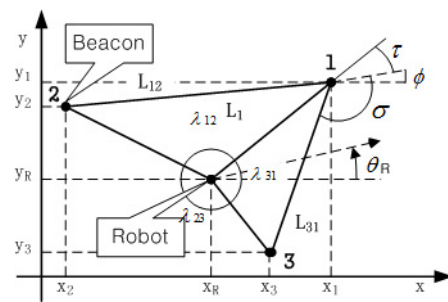
In Fig. 4, thick segments between stations define the base plane, and thin ones, those connecting the moving object and the stations, correspond to the range measurements.

**C. Bounding Box**

The bounding box method uses a simple box-shaped ranging area, as shown in Fig. 3, which can be specified by the extreme lower left coordinates,  $(x^l, y^l)$ , and extreme upper right coordinates,  $(x^u, y^u)$ . Fig. 5 shows an example where node  $i$  is bounded by the ranging area of its neighbors, nodes 1, 2, and 3. Thus, the bounding area for node  $i$  can be specified as follows:

$$\begin{aligned} (x^l(i), y^l(i)) &= (\max(x^l(j) | j \in N(i)), \max(y^l(j) | j \in N(i))) \\ (x^u(i), y^u(i)) &= (\max(x^u(j) | j \in N(i)), \max(y^u(j) | j \in N(i))) \end{aligned} \quad (2)$$

The bounding box method can be implemented as a distributed algorithm; Simic and Sastry [10] suggest that anchors broadcast their respective position estimates periodically, and nodes broadcast any changes in their estimates



**Fig. 4.** Three-object triangulation.

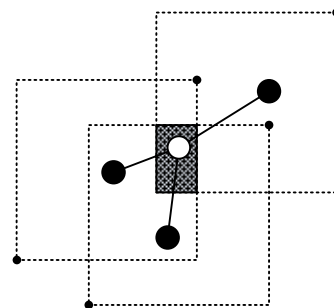
upon reception of any broadcast.

**III. TRACKING ALGORITHM**

**A. Particle Filter Algorithms**

The particle filter approach to track motion, also known as the condensation algorithm [11] and Monte Carlo localization, uses a large number of particles to ‘explore’ the state space. Each particle represents a hypothesized target location in state space. Initially, the particles are uniformly randomly distributed across the state space, and for each subsequent frame, the algorithm cycles through the steps illustrated in Fig. 6:

1. Deterministic drift: particles are moved according to a deterministic motion model (a damped constant velocity motion model was used).
2. Update probability density function (PDF): determine the probability for every new particle location.
3. Resample particles: 90% of the particles are resampled with replacement, such that the probability of choosing a particular sample is equal to the PDF at that point; the remaining 10% of particles are distributed randomly throughout the state space.
4. Diffuse particles: particles are moved a small distance in state space under Brownian motion.



**Fig. 5.** Bounding box.

This results in particles congregating in regions of high probability and dispersing from other regions; thus the particle density indicates the most likely target states. See [2] for a comprehensive discussion of this method. The key strengths of the particle filter approach to localization and tracking are its scalability (the computational requirement varies linearly with the number of particles), and its ability to deal with multiple hypotheses (and thus more readily recover from tracking errors). However, the particle filter was applied here for several additional reasons:

- It provides an efficient means of searching for a target in a multi-dimensional state space.
- It reduces the search problem to a verification problem, i.e., is a given hypothesis face-like according to the sensor information?
- It allows the fusion of cues running at different frequencies.

## B. Application of Particle Filter for Multi-Robots

In order to apply the particle filter algorithm to hand motion recognition, we extend the methods described by Huang [5]. Specifically, a state at time  $t$  is described as a parameter vector:  $s^t = (\mu, \phi^i, \alpha^i, \rho^i)$  where  $\mu$  is the integer index of the predictive model,  $\phi^i$  indicates the current position in the model,  $\alpha^i$  refers to an amplitude scaling factor and  $\rho^i$  is a scale factor in the time dimension.

### 1) Initialization

The sample set is initialized with  $N$  samples distributed over possible starting states and each assigned a weight of  $\frac{1}{N}$ . Specifically, the initial parameters are picked uniformly according to:

$$\left. \begin{aligned} \mu &\in [1, \mu_{max}] \\ \phi^i &= \frac{1-\sqrt{y}}{\sqrt{y}}, y \in [0, 1] \\ \alpha^i &= [\alpha_{min}, \alpha_{max}] \\ \rho^i &\in [\rho^{min}, \rho^{max}] \end{aligned} \right\} \quad (3)$$

