In: Robot Vision: New Research
Editor: Taisho Matsuda, pp.
ISBN 978-1-60456-984-1

Chapter 5

REAL-TIME PEOPLE TRACKING AND FOLLOWING USING A VISION-CONTROLLED MOBILE ROBOT

***Yuhua Zheng**[*] **and Yan Meng**[°]*
Department of Electrical and Computer Engineering
Stevens Institute of Technology, Hoboken, NJ 07030, USA

Abstract

In this chapter, we will focus on the study of using a mobile robot equipped with one Pan-Tilt-Zoom camera to track and follow a person in an indoor environment. To address the challenge of real-time performance, a bio-inspired tracking algorithm is proposed, which combines swarm intelligence and a multi-Gaussian model to achieve higher robustness and flexibility under dynamic environments. Then the tracking results will be fused with other sensory information including laser range finder and SONARs. Based on these multi-sensor signals and the control model of the robot, a Kalman filter is applied to synchronize the vision-based data and the motion control of the robot to prevent the tracked target from losing due to robot motion. A Pioneer 3AT mobile robot system, equipped with one PTZ canon camera, one laser range finder, two SONAR rings, is used to track and follow people in an indoor environment. Experimental results demonstrate the efficiency and robustness of the proposed algorithm.

1. Introduction

People tracking and following is a very critical problem for service robots working in public places, like museums, classrooms, or office environments. Many intelligent robotic systems are designed to fulfill these tasks including museum-guide robot, nurse robot, nanny robot, etc. Intelligent behaviors with real-time performance are essential for such robot systems to adapt to the dynamically changing environment and to focus on the interested people simultaneously. Extensive works in the literature have been conducted on the related topics. Basically, these methods can be classified into two categories. The methods of the first

[*] Email address: yzheng1@stevens.edu
[°] Email address: yan.meng@stevens.edu

category mainly collect environment data through laser range finder, SONARs, or a sensor network to detect people movement in a preset map. The vision data is usually used to find the most matched candidate from the images compared with the tracked object. In [1], an intelligent space was proposed, where many sensors and devices are distributed. A robot was controlled by this intelligent space to follow a people within this space through its resources and data collected from the sensor network. In [2] and [3], only the data from laser range finders were adopted by the probability-based filters to detect and track people, especially for multiple-person with temporary occlusions. However, it is difficult for the laser range finders to distinguish one object from another only based on the distance information. Since the camera is such a powerful sensor, where the vision data contains most powerful and useful object information, such as color, shape, texture, etc., compared to other sensor data. Vision sensors can definitely help to develop more complex and efficient tracking algorithms. Therefore, on the second category, visual information is applied as a primary tracking cue with support from other sensor signals, such as laser range finders or SONARs. In [4] a camera and a laser sensor were used to detect people's faces and legs, separately. And then by integrating these data together, a people can be tracked. In [5], the head of a person is detected by using skin colors. The camera is treated as a pinhole model, and is calibrated in the 3-D space. The person following behavior is demonstrated by recalibrating the camera and wheels axis of the robot. Two cameras were used in [6] to achieve the depth information and approach people tracking and following.

In this chapter, we plan to build up a vision-controlled mobile robot system to execute people tracking and following in an indoor environment. A swarm intelligent based visual tracking algorithm is applied for people tracking. The tracking results are used to lead the following behaviors. Meanwhile, a laser range finder measures the distance between the object and the robot, which is then combined with the visual tracking results using a Kalman filter based method to control the robot movements. Compared with other methods, the proposed visual tracking algorithm is fast, robust and flexible, and supplies a reliable tracking of the object.

The chapter is structured as follows. Section 2 describes the swarm intelligence based visual tracking algorithm. Section 3 proposes a Kalman filter based method to fuse the vision data and other sensing data together to control the robot motion for people tracking and people following. Experimental results for people tracking and following using a Pioneer 3AT mobile robot are demonstrated and discussed in Section 4. Section 5 provides the conclusion and future work.

