806 *Journal of Electrical Engineering & Technology Vol. 6, No. 6, pp. 806~816, 2011*
http://dx.doi.org/10.5370/JEET.2011.6.6.806

A Study on Swarm Robot-Based Invader-Enclosing Technique on Multiple Distributed Object Environments

Kwang-Eun Ko*, Seung-Min Park*, Junheong Park* and Kwee-Bo Sim†

Abstract – Interest about social security has recently increased in favor of safety for infrastructure. In addition, advances in computer vision and pattern recognition research are leading to video-based surveillance systems with improved scene analysis capabilities. However, such video surveillance systems, which are controlled by human operators, cannot actively cope with dynamic and anomalous events, such as having an invader in the corporate, commercial, or public sectors. For this reason, intelligent surveillance systems are increasingly needed to provide active social security services. In this study, we propose a core technique for intelligent surveillance system that is based on swarm robot technology. We present techniques for invader enclosing using swarm robots based on multiple distributed object environment. The proposed methods are composed of three main stages: location estimation of the object, specified object tracking, and decision of the cooperative behavior of the swarm robots. By using particle filter, object tracking and location estimation procedures are performed and a specified enclosing point for the swarm robots is located on the interactive positions in their coordinate system. Furthermore, the cooperative behaviors of the swarm robots are determined via the result of path navigation based on the combination of potential field and wall-following methods. The results of each stage are combined into the swarm robot-based invader-enclosing technique on multiple distributed object environments. Finally, several simulation results are provided to further discuss and verify the accuracy and effectiveness of the proposed techniques.

Keywords: Invader-enclosing, Object tracking, Location estimation, Particle filter, Potential field, Wall-following method

1. Introduction

Studies on intelligent robot systems with an aim of achieving next-generation alternative engines have recently started to emerge. In the future, we expect that every household will have a robot that takes care of people and cleans the house, among others. In addition, robots can replace humans in performing perilous tasks, such as in rescue missions, bomb disposal, and detection of an invader of automated security systems. For these reasons, interest in studying autonomous intelligent robot systems has grown rapidly. In particular, the study of an invader siege using the cooperative behavior of a swarm robot system has attracted widespread publicity [1–3]. In the cooperative robot system, each robot determines its own path based on the position of the other robots. In this case, it is important for the robots to detect each other as a target object and estimate the location of individual swam robots in complex environments.

The Kalman filter (KF) [4] is one of the most widely used methods for tracking and estimation due to its simplicity, optimality, and robustness. KF has the advantage of analytically obtaining the posterior distribution for state variable estimation. However, it is difficult to obtain the posterior distribution as KF if the actual system is nonlinear or the noise is not a Gaussian. Unlike the linear system, there is no optimal filter for the nonlinear system; instead, there are a variety of suboptimal filters. The most common approach is to use the extended KF (EKF) [5], which simply linearizes all nonlinear models in order for the traditional linear KF to be applied. Although EKF is used as a filtering strategy, over 30 years of experience with it has led to general consensus within the tracking and control community that it is difficult to implement and tune. The only reliable case for whole systems is almost linear on the time scale of update intervals. Recently, as a result of improved computer performance, an unscented KF (UFK) [6] and particle filter [7] using the sampling theory are being actively studied for estimating nonlinear systems. Unlike KF, particle filter represents the posterior distribution using a number of particles and the weight of each particle, which can represent the state variable. Particle filter has recently been known to be effective for the object tracking method.

In general, the location estimation method of an individual robot can be divided into a client-based method and a node-based method [8]. A client-based method

† Corresponding Author: School of Electrical and Electronics Engineering, Chung-Ang University, Korea. (kbsim@cau.ac.kr)
* School of Electrical and Electronics Engineering, Chung-Ang University, Korea.
Received: April 23, 2011; Accepted: June 14, 2011

calculates the position of individual robots using the received signal from the radio signal reception client, such as a tag. A global positioning system (GPS) is representative of a client-based tracking system client. The finger printing method through wireless LAN determines a position by using the signal sent to the client from an access point, and it can also be called a position recognition system based on the client. These methods determine the client position using strength, propagation time difference, and direction of the signal received from the client. Additionally, many other communication means are associated with the client-based method, such as RFID, laser, sonar, and radar, for application in an indoor environment. Recently, methods using Wi-Fi, UWB, and ZigBee are increasingly being used [9, 10]. Research on autonomous mobile robot has also been aiming to achieve cooperative behaviors and safe path navigation in a dynamic environment. Multiple robots are required to behave synchronously or cooperatively in the swarm robot system. A variety of algorithms are being studied to ensure safe navigation through obstacles in order to dynamically move or change the robot's migratory route [9, 11, 12]. Previous studies have presented cooperative schemes, which can be used for the synchronization of the swarm robot system [13].

One of the most widely used algorithms is the potential field. The potential field is a simple and effective technique, although it has some problems, such as the vibration and trap phenomena [8, 14]. A vibration phenomenon occurs when the robot meets an obstacle or narrow passage. Meanwhile, a trap phenomenon means that the robot was not able to get out the local minimum. To solve these problems, many studies have been performed. Waldo and Murray proposed the Laplace equation as harmonic functions in [15]. Borenstein and Korens proposed a solution by continuously developing the vector field histogram [3, 10, 14]. Another way to solve these problems is by combining the potential field methods with optimization algorithms, such as fuzzy rule, neural networks, and genetic algorithms [2, 3, 16]. In addition to the potential field method, Dijkstra's algorithm and A* algorithm, among the path planning algorithms, were also introduced. Dijkstra's algorithm is a graph search algorithm that solves the single-source shortest path problem for a graph with nonnegative edge path cost, producing the shortest path tree. For a given source vertex (node) in the graph, the algorithm finds the path with the lowest cost (i.e., the shortest path) between the vertex and every other one. It can be used to find the cost of the shortest paths from a single vertex to a single destination vertex by stopping the algorithm once the shortest path to the destination vertex has been determined. This algorithm must determine the distance between all nodes in advance, and it assumes that the distance cannot have a negative value and is not suitable for dynamic situations. A* algorithm is also used in path finding, which is the process of plotting an efficiently traversable path between points or what are called nodes. A* algorithm is an extension of Dijkstra's algorithm and achieves better performance (with respect to time) by using heuristics. Like Dijkstra's algorithm, A* algorithm requires complete information about the environment, thus a robot should have global information about the environment. To express the act of siege by adopting the abovementioned algorithms, we should consider the local input and global output of the system for mobile robot control [3]. Local input is the position coordinates of each robot and the invader, while global output is the final position of the object when the invader is enclosed by the swarm robots. The robot formation must first effectively prevent an invader escape. One formation that can be considered is the method of forming a circle around the invader. The robots determine this interactive location by considering the position of the other robots and the invader.

