Proceedings of the 2007 IEEE Symposium on Computational
Intelligence in Security and Defense Applications (CISDA 2007)

The PSO-Based Adaptive Window for People Tracking

Yuhua Zheng and Yan Meng
Department of Electrical and Computer Engineering
Stevens Institute of Technology, Hoboken, NJ 07030
yzheng1@stevens.edu, yan.meng@stevens.edu

***Abstract*— This paper presents a robust tracking algorithm using an adaptive tracking window associated with five parameters, where the parameters of the tracking window are optimized by a particle swarm optimization (PSO) algorithm. Basically, the optimization of a tracking window is transformed into a searching algorithm in a five-dimension feature space, which constrains the possibilities of the window. Particles associated with different parameters fly around the searching space independently, while they are sharing information from the society and adjust their behaviors to achieve the global optimization, which means the most optimized parameters for the tracking window. Appearance histogram is employed to calculate the fitness function for particles, where the distance between histograms is measured by histogram intersection. Estimated people motion is utilized to expedite the convergence of particles. Experimental results of people tracking demonstrate that the algorithm is efficient, robust, and adaptive to various rigid and non-rigid people motions.**

I. Introduction

Visual interpretation of people and their movements is an important issue in many applications, such as service robots, surveillance systems, public security systems, virtual reality interfaces, and other government, commercial, and entertainment applications. The ability to find and track people is therefore an important visual problem. When the vision system is installed on a mobile platform, such as a mobile robot, the problem becomes more challenging since the background environments are dynamic and the people are not always visually isolated but are partially occluded by other objects.

Up to now, the underlying mathematical models of most tracking methods are Bayesian Estimation and Hidden Markov Model (HMM). The most popular approaches to predict discrete probability distribution are Kalman filter [1], condensation [2], particle filter [3] and mean shift [4]. Kalman filter uses very similar idea with the HMM, while Kalman filter deals with discrete variables. Some researchers put different control and noise models into the recursion function for image processing, however those assumptions are dependent on varied applications. Condensation deals with how to sample probability and likelihood. When condensation is applied for multiple people, a dominant peak will be established if an object obtains larger likelihood values more frequently; so that some other objects may be lost. The particle filter's performance may be limited by dimensionality of state space, whereas for the cases with fewer targets, it could be a successful way. The mean-shift algorithm is efficient for tracking. However the searching window may drift away from the object. For example, if the kernel is lost in one frame under some emergency, such as illumination changes, the tracker can not recover itself from this unexpected event.

Generally, the object classification approaches apply an analysis window based on the expectation of objects features to scan across the image in search of the areas of interest (AOI) [4]. However, most conventional analysis-window trackers are influenced by the shape and size of the window, which varies from one frame to another. It is difficult to find the appropriate window for each frame, especially under dynamic environments where the content of the images may be dramatically changed.

To tackle this issue, some adaptive-window-tracking algorithms [5], [6] have been proposed to compensate the unpredictability of images. In [6], the authors treated every particle as a classifier with different parameters. Those classifiers swarmed in the solution space to converge to an optimal analysis window. Underneath each particle, brute force searching has been employed which makes the algorithm extremely time-consuming and impractical for real-time applications.

There are various people features can be used for people detection and tracking, such as skin, color, shape, texture, gesture, contour, and motion. Some successful methods also take advantage of knowledge of human figure, such as face-body relations. Skin, color, and texture features tend to be changed under dynamic illumination conditions. It is difficult for shape, gesture, contour, and face-body relations based approaches to handle the situations where people change his/her gestures in different ways, such as bowing, twisting, or squatting, or some parts of the human figure are occluded, deleted, or added.. Therefore, appearance histogram is applied in this paper as the feature cue for people detection and tracking because it is independent of human figure knowledge and robust to occlusion.

1-4244-0700-1/07/$20.00 ©2007 IEEE

. Generally, people walking can be considered as rigid movements, for instance, in surveillance system, where a blob tracker or a straight-up rectangle window is good enough. However, when people change his/her gestures, such as bowing, twisting or squatting, tracking windows should not only shift but also rotate to adapt to people's new gesture. In this sense, basically, a tracker moves in a super space of location, size, shape and orientation. Up to our knowledge, most of the available research has worked on the rigid people motion detection and tracking, but none of them is for non-rigid people motions.