2. Visual Tracking Algorithm

A new swarm intelligence based optimization algorithm, Particle Swarm Optimization (PSO) [7] [8], is applied for people tracking to improve system efficiency and robustness, especially under dynamic environments. PSO is a bio-inspired searching algorithm, which introduces the communication mechanism to accelerate the convergence of searching agents in the potential solution space. Therefore, the optimal solution can be found in a short time. The proposed algorithm treats the object tracking as a searching problem in a high-dimension solution space. Then several object features are embedded into a fitness function to drive the PSO evolution. The basic idea is that a swarm of particles fly around the image and try to find the best-fit tracking window. When some particles successfully detect the interested objects, they will share this information with their neighbors. Each particle makes its own decision not only based on its neighbors, but also on its own cognition, which provides the flexibility and ability of exploring new areas. This decision-making procedure can efficiently prevent the local optimum effect. With the convergence of particles, the optimal solution can be achieved.

Compared with other available object tracking methods, one advantage of the proposed algorithm is the accommodation of dynamic environments. By using the multi-dimensional solution space, the tracking window is being automatically updated on position, scale and appearance simultaneously. Secondly, the exploration ability of particles ensures the algorithm to be robust to wiggling images and local maxima or minima traps. Furthermore, the convergence of the algorithm can accelerate the searching process, and achieve a real-time tracking performance by using small size of particles, for example, less than 30.

2.1. Particle Swarm Optimization (PSO)

Proposed by Kennedy and Eberhart in 1995 [9] [10], PSO is the simulation of a simplified social model, which obviously has its root in artificial life in general, and in swarming theory in particular. As a population-based method, PSO achieves the optimal solution to a problem by driving multiple potential solutions in a higher search space to converge to some optimal points. Usually, potential solutions are represented by particles in a virtual search space; and a fitness function is defined as the underlying mechanism to direct particles' movement. The social metaphor that leads to PSO can be summarized as follows: the individuals that are part of a society hold an opinion that is part of a "belief space" (the search space) shared by neighboring individuals. Individuals may modify this "opinion state" based on three factors:

- The trend of keeping its own way (inertia part)
- The individual's previous history of states (individual part)
- The previous history of states of the individual's neighborhood (social part)

An individual's neighborhood may be defined in several ways, configuring the "social network" of the individuals. Following certain rules of interaction, the individuals in the population adapt their scheme of belief to the ones that are more successful among their social network. Over time, a culture arises, in which the individuals hold opinions that are closely related.

In the PSO algorithm, each individual is called a "particle", and is subject to a movement in a multi-dimensional space that represents the belief space. Particles have memory, thus retaining part of their previous states. There is no restriction for particles to share the same point in belief space, but in any case their individuality is preserved. Each particle's movement is the composition of an initial random velocity and two randomly weighted influences: individuality, the tendency to return to the particle's best previous position, and sociality, the tendency to move towards the neighborhood's best previous position.

Abstractly in the solution space, the velocity and position of the particle at any iteration is updated based on the following equations:

$$v_{id}^{t+1} = w \cdot v_{id}^{t} + c_1 \cdot \varphi_1 \cdot (p_{id}^{t} - x_{id}^{t}) + c_2 \cdot \varphi_2 \cdot (p_{gd}^{t} - x_{id}^{t}) \quad (1)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (2)$$

where v_{id}^{t} is the component in dimension *d* of the *i*th particle velocity at iteration *t*, and x_{id}^{t} is the component in dimension *d* of the *i*th particle position at iteration *t*. c_1, c_2 are constant weight factors. p_{id}^{t} is the best position achieved by particle *i*, and p_{gd}^{t} is the best position found by the neighbors of particle *i*. φ_1, φ_2 are random factors in the (0,1) interval, which have different values on each dimension of the state space. w is the inertia weight. The PSO requires the tuning of some parameters: the cognitive and sociality weights c_1, c_2, and the inertia factor w.