In this study, we present a method for tracking a specified object based on the particle filter in an environment with multiple moving objects. In this case, we suppose that a swarm robot system tracks the specified object, such as the invader. The path navigation-based decision technique of the cooperative behavior of the swarm robots is also suggested. By using the result of location estimation and object tracking, a specified enclosing point for the number of robots is determined as the interactive positions in the swarm robot systems. Also, the behaviors of the swarm robots are determined by the result of the potential field-based path navigation.

2. Related Works

2.1 Wheel encoder and gyroscope-based mobile robot movement control parameters

Wheel encoders can produce accurate measurement of translation but poor measurement of angle, while gyroscopes measure angles highly accurately. By combining the encoder for measuring distance and the gyroscope for measuring angles, a more accurate and useful device can be created.

The robots' motor controller calculates the position and orientation ($x_{encoder}$, $y_{encoder}$, $\theta_{encoder}$) from the encoder and sends the data to a software controller of a computer onboard. The mounted rate gyroscope communicates with a gyroscope driver, which integrates the rate values into an absolute angle (θ_{gyro}). The global position (x_{robot}, y_{robot}) is determined by transforming the translation vector from the encoder space to the gyro space. The global angle (θ_{robot}) is the gyroscope's angle (θ_{gyro}). By using these parameters, we can describe the following factors:

$$dx = x_{encoder}^{t} - x_{encoder}^{t-1} \tag{1}$$

$$dy = y_{encoder}^{t} - y_{encoder}^{t-1} \quad (2)$$

$$d\theta = \theta_{gyro}^{t} - \theta_{encoder}^{t} \quad (3)$$

$$x_{robot}^{t} = x_{robot}^{t-1} + \cos(d\theta)dx - \sin(d\theta)dy \quad (4)$$

$$y_{robot}^{t} = y_{robot}^{t-1} + \sin(d\theta)dx + \cos(d\theta)dy \quad (5)$$

$$\theta_{robot}^{t} = \theta_{gyro}^{t} \quad (6)$$

However, the actual location and measurement are different due to the systematic noises and accidental error during movement according to the control input.

2.2 Particle filter

The achievement of a particle filter for object tracking has been an active research topic in recent years. Particle filtering is a recursive Bayesian filter that estimates the posterior distribution conditioned on observations. It is a sequential Markov-chain Monte-Carlo method [17]. The key idea of the method is to represent the distribution by a set of particles with nonnegative weights [24-26]. Particle filters are sophisticated model estimation techniques based on simulation, and they have been proven very successful for non-linear estimation problems. The estimation process based on particle filters is shown below:

1. Assume that $P(X_{t-1} \mid Y_{1:t-1}) \approx \left\{\left(X_{t-1}^{(i)}, \omega_{t-1}^{(i)}\right)\right\}_{i=1,\ldots,N}$.
2. Generate an updated particle set by sampling from the proposal distribution, $X_t^{(i)} \sim q(X_t \mid X_{0:t-1}, Y_{1:t})$.
Usually $q(X_t \mid X_{0:t-1}, Y_{1:t}) = P(X_t \mid X_{t-1})$
3. Reweigh each particle according to the following formula and normalize it so that $\omega_t^{(i)}$ sums to 1:

$$\omega_t^{(i)} \propto \omega_{t-1}^{(i)} \frac{P(Y_t \mid X_t^{(i)})P(X_t^{(i)} \mid X_{t-1}^{(i)})}{q(X_t^{(i)} \mid X_{0:t-1}^{(i)}, Y_{1:t})}$$

4. Resample with replacement in proportion to new importance weights.

The purpose of a particle filter is to estimate the sequence of hidden parameters X_t based on the observed data Y_t (t = 0,1,2,⋯,n). All Bayesian estimates of X_t follow from the posterior distribution $P(X_t \mid Y_{1:t})$. The particle filtering procedure monitors the posterior probability of a first-order Markov process through the following formula:

$$P(X_t \mid Y_{1:t}) = \alpha P(Y_t \mid X_t) \int_{X_{t-1}} P(X_t \mid X_{t-1}) P(X_{t-1} \mid Y_{1:t-1}) \quad (7)$$

where, X_t is the unobserved process state at time t and Y_t is the observation at time t. And, $Y_{1:t}$ is the sequence of observation from time 1 to time t, $P(X_t \mid X_{t-1})$ is the state motion model or the dynamic distribution process, $P(Y_t|X_t)$ is the observation model or the observation likelihood distribution, $P(X_t \mid Y_{1:t})$ is the current object state or the posterior distribution, $P(X_{t-1} \mid Y_{1:t-1})$ is the previous object state, and α is a normalizing factor. The integral in (7) does not have a closed form solution, except in the most basic cases; hence, particle filters are used to approximate (7) using a set of weighted samples $\{X_t^{(i)}, \omega_t^{(i)}\}$, where each $X_t^{(i)}$ is an instantiation of the process state, known as a particle, and $\omega_t^{(i)}$ is the corresponding particle weights. A particle presentation of this density is as follows:

$$P(X_t \mid Y_{1:t}) \approx \sum_{i=1}^{N} \omega_t^{(i)} \delta(X_t - X_t^{(i)}) \quad (8)$$

where δ is the *Dirac-delta* function.

Similarly, we use the recursive definition of Eq. (8) to compute the filtered distribution $P(X_t \mid Y_{1:t})$ given the distribution $P(X_{t-1} \mid Y_{1:t-1})$. With a particle representation for $P(X_{t-1} \mid Y_{1:t-1})$, (8) can be approximated as follows:

$$P(X_t \mid Y_{1:t}) \approx \alpha P(Y_t \mid X_t) \sum_{i=1}^{N} \omega_{t-1}^{(i)} P(X_t \mid X_{t-1}^{(i)}) \quad (9)$$

To maintain the filtering distribution in (10), a particle filter inductively assumes that a set of N particles represents the filtering distribution at the previous time step, i.e., $P(X_{t-1} \mid Y_{1:t-1}) \approx \{(X_{t-1}^{(i)}, \omega_{t-1}^{(i)})\}_{i=1,\ldots,N}$.