This paper presents a new method for visual objects tracking using a particle swarm optimization (PSO) algorithm [7][8] to search for an optimal window which can identify both the rigid and non-rigid people motions. The PSO algorithm was inspired by the social behavior of a flock of birds. In the PSO algorithm, the birds in a flock are symbolically represented as particles. These particles can be considered as simple agents "flying" through a problem space. A particle's location in the multi-dimensional problem space represents one solution for the problem. When a particle moves to a new location, a different problem solution is generated. This solution is evaluated by a fitness function that provides a quantitative value of the solution's utility.

The PSO algorithm is effective for optimization of a wide range of searching problems. In this project, particles fly around the image, trying to find the best-fit tracking window parameters based on the fitness function of object features. When some particles successfully detect the objects, they will share this information with their neighbors, and their neighbors may follow the directions to reach objects very quickly. 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, which may cause the searching window drift.

Generally it is reasonable to assume that objects move consecutively in successive frames. So most particles can explore in a relative small region and converge in a short time. The completeness of proposed five-dimension searching space provides the tracker to be more robust for various motion structures of the object, say people. Comparing to the conventional window-tracking algorithms, PSO-based adaptive window can be updated automatically, and both rigid and non-rigid motions can be detected and tracked in a more flexible and robust way. By considering the motion constraints of the tracked object, the searching procedure can be expedited to achieve the real-time performance of the tracking system.

This paper is organized as follows. Section II introduces some related works. Section III describes the PSO algorithm. The proposed PSO-based adaptive window algorithm is discussed in section IV. In section V some experimental results are analyzed and compared with the mean shift method. Conclusion and further work are given in section VI.

II. Related Work

There are many systems proposed in past few decades for people detection and tracking. Pfinder [9] presented a real-time system for people tracking which used a multi-class statistical model of color and shape to segment a person from a background scene. It detected and tracked the people's head and hands under a wide range of viewing condition. Viola [10] proposed a boosted cascade algorithm, which is a learning method by combining a set of weak 0/1 classifiers to quickly detect face, and locate people. Olson and Brill [11] built a general purpose system for moving object detection and event recognition, where objects were detected by changing detection, and tracked by both first-order prediction and nearest neighbor matching. Song and Nevatia [12] proposed a combined face-body feature to track human actions under indoor environments, where the models based on face-body relations were used to find upper body candidates that might contain real objects. The system in [13] extracted moving targets from a real-time video stream, and classified them into pre-defined categories and tracked them.

Zhao and Nevatia [14] presented a model-based people segmentation in crowded situations. Four ellipses presenting length and fatness of head, torso and two legs were used to represent three behaviors: both legs together, left leg forward and right leg forward. Then they put this knowledge into MCMC (Markov Chain Monte Carlo) to accelerate the convergence. This complex algorithm can probable be applied for still images only due to its extensive computations. Tao, Sawhney, and Kumar [15] tried to build configurations for each target to naturally handle appearance, disappearance and occlusion. Then a local sampling stage and a global sampling stage were executed to deal with object motion, addition, and deletion separately.

Beleznail [16] adopted the fast mean shift to cluster and track people. It was efficient when objects were spatially well-separated, because it assumed offsets between frames were smaller than the kernel size. But special probability strategy was needed when occlusion happened. The Hydra system [17] was proposed to detect and track multiple people. A silhouette-based shape model, a motion model, and the correlation-based matching method were combined to make classification.

III. Particle Swarm Optimization

The PSO algorithm is an efficient optimization method proposed by Kennedy and Eberhart in 1995 [18], [19] from the simulation of a simplified social model, which obviously has its root in artificial life in general, and in bird flocking, fish schooling and swarming theory in particular. On the other hand, it is also a method of evolutionary computation, related with both genetic algorithm and evolutionary programming.

The PSO algorithm is population-based: a set of potential solutions evolves to approach a convenient solution (or several solutions) for a problem. Being an optimization method, the aim is to find a global optimum of a real-valued function (fitness function) defined in a given space (search space). Rather than a simple social simulation, PSO can be treated as a powerful new search algorithm, capable of optimizing a wide range of N-dimensional problems.

The social metaphor that led to this algorithm can be summarized as follows: the individuals that are part of a society hold an opinion that is part of a "belief space" (search space) shared by every possible individual. Individuals may modify this "opinion state" based on three factors:

- The knowledge of the environment (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 somehow 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 the 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 multidimensional space that represents the belief space. Particles have memories, 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. The goodness of a position is defined by the fitness function, whose format depends on explicit applications.