According to Equation (1), each particle adjusts its velocity by combing three forces: keeping the velocity of last moment, moving to the best position from its own memory, and moving to the best position found by its neighbors. Different parameters in Equation (1) provide varied balance among those three factors. Then defined by Equation (2), a particle moves in the search space according to the combined velocity to achieve a new position, which presents a new potential solution.

2.2. PSO Based Method for Visual Tracking

A PSO-based searching algorithm can be applied for people tracking and following, where the searching is conducted to look for the best match of the predefined object in the current scene. Firstly, according to object model, the potential solution space is generated. This multi-dimensional solution space usually represents the interested parameters of the tracking window; and different points in the space represent different values. For object tracking, the location and the size of the tracking window might be the most concerned factors. And each point in the solution space is also associated with a fitness value which indicates how good the current point is. The good fitness function should be able to distinguish the object from noisy background easily and accelerate the PSO searching.

Once the solution space is generated, particles are initialized to cover the possible solution region. Then the particles move around by following the rules defined by Equations (1) and (2). Being driven by the fitness function based on people features, the optima in the solution space would attract more and more particles to converge. After convergence, the new tracking window can be presented according to these optimal points. Then based on the appearance of the tracked people in a new environment, the values of feature model will be updated for next frame.

2.2.1 Solution Space of the PSO Algorithm

To identify an object in an image, usually rectangle windows are utilized. Four parameters will be identified to describe the rectangle windows, including 2D location of the central point, width and height of the rectangle, as shown in Fig.1. These parameters can be used to build up a four-dimensional search space. In such a space, each particle actually presents a search window with specific location and size as: $P = \{p_i \mid p_i(x_i, y_i, l_i, w_i), i = 1,2,...,N\}$, where

(x_i, y_i) represent the central point of the rectangle related to particle i; l_i and w_i represents the length and width related to particle i, respectively; and N is the population of swarm particles. Each individual particle has different values of these parameters. In other words, they are distributed in a four-dimensional search space.

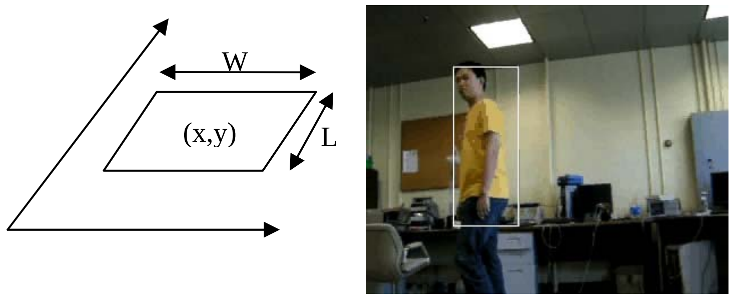


Figure 1. Four parameters associated with a particle window. The left image shows the 4-dimesional solution space that particles are associated with. The right image presents the rectangle tracking window on a real image. A person is marked.

Theoretically, the particles should be distributed all over the four-dimensional solution space. However, the search would take much longer if the solution space is very large. Here, we apply the motion-based constraints to diminish the search area. A straightforward constraint is the movement continuity of the tracked object since it is reasonable to assume that the object motion is continuous under most tracking situations. In other words, the tracking window of a new frame should be close or adjacent to the window of the previous one on both location and size.

After being initialized, particles move around within the solution space, share information and try to converge to the optima. According to the convergence characteristics discussed before, particles' trajectories depend on both the fitness function and the control weights. The fitness function actually decides the distribution of the solution space; and the control weights specifically constrain particle motions.

2.2.2 Selection of the PSO Parameters

Due to the mechanism of sharing information, the PSO algorithm is capable of finding solutions with a small size of particles. Generally, a group of 10 to 30 particles should be enough for most problems. As we discussed before, the initialization of the particles are based on the assumption of the movement continuity of the tracked object. So the particles on the new frame are evenly distributed around the previous window. For images with the size of 320 x 240, the locations of the new particles can move up and down up to 25 pixels from the center, and the sizes of the windows can be shrunk and extended up to 20%.