A new set of N particles $\{X_t^{(i)}\}_{i=1,\ldots,N}$ is sampled from a proposal distribution $X_t^{(i)} \sim q(X_t|X_{0:t-1}, Y_{1:t})$, and the importance weights of this set are computed according to the following:

$$\omega_t^{(i)} \propto \omega_{t-1}^{(i)} \frac{P(Y_t \mid X_t^{(i)})P(X_t^{(i)} \mid X_{t-1}^{(i)})}{q(X_t^{(i)} \mid X_{0:t-1}^{(i)}, Y_{1:t})} \quad (10)$$

As an important final step, the particles in the new set are resampled (with replacement) in proportion to their importance weights to generate a uniformly weighted particle approximation of $P(X_t \mid Y_{1:t})$. This resampling step is necessary because without it, the distribution of the importance weights tends to become skewed such that after a few time steps, very few particles have non-zero importance weight.

3. Particle Filter-Based Location Estimation

Because of the influence of systematic noise and accidental errors during movement according to the control input, the actual location and measurement results differ. In order to interpolate the difference, a particle filter model is constructed to estimate the location of the target object. This problem is associated with the circle formation problem of the enclosed moving target. This method is based on the enclosed invader in a circle formation on the mutual localization of the swarm robots without

infrastructure. The first robot to be able to detect the invader becomes the leader. After creating a relative coordinate system, it calculates the enclosed point and transmits data to the helper robots. The correct position of the helper robots in the relative coordinate system must be determined in order to precisely enclose the invader. The helper robots must arrive at the correct enclosed position. However, because the invader is moving, the relative coordinate system changes. Therefore, the position of the helper robots must be recalculated. In the new relative coordinate system, the helper robots' position is changed, but the enclosure point remains the same. In this study, we propose a particle filter for achieving a more accurate location recognition of the robot.

3.1 Prediction

Prediction means the presumption of the next positions of the particles. Because of the static property of a node, its true location cannot be changed. When particles are updating by the prediction step, those that are close to the true position of a node will achieve bigger weights and have higher probability to be duplicated. In this study, all particles are first distributed near the already known starting position of a helper robot. We then predict the next location using an encoder and a gyroscope in the movement of the helper robot.

$$\overline{bel(x_t)} = \int P(x_t \mid x_{t-1}, u_{t-1}) bel(x_{t-1}) dx_{t-1} \tag{11}$$

A probabilistic change of particles following the motion model to applying Bayes Filter is expressed in (11). The probability $bel(x_{t-1})$ from the previous location x_{t-1} applies with the motion model $P(x_t|x_{t-1},u_{t-1})$. Consequently, $bel(x_{t-1})$ can be calculated by $\overline{bel(x_t)}$ from the new location x_t.

3.2 Weight calculation

Calculating the weights of particles is an important problem. As in traditional particle filters, the weight of a particle can be determined by calculating the difference between the predicted position and the distance measurement. In this study, we calculate the weight of each particle by comparing the calculated location of the robot by triangulation using beacons with the predicted position of each particle. The probability of the current position $bel(x_t)$ is calculated by (12), and the sensor model $P(z_t \mid x_t)$ is used to predict the location of the robot.

$$bel(x_t) = \eta_t P(z_t \mid x_t) \overline{bel}(x_t) \tag{12}$$

Then $\eta_t P(z_t \mid x_t)$ becomes the concept of weight. Hence, weight calculation through (12) is determined using (13).

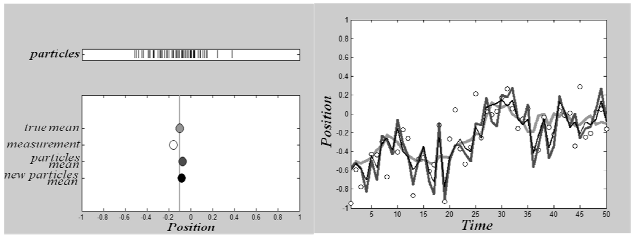
$$\omega_t = P(z_t \mid x_t) \times \omega_{t-1} \tag{13}$$

3.3 Resampling

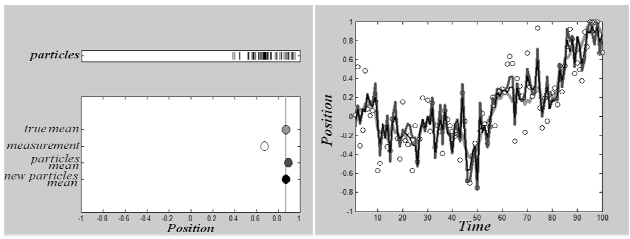
The basic idea of resampling is to eliminate trajectories that have small normalized importance weights in order to focus on trajectories with large weights. The sampling step is employed to reduce the degeneracy of particle filtering. Each particle acquires weight through the previous two phases. In this phase, the new particles are sampled again using the weight at time t and they express the current location of a robot more accurately [27].

3.4 Particle filter-based location estimation experiment

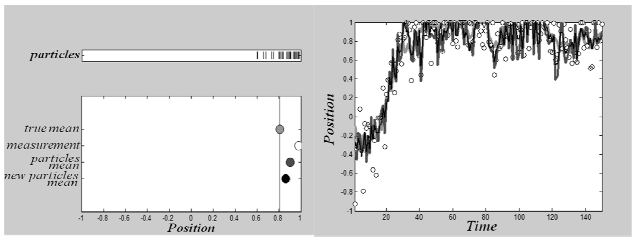
To demonstrate the effect of the particle filter-based location estimation, a simulation was performed in a MATLAB simulation environment. Fig. 1 shows the analytical result, which sets the number of creation sampling particles (N) at 50, 100, and 150.



(a) $N = 50$



(b) $N = 100$



(c) $N = 150$

Fig. 1. Result of particle filter-based location estimation experiment ($N = 50$, 100, and 150)

True mean is the actual location of a robot, which can be measured by wheel encoder and gyroscope. The particles mean and new particles mean are the results of the location

estimation. These experimental results show that the proposed method is more accurate than a simple sensor measurement in estimating the current location of a robot.