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 in iteration *t*, x_{id}^{t} is the component in dimension *d* of the *i*th particle position in iteration *t*, c_1, c_2 are constant weight factors, p_{id}^{t} is the best position achieved by particle *i*, 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, and w is the inertia weight. The PSO requires tuning of some parameters: the individual and sociality weights c_1, c_2 , and the inertia factor w .

According to (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, moving to the best position found by its neighbors. Different parameters in (1) provide varied balance among those three factors. Then a particle moves in the search space according to the combined velocity calculated by (1) to achieve a new position, which can present a new value of object features. In this paper, image features are projected into a search space, and a set of parameters of the tracking window are abstracted into a virtual point. PSO tries to find out the best virtual point which represents the optimized parameters for the tracking window.

The mechanism of PSO implicitly assumes that in most real world situations, the optima have better residence around them. Corresponding regions with higher fitness values are more possible to contain optima than areas with lower values. Experimentally, during the search, regions with higher fitness values attract more particles and make them more and more concentrated after a number of iterations. So this type of search is fast and effective. On the other hand, PSO is simpler than genetic algorithm since all particles employ the same mechanism during evolutions. Although basic PSO is designed for only single optimum, there are many works have been done to process more complex issues [20].

IV. ADAPTIVE TRACKING WINDOW

A. General Idea

Extensive work have been conducted for people detection and tracking. Basically this issue can be considered as a probability-based classification and estimation, which searches for the best match of the given target model that presents the interested object. Those algorithms can be analyzed from two views: search-space based approaches and search-window based approaches. If more features are used to describe objects, which bring more constrains, the search space becomes smaller. For image processing, features can be extracted from still image or subtracted from motion relations between frames. For search- window based approaches, there are two types of search windows: global window and distributed window. A global window takes the object as a whole so that the computation is relatively light. However, it’s hard to adjust shape, size, and orientation of a global window because of the absence of expect capability, which may degrade the system performance dramatically. For a distributed window, all of these factors are naturally hidden. But in order to drive distributed windows, the objects must have some local features to be easily identified by distributed windows.

From the view of searching, the PSO algorithm is a distributed convergence method. The key is to take advantage of sharing information between particles as well as their own past experiences to accelerate the convergence. The PSO algorithm would provide an optimal or near-optimal solution using appropriate fitness functions without the complete knowledge of the searching space. Therefore, in this project, we will use the PSO algorithm to build up auto-adaptive tracking windows in a tracking space to search for optimal parameters of the tracker. The basic idea is that particles associated with multiple parameters of a global tracking window fly around the search space, communicate and share information among their society, follow better choices of their neighbors, and converge to an optimal one.

B. The Tracking Window

To identify people in an image, rectangle windows are used here. Five parameters will be identified to describe the rectangle windows, including 2D location of the central point, length and width, and the orientation of the rectangle relative to the x-axis, as shown in Fig. 1. These parameters build up a five-dimensional search space.

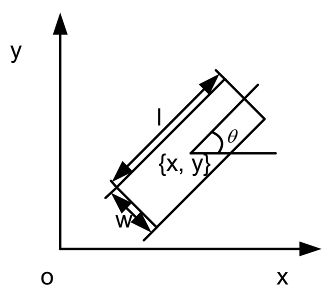


Fig.1. The five parameters associated with a particle window

So in such a space, each particle presents a search window with specific values of parameters, which can be defined as:

$$P = \{p_i \mid p_i(x_i, y_i, l_i, w_i, \theta_i), i = 1,2,...,N\} \quad (3)$$

Where x_i and 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*; θ_i denotes orientation of the rectangle in the image frame; and *N* is the population of the swarm particles. Each individual particle has different values of these parameters. In other words, they are distributed in a five-dimensional search space.

C. Fitness Function

The behaviors of the particles are guided by the associated fitness function, which defines the search criteria underlying the PSO searching algorithm. Generally target model is described by object features like shape, color, and texture. The appearance histogram is used here to describe objects due to its simplicity and efficiency. For object tracking, the value of fitness function can be considered as the similarity between the region covered by a specific search window and the target model. The goal of search algorithm is to find out the most alike region which has the highest probability to be the object. Since histogram intersection is widely used and proved to be a robust measurement for distance between histograms [21], it can be used to define the fitness function for search windows.