### 2) Prediction

In the prediction step, each parameter of a randomly sampled  $s_t$  is used with  $s_{t+1}$  determined based on the parameters of that particular  $s_t$ . Each old state,  $s_t$ , is randomly chosen from the sample set, based on the weight of each sample. That is, the weight of each sample determines the probability of its being chosen. This is done efficiently by creating a cumulative probability table, choosing a uniform random number on  $[0, 1]$ , and then using binary search to pull out a sample [12]. The following equations are used to choose the new state:

$$\left. \begin{aligned} \mu_{t+1} &= \mu_t \\ \phi_{t+1}^i &= \phi_t^i + \rho_t^i + N(\sigma_\phi) \\ \alpha_{t+1}^i &= \alpha_t^i + N(\sigma_\alpha) \\ \rho_{t+1}^i &= \rho_t^i + N(\sigma_\rho) \end{aligned} \right\} \quad (4)$$

where  $N(\sigma_*)$  refers to a number chosen randomly according to the normal distribution with standard deviation  $\sigma_*$ . This adds an element of uncertainty to each prediction, which keeps the sample set diffuse enough to deal with noisy data. For a given drawn sample, predictions are generated until all of the parameters are within the accepted range. If, after, a set number of attempts it is still impossible to generate a valid prediction, a new sample is created according to the initialization procedure above.

### 3) Updating

After the prediction step above, there exists a new set of  $N$  predicted samples that need to be assigned weights. The weight of each sample is a measure of its likelihood given the observed data  $Z_t = (z_t, z_{t,1}, \dots)$ . We define  $Z_{t,i} = (z_{t,i}, z_{(t-1),i}, \dots)$  as a sequence of observations for the  $i^{\text{th}}$  coefficient over time; specifically, let  $Z_{(t,1)}, Z_{(t,2)}, Z_{(t,3)}, Z_{(t,4)}$  be the sequence of observations of the horizontal velocity of the left robot, the vertical velocity of the left robot, the horizontal velocity of the right robot, and the vertical velocity of the right robot, respectively. Extending Black and Jepsen, we then calculate the weight by the following equation:

$$p(z_t | s_t) = \prod_{i=1}^4 p(Z_{t,i} | s_t) \quad (5)$$

$$p(z_{t,i} | s_t) = \frac{1}{\sqrt{2\pi}} \exp \frac{-\sum_{j=0}^{\omega-1} (z_{(t-j),i} - \alpha^* m_{(\phi-\rho^*,i)}^{\mu})}{2(\omega-1)} \quad (6)$$

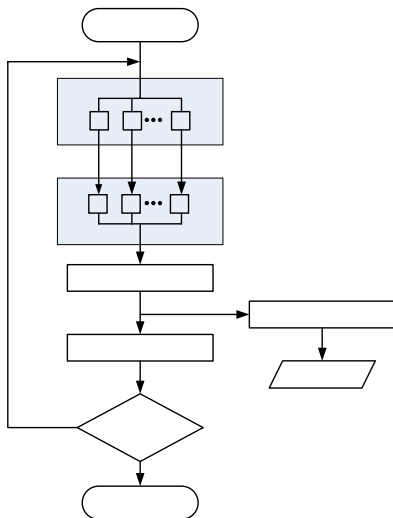


Fig. 6. Process of particle filter calculation.

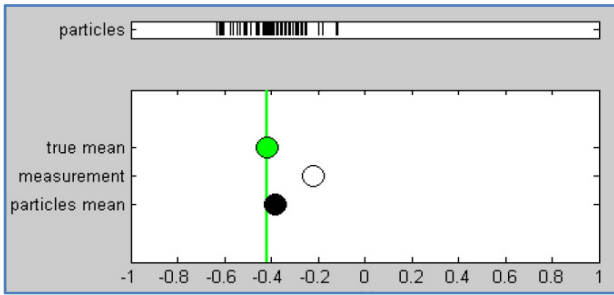


Fig. 7. Result 1 of experiment by particle filter.

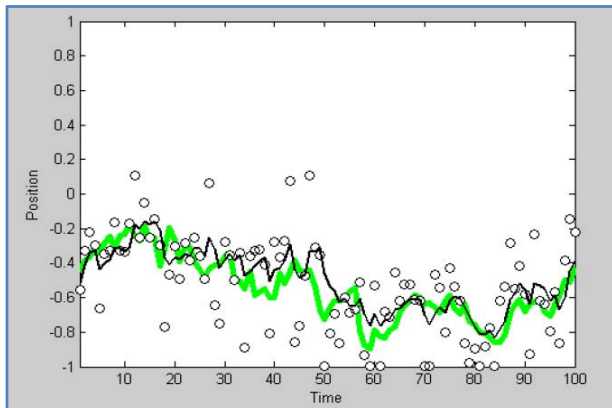


Fig. 8. Result 2 of experiment by particle filter.

where  $\omega$  is the size of a temporal window that spans back in time. Note that  $\phi^*$ ,  $\alpha^*$  and  $\rho^*$  refer to the appropriate parameters of the model for the group of robots in question and that  $\alpha^* m_{(\phi^* - \rho^* j), i}^{(\mu)}$  refers to the value given to the  $i^{\text{th}}$  coefficient of the model  $\mu$  interpolated at time  $\phi^* - \rho^* j$  and scaled by  $\alpha^*$ .

#### IV. EXPERIMENTAL RESULTS

To test the proposed particle filter scheme, we used MATLAB (MathWorks, Natick, MA, USA) and Visual Studio (Microsoft, Redmond, WA, USA). MATLAB was used for simulation of the particle filter and Visual Studio was used to calculate the mutual localization of the intruder and robot. Through the experiment, we confirmed that the accuracy of robot localization using the particle filter is greater than localization using only sensor information. Therefore, it is necessary to use the particle filter to localize the robot when there is no information except triangular measurement data. This is shown in Figs. 7 and 8.