4. Particle Filter-Based Object Tracking

4.1 Particle filter model for object tracking

In general, video-based object tracking deals with non-stationary image streams that change over time. Robust and real-time tracking of a moving object with video-based input is a problematic issue in this research area. In order to realize this technique, the particle filter -based method is used in this study. In this method, a state representation X_t, which may include object locations and scales, must be chosen. In addition, three distributions must be established: 1) the process dynamic distribution, $P(X_t|X_{t-1})$, which describes object movements between time steps; 2) the proposal distribution, $X_t^{(i)} \sim q(X_t \mid X_{0:t-1}, Y_{1:t})$, which is sampled at each time step to update the particle distribution; and 3) the observation likelihood distribution, $P(Y_t \mid X_t^{(i)})$, which describes how objects appear within the image sequence Y_t.

In this study, we selected the initial state of the object to be tracked through mouse input as $P(X_0)$ and calculated the dynamical probability distribution, $P(X_t|X_{t-1})$. The object X_t is defined as the object we want to track, which is denoted as a square {*x, y, width, height*} in the image frame. We also generated particles $X_t^{(i)}$ through a $N(\mu, \Sigma)$ with mean value μ, which is the current position of the object, as shown below:

$$X_t^{(i)} \sim N(\text{position of } X_{t-1}, \Sigma) \tag{14}$$

We then defined the proposal distribution as the process of dynamical probability distribution, as follows:

$$q(X_t \mid X_{0:t-1}, Y_{1:t}) = P(X_t \mid X_{t-1}) \tag{15}$$

In this study, we used the hue, saturation, value (HSV) color-based histogram. HSV is the most common cylindrical coordinate representations of points in an RGB color model, which rearranges the geometry of RGB in an attempt to be more perceptually relevant than the Cartesian representation. In general, histogram-based retrievals in HSV color space show better performance than those in RGB color space [18]. These procedures determine the next location of the object by comparing the result of an object's histogram and each particle's location.

4.2 Results of object tracking experiment

Figs. 2, 3, and 4 show tracking samples using particle filter. Fig. 2 shows a video sequence of a moving ASIMO that records a relatively large object tracking. Fig. 3 shows the image sequence of a computer simulation that records relatively small object tracking. From these figures, it can be seen that the movement of an object is being tracked well. Fig. 4 shows the tracking result of a specified object in an environment with multiple moving objects using the proposed particle filter-based method. In this experiment, one video is adopted to test the performance of our proposed method. This video is available in [28]. From these figures, it can be seen that the moving object is being tracked well in an environment with multiple objects.

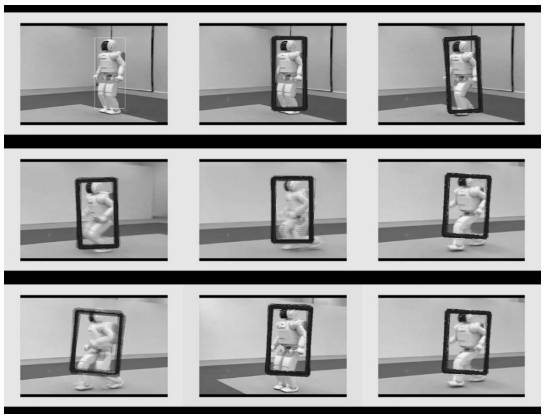


Fig. 2. Single large object motion tracking

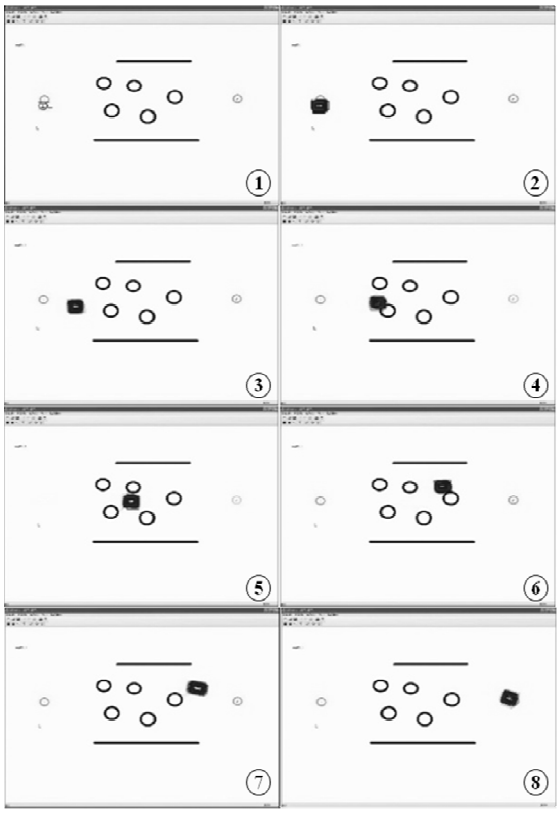


Fig. 3. Single small object motion tracking

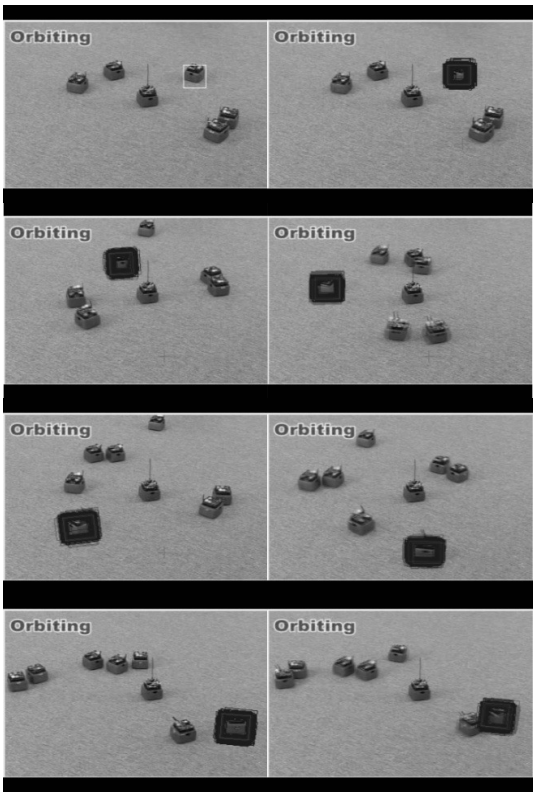


Fig. 4. Specified object motion tracking in multiple moving objects environment

5. Swarm Robot-Based Invader-Enclosing Technique

In an environment with swarm robot systems, the robot that first detects the invader becomes the leader robot during the patrol. After creating the relative coordinate system, the leader robot must calculate the enclosed point and transmit it to a helper robot. When the coordinate of the leader robot is the origin (0, 0), the coordinate of the invader becomes the enclosing point (0, *r*). The user has to decide the *r* of the distance in the surrounding environment and the invader's size. The leader robot tracks the invader to keep the coordinates and sends detection message to the surrounding robots.