From the initialization to the final convergence, the PSO algorithm needs a number of iterations to catch the object. The stop conditions of the iterations include the maximum number of iterations is reached, the fitness value is within a predefined threshold value, and no more improvement can be observed. These conditions can be used separately or combined together. For most tested experiments, a good result can be approached after about 6 or 7 iterations, which makes it feasible to achieve a real-time performance.

For people tracking, a global society network is expected to work better than the local neighborhood. Because a global network means all members in the group share only one global best which conducts a faster clustering. The inertia factor is set as an initial value at the beginning to explore new solutions; then it would be reduced to ensure convergence. The weight of cognitive factor decreases slowly to prevent the particles being trapped into a local maximum.

When the search is finished, it is difficult to ensure that all particles converge to the exactly same point in the solution space. However, most particles are clustered closely around the global optima. To get a reasonable solution, all the central points of the qualified particles will be calculated for the new tracking window. A particle is qualified if its fitness value is beyond the preset threshold which actually ensures the associated window is quiet close to the optimum.

2.2.3 Fitness Function

Using PSO, the feature model decides the fitness value of a point in the solution space, which tells how good this point is. A good feature model can accelerate and enhance the searching of particles. In this paper, color histogram is applied to build the fitness function.

First, the images need to be transformed from RGB to HSV color space. Then the hue values over all pixels will be collected to generate a histogram. When the PSO-based searching algorithm is running, each particle at every moment covers a region that is associated with a histogram. The best matched candidates for tracking can be obtained by comparing their histograms with the target histogram. Therefore, a method to measure the distance between two histograms is required.

The Bhattacharyya Coefficient [11] is used to measure the similarity between these two histograms as:

$$BC(H(t), \hat{H}(t-1)) = \sum_{i \in n} \sqrt{H_i(t), \hat{H}_i(t-1)} \tag{3}$$

where $H(t)$ represents the histogram of a particle, $\hat{H}(t-1)$ represents the histogram of the target at time *t-1*, and *n* denotes the range. $H_i(t)$ and $\hat{H}_i(t-1)$ are components of color *i*. By using (3), the distance between two histograms can be defined as:

$$D(H(t), \hat{H}(t-1)) = \sqrt{1 - BC(H(t), \hat{H}(t-1))} \tag{4}$$

This distance is invariant to the scale of the target, while the popular used histogram intersection is scale variant [12] [13]. The smaller this distance is, the greater the fitness value is, and the better match the particle has with the target object.

2.2.4 Algorithm Summary

Considering the problem of object tracking as a searching process for the best match region of the object model, PSO is applied to achieve more efficiency, higher robustness, and real-time performance. This algorithm can be summarized as followings:

PSO-based object tracking:

Given object description $p_{b'}(x_{b'}, y_{b'}, l_{b'}, w_{b'}, \theta_{b'})$, object histogram $\hat{H}(t-1)$, and new image frame $I(t)$.

Step1: pre-processing the image frame to reduce noise.

Step2: initialize particles around the previous object as $P = \{p_i \mid p_i(x_i, y_i, l_i, w_i), i = 1,2,..., N\}$.

For every particle *i*:

{

Step3: update its location using Equations (1) and (2).

Step4: calculate fitness value of new locations using Equation (4); update the local and global best $\hat{x}$ and $\hat{s}$ for next iteration.

}

Step5: after visiting all particles, if stop conditions are met, go to step6; otherwise go to step3.

Step6: update object tracking window by averaging the qualified particles; update object histogram.

3. People Following using A Vision-Controlled Mobile Robot System

3.1 The System Overview

The current robot system is capable of executing navigation, mapping, and avoiding obstacles. By adopting the powerful algorithms of computer vision, the robot can achieve more tasks with higher level, especially for human-robot-interface applications. The vision-controlled mobile robot system is developed here. The concept is using high-level visual processing to guide the movements and behaviors of the robot. Meanwhile, other sensors are applied as supporting sensors to collect environmental data. An overview of the proposed system is showed in Fig. 2.