First, the image is transformed from RGB format into HSV format, which is more natural for people's eyes. The values of hue are abstracted to build a histogram for each window. The intersection between this histogram and the target histogram can be calculated. The fitness function is defined as the histogram intersection as followings:

$$f(x_i, y_i, l_i, w_i, \theta_i) = H(h_i, g) = h_i \cap g = norm\left(\sum_{j=1}^{256} \frac{\min(h_j, g_j)}{\max(h_j, g_j)}\right)$$
$$i = 1,2,..., N. \quad (4)$$

Where h_i and *g* represent the histogram of search window *i* and target model histogram, respectively. $H(h_i, g)$ is the histogram intersection between the h_i and *g,* which is the sum of overlaps along all possible pixel values. The normalized value is between 0 and 1. The higher the histogram intersection, the more similar the two histograms are.

D. Motion-based Constraints

Generally a five-dimensional feature space is very large, which makes search algorithms to be computation extensive. Some motion-based constraints can be applied to limit the search area to a smaller region where particles are initialized and move around. A straightforward constraint is to use the continuity of movement since it is reasonable to assume that motion is continues under most tracking situations. In other words, the most possible location of new tracking window should be close to the position of last tracking window. In this way, the initialization of particles could be accelerated. Suppose $p_{b'}(x_{b'}, y_{b'}, l_{b'}, w_{b'}, \theta_{b'})$ is the best particle (i.e., tracking window) in last frame, the initialized particles $p_i(x_i, y_i, l_i, w_i, \theta_i)$, where *i* = 1,2,…,*N*, in the new frame should be around $p_{b'}$ with some offset in each dimension. In our experiments, locations are shifted up to 15 pixels; sizes are shrunk and extended up to 20 percent; and angles are changed up to 15 degrees. Therefore, by dispersing particles in a relatively smaller region instead of the whole image, searching procedure can be definitely accelerated.

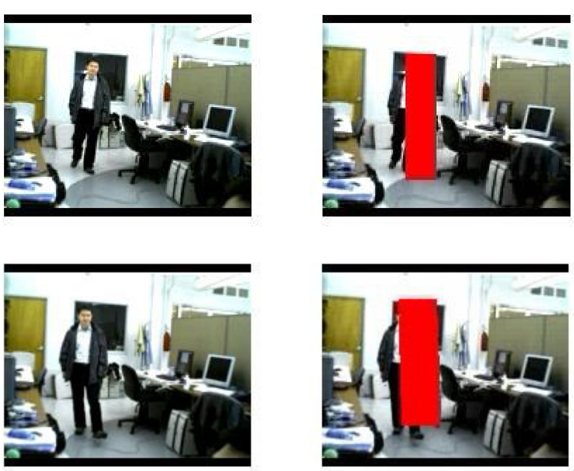
E. Adaptive Tracking Window Algorithm

Since the proposed algorithm focuses on the object tracking, we assumed that the tracking window has been detected and the associated fitness value of the target has been setup as g in the detection phase. When a new frame comes, particles which are represented by search windows with different parameters are initialized for this new image with the motion constraints. Then the fitness values of all particles are calculated using (4). These values provide the particles the local best and global best positions. The particles fly around the search space according to PSO algorithm (1) and (2) to search for an optimal tracking window based on the calculated fitness values. When a particle finds a better match, it would send this information to its neighbors to attract more particles. If the maximum of fitness value beyond the user-defined threshold, or iterations reach to a preset limitation, PSO search for current frame stops. The particle with the highest fitness value is considered to be the best candidate, and the new tracking window is defined by the associated parameters with the best candidate.

V. Experimental Results

The proposed algorithm is simulated by MATLAB 7.0 on Windows XP with a Pentium4 desktop machine. Initially, the object is marked by users with a rectangle window. To evaluate the tracking performance of the proposed PSO-based algorithm, several video clips have been tested. Since the histogram is very sensitive to the illumination change, the following experiments are conducted in an office environment with same light condition.

Some snapshots of the first clip are shown in Fig. 2, where one person is walking towards the camera. The left sequences are ground truth and the right ones are tracked by the tracking windows using the proposed algorithm. During the whole process, the person is tracked well and the window size is enlarged with the bigger object in the scenes, and is bended in an angle adapt to the object movement.