In general, an enclosed formation will eventually take the form of a circle, as shown in Fig. 5. The message is received by the surrounding robots, which become the helper robots. The leader robot calculates the coordinates of the enclosure point as much as the number of helper robots. As shown in Fig. 5, the angle between the enclosed points to the center of the invader (i.e., target (0, *r*)) θ is calculated by the formula 360°/(*n*+1), where θ is equivalent to 60°. Therefore, the angle between the leader robot and each helper robot to the center of the invader can be calculated by the equation $m \times \theta$ (m=1, 2,…, n), where the angle is equivalent to 60° in the case of Helper 1 and 120° in the case of Helper 2.

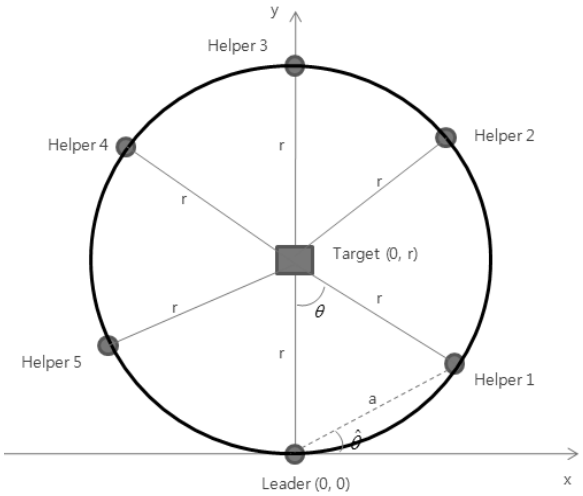


Fig. 5. Enclosure point of circle formation

At this point, the *Leader* calculates the enclosed point in the three different cases by the obtained space angle, as follows:

Case 1: $m \times \theta < 180°$, (m = 1, 2,…, n)

The distance between the *Leader* robot and the enclosed points is unknown by the second law of cosine. For example, the distance between the *Leader* and *Helper* 1 is equal to the following equation:

$$a^2 = r^2 + r^2 - 2r^2 \cos 60° \tag{16}$$

The angle between the enclosed point and the *x* axis is calculated by the following equation:

$$\hat{\theta} = 90° - \frac{(180° - m \times \theta)}{2} \tag{17}$$

In Fig. 5, the $\hat{\theta}$ of *Helper* 1 is equal to 30°, and the coordinates of the enclosed point can be calculated because the distance of a and $\hat{\theta}$ is already determined. The coordinates of the enclosed point of the *Helper*s are computed by the following:

$$(x, y) = (a \times \cos \hat{\theta}\,,\, a \times \sin \hat{\theta}\,) \tag{18}$$

Case 2: $m \times \theta = 180°$, (m = 1, 2,…, n)

The enclosed point is in a straight line with the *Leader* and the invader; hence, it is (0, 2*r*).

Case 3: $m \times \theta > 180°$, (m = 1, 2,…, n)

The enclosed point is symmetric on the *y* axis against the enclosed point with an angle of 360° ($m \times \theta$). This means that the enclosed point is equal to ($-a \times \cos \hat{\theta}\,,\, a \times \sin \hat{\theta}\,$).

As shown in Fig. 5, assuming that the enclosing point is calculated, the number of responding robots to the message

equals 5, and r is equal to 2. From these conditions, the following parameters can be obtained.

***Helper 1*:** a = 2 because of $a^2 = r^2 + r^2 - 2r^2 \cos 60°$ and $\hat{\theta}$ = 30° because of $m \times \theta$ = 60°. Hence, coordinate ($a \times \cos\hat{\theta}$, $a \times \sin\hat{\theta}$) is equal to ($\sqrt{3}$, 1).

***Helper 2*:** $a = 2\sqrt{3}$ and $\hat{\theta}$ = 60° because of $m \times \theta$ = 120°. Hence, coordinate ($a \times \cos\hat{\theta}$, $a \times \sin\hat{\theta}$) is equal to ($\sqrt{3}$, 3).

***Helper 3*:** Because $m \times \theta$ = 180°, coordinate ($a \times \cos\hat{\theta}$, $a \times \sin\hat{\theta}$) is equal to (0, 4).

***Helper 4*:** Because $m \times \theta$ = 240°, 360° − $m \times \theta$ = 120°. Hence, the enclosed point is symmetric on the y axis against *Helper* 2. The coordinate is ($-\sqrt{3}$, 3).

***Helper 5*:** Because $m \times \theta$ = 300°, 360° − $m \times \theta$ = 60°. Hence, the enclosed point is symmetric on the y axis against *Helper* 1. The coordinate is ($-\sqrt{3}$, 1).

6. Potential Field-Based Path Planning and Mobile Robot Navigation

6.1 Potential field

Probably none of the robot navigation methods has attracted so great an interest from researchers as the potential fields [19], such that so many variations of the method have been developed and used [20-23]. The method was developed as a basis for generating smooth trajectories for both mobile and manipulator robotic systems. Separately, an attractive and a repelling potential field are constructed to represent the relationship between the robot and each of the objects within the object's sensory range. These fields are then combined to yield a single global field. A smooth trajectory is computed based on the gradient within the globally computed potential field. In this case, the potential field model consists of the repelling force, $\mathbf{F}_{rep}$ of the obstacles and the attractive force, $\mathbf{F}_{att}$ of the goal point of the robots. The vector sum of the repelling and attractive forces is defined as the direction of the robot, which determines factor $\mathbf{F}_{tot}$. The navigation of the robots can be performed by moving the robot in order to minimize this factor $\mathbf{F}_{tot}$. This approach generally assumes knowledge of the type of obstacles in the environment, and polygons or spheres approximate these known obstacles in the planning phase. It was assumed that the environment in the original formulation of this concept is static; however, the use of this approach for dynamic environments has been adopted.. The potentials are associated with the objects in the environment when they are encountered. Fig. 6 shows an example with repelling forces from the obstacles, walls, and a superimposed general field direction from the start to goal. The figure also exemplifies the potential field generation steps in the form of 3D surface plots.