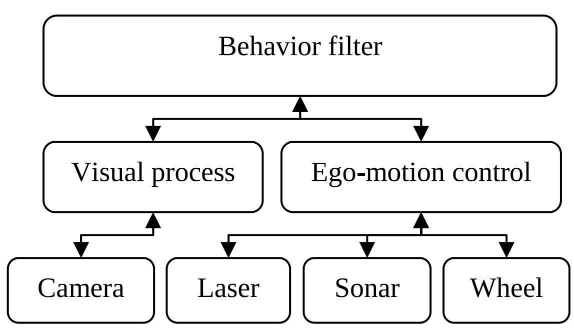


Figure 2. The overview of a vision-controlled mobile robot system. It has a three-layer structure. The bottom layer includes basic sensors and wheels, which runs the actual data collecting and action execution. The middle layer contains a number of data processing units. Each unit takes a specific function. The top layer filters data from middle units and conducts behavior strategies.

Vertically the system has three layers. The bottom layer includes the sensors and wheels. Sensors collect data and pass the data to the middle layer. Meanwhile, wheels receive commands from the upper level and drive the robot moving. This layer supplies the very basic functions that the higher level can use. Generally, the middle layer contains some computation units and each unit is capable of executing one or more specific functions. Among them there are two important components, visual processing unit and ego-motion control unit. In a vision-controlled mobile robot system, the visual processing unit contains algorithms of computer vision, which mainly process images captured by the camera and distill useful visual information to the top layer. For instance, this unit tracks people from image frames and pass the tracking results to the top behavior layer. Similarly, the ego-motion control unit deals with general robot movements like stepping forward, backward, turning and etc. Usually the robot's behaviors are generated by combing or repeating these basic actions. The top layer is called behavior filter, which combines vision and motion data to conduct real complex robot behavior. These behaviors usually consist of a group of basic actions and present meaningful movements, like obstacle avoidance, people following, etc. Under some situations, the robot may need to produce more than one behavior and the filter will decide when and how to switch between those behaviors.

3.2 The People Following Strategy

By adopting the system structured in Fig.2, a people tracking and following strategy is proposed, as shown in Fig. 3. Firstly, the robot navigates itself within the environment by collecting images from the camera. When the interested person is detected, it starts its people tracking algorithm by calling the visual process unit. At the same time, the distance between the robot and the person is calculated by employing a laser range finder. Once the interested people is tracked, the robot tries to follow the walking people by integrating the imagery tracking result and the distance information. Then, the path to follow the person is conducted. And obstacle avoidance is triggered when the robot runs into any obstacle on its way. Following a path and obstacle avoidance can be treated as behaviors, which will be executed through the ego-motion control unit.

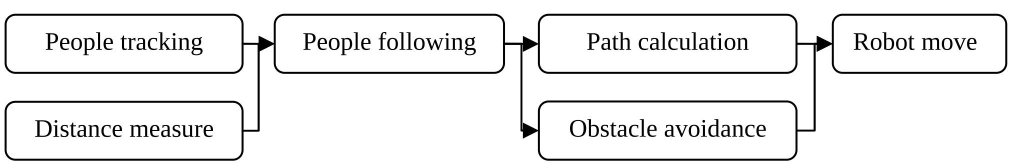


Figure 3. The people tracking and following strategy. It is a recursively sequential process. Based on the visual tracking result, the people following behaviors are calculated. Then the obstacle avoidance is added to prevent the robot from colliding with obstacles.

In this chapter, the visual tracking with a single camera is considered, which cannot provide accurate distance information to the person. The laser range finder provides the range if the scanning plan intersects the tracked person. This assumption is not always true, especially for outdoor environments with uneven terrain. However, for most indoor situations like offices, the tracking and range data are generally feasible and stable.