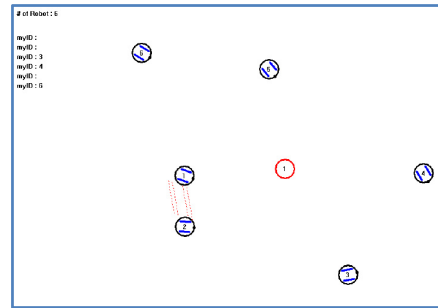


Fig. 9. Initial position of robot and intruder.

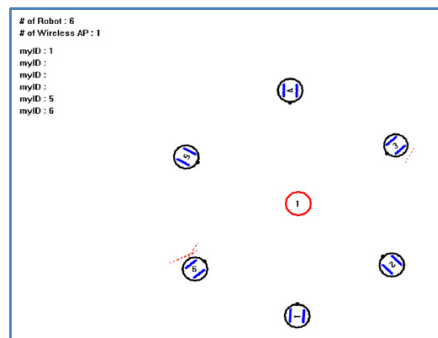


Fig. 10. Result of intruder enclosing experiment.

Also, we evaluate the algorithm of intruder enclosure formation using trilateration. First of all, we assume that initially there are 6 robots and repeat the experiment while swapping each robot's position with that of the intruder. Figs. 9 and 10 show the robots making a circle formation to enclose the intruder.

#### V. CONCLUSIONS

In this paper, we have developed the particle filter for swarm robot localization and circle formation enclosing a target that moves. This method encloses the invader in a circle formation based on mutual localization of the swarm robots without infrastructure. Therefore, we use trilateration, which does not require knowing the value of the coordinates of the reference points. We specify the enclosure point for the number of robots based on the relative positions of the robots in the coordinate system.

This scheme is important in providing a computationally feasible alternative for classifying the robot motion in a real system. We have proven that given an environment, a particle filter scheme classifies the robot location in real time.

## REFERENCES

- [1] Y. W. Lee, "Development of tracking filter for the location awareness of moving objects in ubiquitous computing," *International Journal of Maritime Information and Communication Sciences*, vol. 6, no. 1, pp. 86-90, 2008.
- [2] A. Farinelli, L. Iocchi, and D. Nardi, "Multirobot systems: a classification focused on coordination," *IEEE Transactions on Systems, Man, and Cybernetics Part B*, vol. 34, no. 5, pp. 2015-2028, 2004.
- [3] L. E. Parker, "ALLIANCE: an architecture for fault tolerant multirobot cooperation," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 220-240, 1998.
- [4] K. P. Sycara, "Multiagent system," *AI Magazine*, vol. 19, no. 2, pp. 79-92, 1998.
- [5] D. S. Huang, H. H. S. Ip, K. C. K. Law, and Z. Chi, "Zeroing polynomials using modified constrained neural network approach," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 721-732, 2005.
- [6] C. G. Kang and D. H. Kim, "Designing of dynamic sensor networks based on meter-range swarming flight type air nodes," *International Journal of Maritime Information and Communication Sciences*, vol. 9, no. 6, pp. 625-628, 2011.
- [7] Y. W. Lee, "Implementation of code generator of particle filter," *International Journal of Maritime Information and Communication Sciences*, vol. 8, no. 5, pp. 493-497, 2010.
- [8] J. Borenstein, H. R. Everett, and L. Feng, "Where am I? Sensors and methods for mobile robot positioning," The University of Michigan, Ann Arbor: MI, Technical Report, 1996.
- [9] L. E. Navarro-Serment, C. J. J. Paredis, and P. Khosla, "A beacon system for the localization of distributed robotic teams," in *Proceedings of the International Conference on Field and Service Robotics*, Pittsburgh: PA, pp. 232-237, 1999.
- [10] S. Simic and S. Sastry, "Distributed localization in wireless ad hoc networks," Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, Technical Report *UCB/ERL M02/26*, 2002.
- [11] J. Liu and J. Wu, *Multi-Agent Robotic Systems*, Boca Raton, FL: CRC Press, 2001.
- [12] Y. W. Lee, "Automation of an interactive interview system by hand gesture recognition using particle filter," *International Journal of Maritime Information and Communication Sciences*, vol. 9, no. 6, pp. 633-636, 2011.

**Yang-Weon Lee**

received a B.S degree in the Department of Electrical Engineering from Chung-Ang University, Korea, in 1982, and the M.S. degree in the Department of Control and Measurement Engineering from Seoul National University, in 1991. In 1996, he earned a Ph.D. degree in the Department of Electrical Engineering from Postech, Korea. During 1982-1996, he worked for ADD as a senior researcher. Since 1996, he has worked in the Department of Information and Communication Engineering at Honam University, where he now works as a professor. His current research interests include smart computing, target estimation and tracking filter design, communication signal processing, and .NET applications.