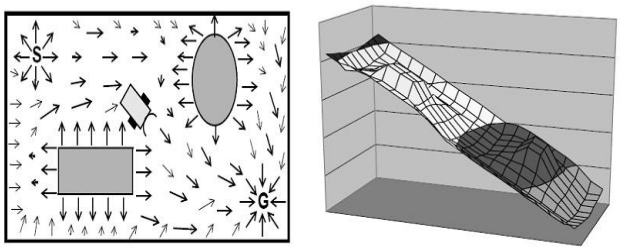


Fig. 6. Illustration of the potential field in 2D and 3D surfaces

However, the potential field has some defects. First, the trap phenomenon, which is caused by the narrow width of the road, can occur. Second, the phenomenon of a lack of passage between the two near obstacles can exist. Third, a local minimum problem can take place.

Fig. 7 shows an experiment that avoids obstacles and wherein travels to the target use the potential field. In this experiment, R5 and R6 robots avoid obstacles and reach their goals well, unlike the other robots (R1–R4), which cannot move due to the defects of the potential field. For example, R1 shows that if the width of the road suddenly becomes narrow, then a vibration occurs and causes the trap phenomenon. Moreover, R2 and R3 do not pass between two near obstacles because of the lack of passage. Lastly, R4 shows a failure to avoid the obstacles, because of the local minimum problem.

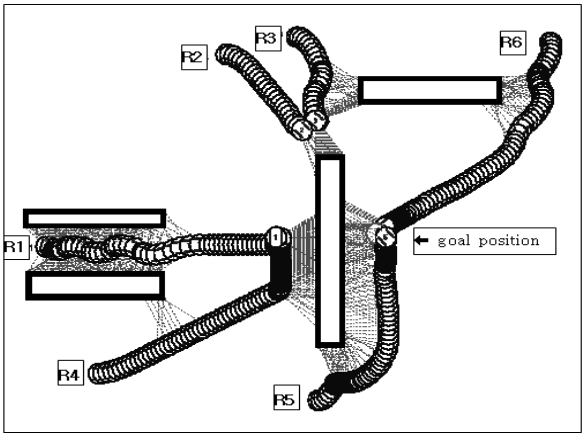


Fig. 7. Potential field-based obstacles avoidance

6.2 Wall-following mode

There are two types of modes for obstacle avoidance: the potential field mode and the wall-following mode. The potential field mode can cause trap phenomenon, narrow passage, and local minimum problem. These limitations

have been serious issues, and early attempts were made to overcome them in a variety of ways.

In this study, the wall-following model is optionally applied as a possible alternative to solve the above mentioned problems.

If the trap phenomenon, narrow passage, and local minimum problems arise during the potential field procedure, the wall-following mode is applied on behalf of the potential field mode.

When the robots are faced with a narrow passage between two near walls, the robot's frontal sensors will be not able to detect the walls; however, using the wall-following technique, the flank sensors can detect the walls. In this case, the robot has to reduce its movement speed, which can drive the original direction along the walls. If the robot's sensor detects the range at less than half of the maximum range of the sensor, then the robot's direction is modulated up to 30° from the original direction. This move by the robot can help induce an entry into the narrow passage and reduce an oscillation in this environment.

Secondly, if $\mathbf{F}_{tot}$ becomes 0, then the robot determines the direction via the repulsive force, $\mathbf{F}_{rep}$ only and ignores the attractive force, $\mathbf{F}_{att}$. If there are obstacles on the right side, then the robot's direction is modulated up to 90° clockwise from the original $\mathbf{F}_{rep}$. If there are obstacles on the left side, then the direction is modulated up to 90° counter-clockwise from the original $\mathbf{F}_{rep}$. Fig. 8 shows the simulation of the obstacle avoidance based on the wall-following mode.

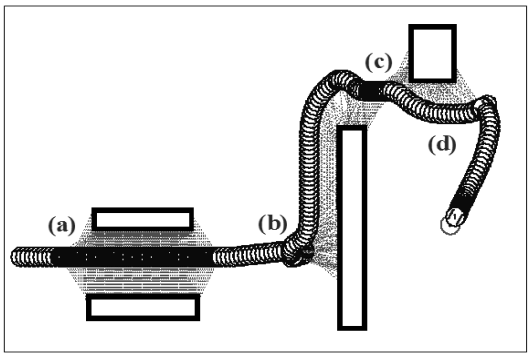


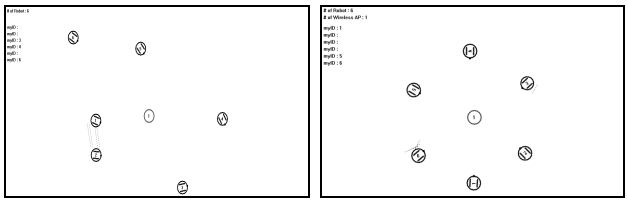
Fig. 8. Wall-following-based obstacle avoidance

In Fig. 8, case (a) deals with the trap phenomenon, which causes the deceleration and oscillation of the robots. However, the oscillation-like behavior of R1 in Fig. 7 is almost reduced by the wall-following mode. Case (b) shows obstacle avoidance without falling into the local minimum. Furthermore, in case (c), the robot successfully passed through the two obstacles nearby, unlike R2 and R3 in Fig. 7. Only in case (d), the robot does not drive to the optimal path and returns because the wall-following mode could not turn into the potential field mode on time. Future modifications are necessary for the optimal algorithm, but it can be seen that the robot successfully reaches the goal.

7. Simulation

In this study, in order to prove the effectiveness of the proposed algorithm, different simulations were performed three times. The results are shown in Figs. 9, 10, and 11, respectively. The red circles mark the starting point; the goals of the robot are denoted by each point that is inscribed in a circle; and the two lines in blue symbols represent the robot. The black symbols represent the obstacles. When the robot's sensors detect an obstacle, the robot draws a red dotted line from itself to the obstacle. Small circles represent the invader.