Once the tracking results and distance to the person are given, the path to follow the person can be calculated. Since the robot is moving, there are two issues need to be considered: camera calibration and ego-motion compensation. Typically, the ego-motion compensation can be accomplished by using the odometry with low deviation over a short distance. In this project, we adopt the compensation method based on difference of image frames from [14]. To follow the person, the direct way is to keep the absolute distance between the robot and the person in a desired range. Furthermore, based on the trajectory of the person, the motion model can be constructed. A Kalman filter based method is then applied to estimate the next movement of the tracked person.

4. Experimental Results

4.1 The Experimental Platform

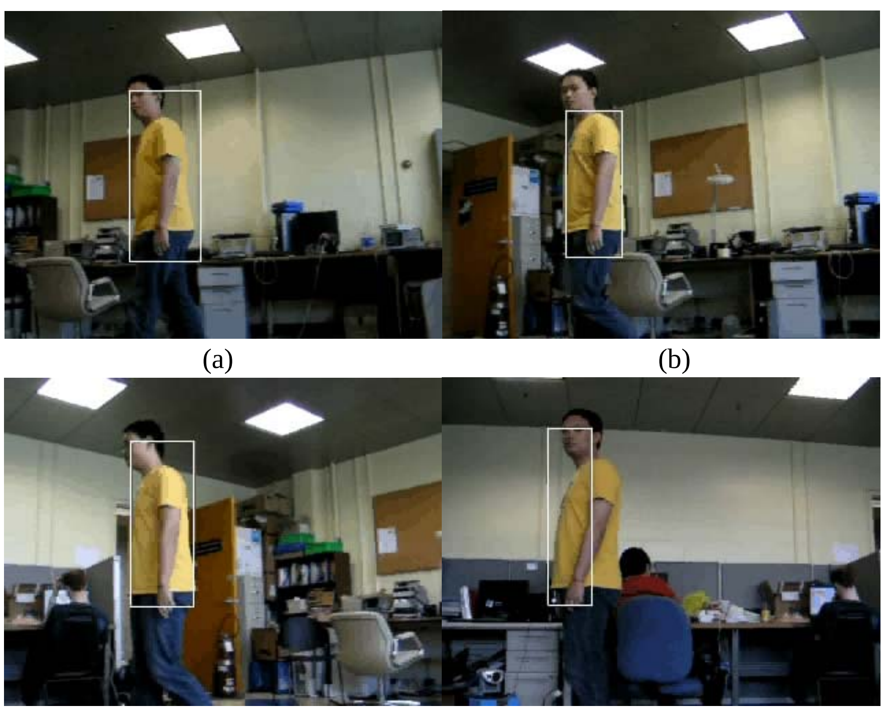
The proposed system is implemented and evaluated on a Pioneer 3AT mobile robot system. As shown in Fig. 4, this P3-AT robot is equipped with one PTZ camera, one laser range finder, eight forward and eight rear SONARs. With powerful motors and wheels, it can reach the speed of 0.8 m/s. This robot offers an embedded computer with a Renesas SH7144-based microcontroller, which provides the computational power for onboard vision processing and robot motion control. And a software library called ARIA supplies the development environment for users. To implement the vision processing, OpenCV, a library of programming functions mainly aimed at real time computer vision, is adopted. With functions supplied by ARIA and OpenCV, users can focus more on developing high-level programs.

Figure 4. The robot for people tracking and following experiments.

4.2 Experimental Result

To test the proposed algorithm, a people tracking and following experiment is set up as Fig. 5. A person is walking in an office and a robot is trying to track and follow him. From the experimental results in Fig. 6, it is clear that the person is tracked and followed well. During the person walking, the robot adjusts itself to keep him in the middle of scenes. Even with the changing backgrounds and lightening conditions, the tracking performance is robust. Also the robot keeps a certain distance with the person so that the scale of the person in the image keeps stable as well.

Figure 5. The people tracking and following experiment.