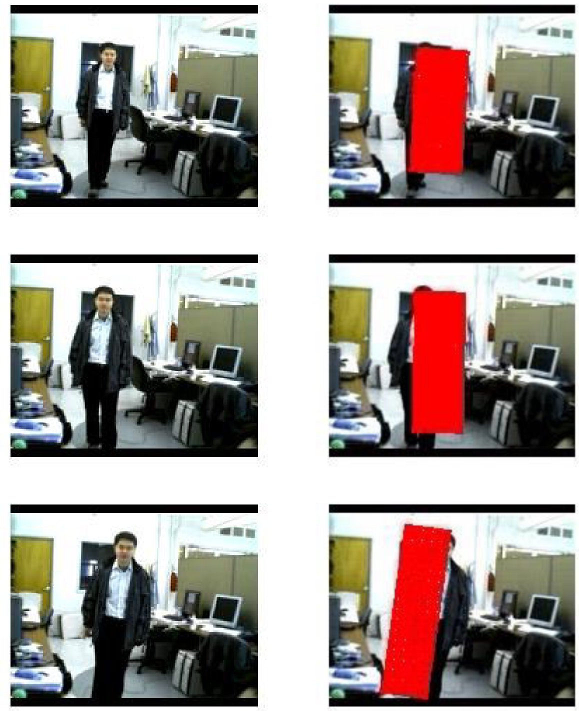


Fig.2. The snapshots of a walking people tracking example in a video clip.

Then we implemented our algorithm on the second clip, as shown in Fig.3, where a student changes his gesture from standing up to squat. From the snapshots of the video, it can be seen that when the student changes his gesture, the size and orientation of the tracking window changes to adapt to the student's behaviors.

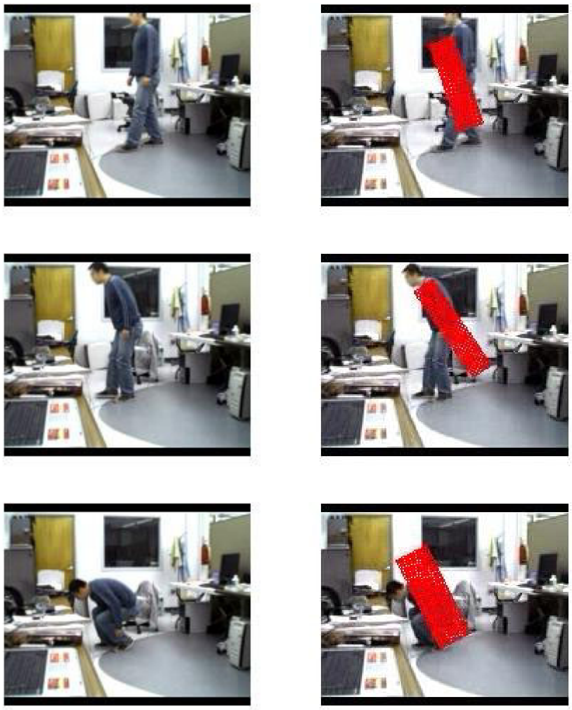


Fig.3. The snapshots of a people tracking example in a video clip with changed behaviors

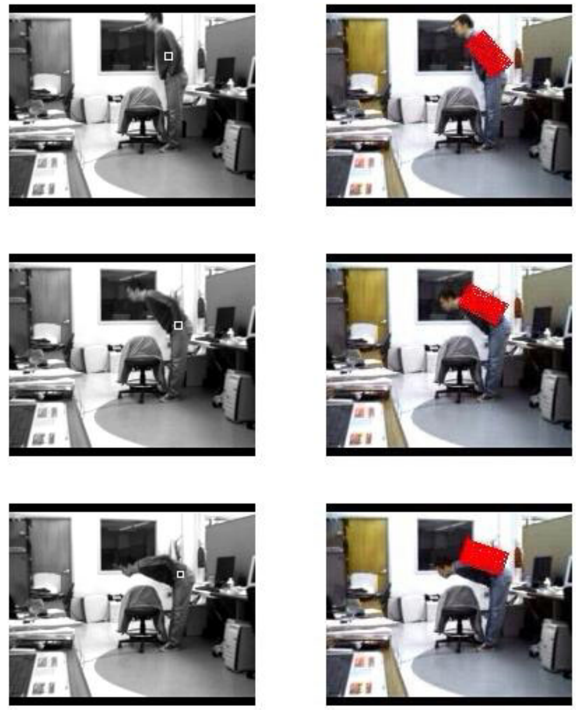


Fig.4. The snapshots of a people tracking in a video clip with body bend, compared with mean shift.