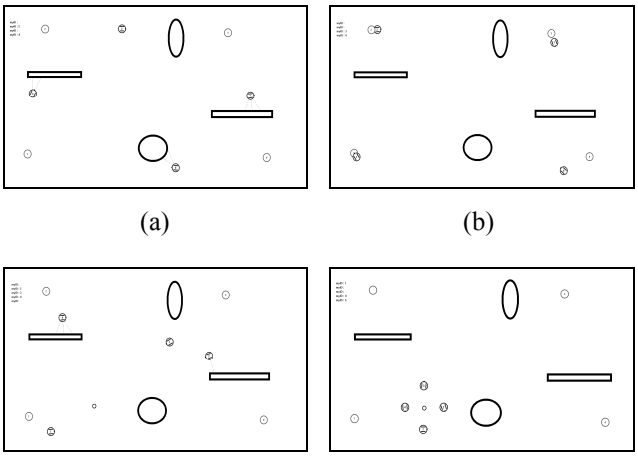
The first experiment shown in Fig. 9 is the simplest experiment. This experiment involves six robots in an obstacle-free environment enclosing the static invader by circle formation. Fig. 9(a) shows a situation wherein the six robots are located randomly, and the invader is shown in the middle. In Fig. 9(b), all the robots enclose the invader in the middle by circle formation.



(a) (b)

Fig. 9. First experiment: (a)→(b)

The second experiment shows that four robots discover and enclose the invader during patrolling. In this case, the invader does not move. In Figs. 10(a) and 10(b), the robots are patrolling. In Figs. 10(c) and 10(d), the robots discover and enclose the invader in circle formation.



(a) (b)

(c) (d)

Fig. 10. Second experiment: (a)→(b)→(c)→(d)

The final experiment shows the case wherein the invader is moving. Complex obstacles are placed in this

environment; the invader is moving while the robots are going to enclose the intruder. The robots succeed again to enclose the invader. The patrol of the four robots is shown in Figs. 11(a), 11(b), and 11(c). The invader appears during the time the four robots are patrolling in (d). Fig. 11(e) shows that a part of the swarm robots succeeds in enclosing the moving invader. Finally, it can be seen that the entire robots succeed in enclosing the invader in (f).

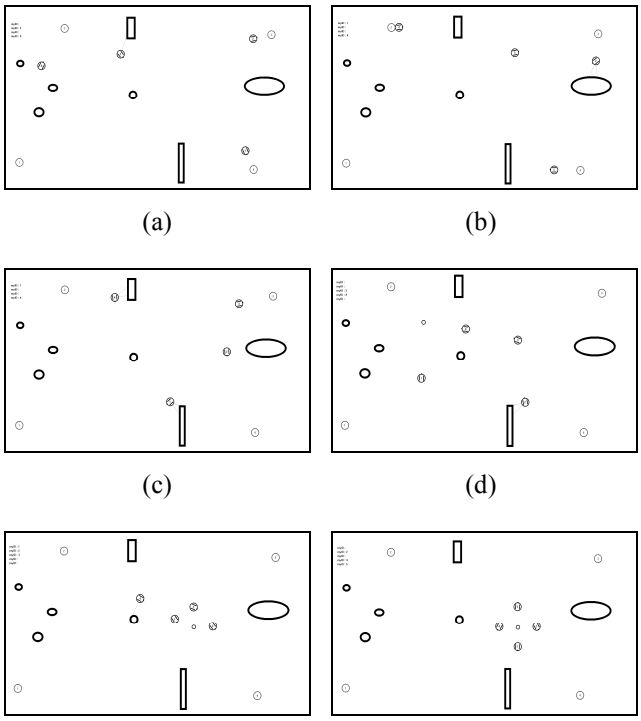


Fig. 11. Third experiment: (a)→(b)→(c)→(d)→(e)→(f)

From the above simulations of different situations, we can see the robots enclosing the invader successfully using circle formation by the proposed algorithm. The experiments were performed repeatedly, with each simulation result closely monitored. Through these simulation results, we conclude that satisfactory performance could be demonstrated by the proposed algorithm.

8. Conclusion

In this study, we deal with an invader-enclosing technique based on the interactive location recognition system with swarm robots. At first, a specified object tracking method using a particle filter is proposed. Unlike previous approaches that requires analytical posterior distribution, such as KF,, the particle filter-based method represents the posterior distribution using a number of particles and the weight of each particle, which can represent the state variable. The experimental results show that the movement of an object is tracked well in an environment of multiple moving objects. Secondly, we also used a particle filter to improve the accuracy of the result of the robot location estimation. Swarm robots create the relative coordinate system, which consists of multiple robot nodes and the target node, by using this objective tracking method. After creating the relative coordinate system, it calculates the enclosed point and transmits information to each helper robot. Lastly, each robot is moved by the potential field based on the wall-following mode for obstacle avoidance and path planning-based safe navigation.

The proposed invader-enclosing algorithm is implemented as an MFC-based simulation program. The improvements of the proposed method are suggested based on experimental results obtained via virtual invader-enclosing simulation. The experimental results are presented to show the effectiveness of the proposed technique.

Acknowledgments

This work was supported by the Mid-career Researcher Program through NRF grant funded by the MEST (No. 2010-0029226).

References

[1] H. Yamaguchi, “A Cooperative Hunting Behavior by Mobile Robot Troops,” *International Journal of Robotics Research*, vol. 18, no. 9, pp. 931-940, 1999

[2] D. J. Park and B. E. Mullins, “Toward Finding an Universal Search Algorithm for Swarm Robots,” *in Proceedings of 2003 IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, pp. 1945-1950, 2003

[3] K. Sugawara, I. Yoshihara, K. Abe and M. Sano, “Cooperative behavior of interacting robots,” *Artificial Life and Robotics*, vol. 2, no. 2, pp. 62-67, 1998

[4] G. Welch, G. Bishop, “An Introduction to the Kalman Filter,” *UNC-Chapel Hill, TR 95-041*, July 24, 2006

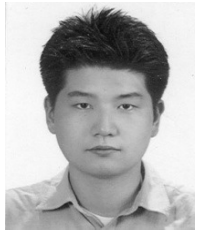
[5] S. J. Julier, J. K. Uhlmann, “A New Extension of the Kalman Filter to Nonlinear Systems,” *in Proceedings of AeroSense: 11th Int. Symp. Aerospace/Defense Sensing, Simulation and Controls*, pp. 182-193, 1997