(a) (b)

(c) (d)

Figure 6. The people tracking and following results. During the whole progress, the person is tracked and followed. With the following by the robot, the person is kept in the middle of scenes.

5. Conclusion

This paper proposes a vision-controlled mobile robot system to execute people tracking and following in an indoor environment. A swarm intelligence based visual tracking algorithm is applied to for robust people tracking. With the tracking ability, a Kalman filter is constructed based on visual images, distance from laser range finder, and the robot control model is applied to follow the tracked people. An indoor experiment demonstrates the efficiency and robustness of the proposed algorithm.

People tracking and following actually can include many complex scenarios, like following people with occlusions, multi-people tracking, etc. The proposed system structure shown in Fig. 2 is capable of integrating more components for each layer. Therefore, other algorithms can be added into the system to enhance/improve the system performance for other more complex situations. In the future, we will investigate and extend our proposed method to people tracking and following under more complex dynamic environments.

Reference

[1] K. Morioka, Joo-Ho Lee, H. Hashimoto, “Physical agent for human following in intelligent sensor network”, *IEEE/RSJ International Conference on Intelligent Robots and System*, vol.2, pp. 1234- 1239, 2002.

[2] Dirk Schulz, Wolfram Burgard, Dieter Fox, Armin B. Cremers, “People Tracking with a Mobile Robot Using Sample-based Joint Probabilistic Data Association Filters”, *International Journal of Robotics Research (IJRR)*, Vol. 22(2), pp. 99-116, 2003.

[3] E.A. Topp, H.I. Christensen, “Tracking for following and passing persons”, IEEE/*RSJ International Conference on Intelligent Robots and Systems,* pp. 2321- 2327, 2005.

[4] M. Kleinehagenbrock, “Person Tracking with a Mobile Robot Based on Multi-Modal Anchoring”, *In IEEE Int. Workshop on Robot and Human Interactive Communication (ROMAN)*, 2002

[5] H. Sidenbladh, D. Kragic, H.I. Christensen, “A person following behaviour for a mobile robot”, *IEEE International Conference on Robotics and Automation,* vol.1, pp. 670-675, 1999.

[6] H. Kwon, Y. Yoon, J. B. Park, and A. C. Kak, "Person Tracking with a Mobile Robot using Two Uncalibrated Independently Moving Cameras," *Proceedings of IEEE International Conference on Robotics and Automation*, Barcelona, Spain, April 2005

[7] J. Kennedy, R. C. Eberhart, and Y. Shi, “Swarm intelligence”, San Francisco, Morgan Kaufmann Publishers, 2001.

[8] R. C. Eberhart and Y. Shi, “Particle Swarm Optimization: Developments, Applications, and Resources,” *in Proc. Congress on Evolutionary Computation*, Korean, 2001.

[9] R. C. Eberhart and J. Kennedy. "A New Optimizer Using Particles Swarm Theory", *in Proc. Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan., 1995.

[10] J. Kennedy and R.C. Eberhart, "Particle Swarm Optimization", *in Proceedings of IEEE International Conference on Neural Networks*, pp. 1942-1948, 1995.

[11] T. Kailath, “The Divergence and Bhattacharyya Dis-tance Measures in Signal Selection," *IEEE Transaction on Communication Technology*, Vol. COM-15, pp.52-60, 1967.

[12] D. Comaniciu, V. Ramesh, and P. Meer, “Real-Time Tracking of Non-Rigid Objects using Mean Shift”, *International Conference on Computer Vision and Pattern Recognition* , Vol. 2, pp. 142-149, 2000.

[13] M. J. Swain, D.H. Ballard, “Color Indexing," *International Journal of Computer Vision*, Vol. 7(1), pp. 11-32, 1991.

[14] Boyoon Jung and Gaurav S. Sukhatme, “Detecting Moving Objects Using a Single Camera on a Mobile Robot in an Outdoor Environment”, *International Conference on Intelligent Autonomous Systems*, pp. 980-987, Amsterdam, The Netherlands, March 2004.