[6] E. A. Wan and R. Van der Merwe, “The Unscented Kalman Filter for Nonlinear Estimation,” *in Proceddings of Symp. Adaptive Syst. Signal Process., Commun. Contr.,* pp. 153-158, 2000

[7] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, 2002

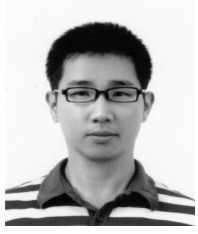
[8] G. Sun, J. Chen, and W. Guo, “Single Processing

Techniques in Network-aided Positioning," *IEEE Signal Proc. Magazine*, vol. 22, no. 4, pp. 24-40, 2005
[9] Yousief, M. Horus, "A WLAN-based indoor location determination system," *Ph.D thesis, Univ. of Maryland at College Park*, 2004
[10] Y. Lin, I. Jan, P. Ko, Y. Chen, J. Wong, G. Jan, "A Wireless PDA-based Physiological Monitoring System for Patient Transport," *IEEE Transactions on IT in Biomedicine*, vol. 8, no. 4, pp. 439-447, 2004
[11] F. Gustafsson and F. Gunnarsson, "Mobile Positioning Using Wireless Networks", *IEEE Signal Proc. Magazine*, vol. 22, no. 4, pp. 41-53, 2005
[12] Marco Crepaldi, Analysis, design and simulation of an UWB receiver for indoor localization, *Dissertation, Politecnico Di Torino*, 2005.
[13] J. S, Kim, F. Allgower, "A Nonlinear Synchronization Scheme for Hindmarsh-Rose Models," *Journal of Electrical Engineering & Technology*, vol. 5, no. 1, pp. 163-170, 2010
[14] D. Koller, J. Weber, and J. Malik, "Robust Multiple Car Tracking with Occlusion Reasoning," *in Proceedings of the 3rd European conference on Computer Vision*, vol. 1, pp.189-199, 1994.
[15] S. Gezici, Z. Tian, G. B. Giannakis, H. Ko-bayashi, A.F. Molisch, H. Vincent Poor, and Z. Sahinoglu, "Localization via Ultra-Wideband Radios," *IEEE Signal Proc. Magazine*, vol. 22, no. 4, pp. 70-84, 2005
[16] A. H. Sayed, A. Tarighat, and N. Kha-jehnouri, "Networked-based Wireless Networks," *IEEE Signal Proc. Magazine*, Vol. 22, No. 4, pp. 24-40, 2005
[17] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, editors. "Sequential Monte Carlo Methods in Practice," *Springer, New York*, 2001
[18] Sangoh Jeong, "Histogram-Based Color Image Retrieval", *Psych221/EE362 Project Report*, 2001
[19] J. Borenstein, Y. Koren, "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation", *in Proceeding of the IEEE International Conference on Robotics and Automation*, California, April 1991
[20] L. C. A. Pimenta and A. R. Fonseca, "Robot Navigation Based on Electrostatic Field Computation", *IEEE Transaction on Magnetics*, vol. 42, no. 4, 2006
[21] J. Ren, K. A. McIsaac and R. V. Patel, "Modified Newton's Method Applied to Potential Field-Based Navigation for Mobile Robots", *IEEE Transaction on Robotics*, vol. 22, no. 2, 2006
[22] R. Daily, D. M. Bevly, "Harmonic Potential Field Path Planning for High Speed Vehicles", *American Control Conference 2008*, Seattle, USA, June 11-13 2008
[23] S. Yannier, A. Onat, A. sabanovic, "Basic Configuration for Mobile Robots", *in Proceeding of International Conference on Industrial Technology*, vol. 1, pp. 256-261, Maribor, Slovenia, 2003
[24] Y. Li and S. Li, "Particle Filtering for Range-Based Localization in Wireless Sensor Networks," *in Proceedings of the 7th World Congress on Intelligent Control and Automation*, Chongqing, China, June 25-27, 2008
[25] E. S-. Navarro, V. Vivekananda, V. W. S. Wong, "Dual and Mixture Monte Carlo Localization Algorithms for Mobile Wireless Sensor Networks," *IEEE Wireless Communications and Networking Conference*, Hong Kong, March 11-15 2007
[26] M. Rudafshani and S. Datta, "Localization in Wireless Sensor Networks", *in Proceedings of the 6th International Conference on Information Processing in Sensor Networks*, Cambridge, Massachusetts, April, 25-27, 2007
[27] Madow, "On the theory of systematic sampling II," *Annals of Mathematical Statistics*, pp. 333-354, 1949
[28] J. Mclurkin, "Multi-Robot Systems Engineering," *Rice University, Department of Computer Science*, URL=<http://people.csail.mit.edu/jamesm/project-MultiRobotSystemsEngineering.php >



Kwang-Eun Ko received his B.S. and M.S. degrees in Electrical and Electronics Engineering from Chung-Ang University, Seoul, Korea, in 2007 and 2009, respectively. He is currently a candidate for Ph.D. degree in the School of Electrical and Electronics Engineering at Chung-Ang University. His research interests include machine learning, emotion recognition, and multimodal intention recognition.

Seung-Min Park received his B.S. degree in Electrical and Electronics Engineering from Chung-Ang University, Seoul, Korea, in 2010. He is currently pursuing his Master course in the School of Electrical and Electronics Engineering at Chung-Ang University. His research interests include pattern recognition, brain-computer interface, and evolutionary computation.



Junheong Park received his B.S. degree in Electrical and Electronics Engineering from Chung-Ang University, Seoul, Korea, in 2011. He is currently pursuing his Master course in the School of Electrical and Electronics Engineering at Chung-Ang University. His research interests include machine learning, brain-computer interface, and gesture recognition.

Kwee-Bo Sim received his B.S. and M.S. degrees in Electronic Engineering from Chung-Ang University, Seoul, Korea, in 1984 and 1986, respectively, and his Ph. D. degree in Electronic Engineering from the University of Tokyo, Japan, in 1990. Since 1991, he has been a faculty member of the School of Electrical and Electronic Engineering at Chung-Ang University, where he is currently a professor. His research interests are artificial life, neuro-fuzzy and soft computing, evolutionary computation, learning and adaptation algorithm, autonomous decentralized system, intelligent control and robot system, artificial immune system, evolvable hardware, and artificial brain.