In the third video clip, as shown in Fig.4, we compare the mean shift method with our algorithm in a gesture changing scenario. A student bends his body, which is a non-rigid behavior, and we try to track his torso. In the left sequence of mean shift, we can see the center of tracking window (the small white square) shifts away from the student's upper body, while our algorithm can still track the person's motion.

VI. Conclusion and Future Work

In this paper we present an adaptive tracking window to automatically track the rigid and non-rigid people motion. First, a five-dimensional searching space is introduced to build up the tracking window. Then the PSO algorithm is employed to automatically find the best parameters of the windows to track the target. During the optimization searching, knowledge of motion is used to decrease scout region. Our method helps the tracking windows evolve more flexibly and robustly to catch the changes in the scenes. From varied results of experiments, we can see the proposed method is efficient and robust for both rigid and non-rigid people motion.

This is our first step of trying the swarming particles on people tracking problems. There are still lots of issues remained and need to be improved and extended in future work. For example, when more and more particles concentrate on some regions, they may lose the ability of exploring remote parts of the scene, especially for those complex situations where objects may appear and disappear in an unpredictably. Illumination based algorithms are inevitable affected by appearance of images. Some particles may drift away from objects and in that case a recall mechanism will be helpful. In terms of real-time performance, motion estimation may be employed to guide the particles wondering to speed up the searching.

Appearance histogram is affected by lighting conditions. So it is not robust for dynamic environments. We will try to introduce more features into our algorithm for detection, which leads to a fully autonomous system.

REFERENCES

[1] G. Welch and G. Bishop, “An Introduction to Kalman Filter”, *SIGGRAPH 2001*, Course 8, 2001.

[2] M. Isard, “Visual Motion Analysis by Probabilistic Propagation of Conditional Density”, D.Phil. Thesis, Oxford University, 1998.

[3] S. Maskell and N. Gordon, “A Tutorial on Particle Filters for On-line Nonlinear/Non-Gaussian Bayesian Tracking”, 2001.

[4] D. Comaniciu, and P. Meer, “Mean shift: a robust approach toward feature space analysis,” *IEEE Trans. Pattern Analysis Machine Intel.*, Vol. 24, No. 5, pp. 603-619, 2002.

[5] John G. Allen, Richard Y. D. Xu, and Jesse S. Jin, "Object tracking using Camshift algorithm and multiple quantized feature spaces," in *Pan-Sydney Area Workshop on Visual Information Processing* (VIP2003), Sydney. Conferences in Research and Practice in Information Technology, Vol. 36, pp. 3-7, 2003.

[6] Yuri Owechko, Swarup Medasani, and Narayan Srinivasa, "Classifier Swarms for Human Detection in Infrared Imagery,” in *2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW’04)*, Vol. 8, pp. 121, 2004.

[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 2001 IEEE service center*, Piscataway, NJ., Seoul, Korea., 2001.

[9] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. “Pfinder: Real-Time Tracking of the Human Body”, Technical Report 353, MIT Media Lab Perceptual Computer Section, 1995.

[10] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *IEEE Conf. on Computer Vision and Pattern Recognition 2001*, Vol. 1, pp. 511-518, 2001.

[11] T. Olson and F. Brill. “Moving Object Detection and Event Recognition Algorithms for Smart Cameras”, in *Proc. DARPA Image Understanding Workshop*, pp. 159-175, 1997.

[12] X. Song and R. Nevatia, “Combined Face-body Tracking in Indoor Environment,” in *Proceedings of the 17th International conference on pattern recognition (ICPR’04)*, Cambridge, U. K, Vol. IV, pp.159-162, August 2004.

[13] A. Lipton, H. Fujiyoshi, and R. Patil, “Moving Target Detection and Classification from Real-Time Video”, In *Proceedings of IEEE Workshop on Application of Computer Vision, 1998*,

[14] T. Zhao and R. Nevatia. “Bayesian human segmentation in crowded situations,” in *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 459~466, Madison, USA, June 2003.

[15] H. Tao, H. S. Sawhney and R. Kumar, “A sampling algorithm for tracking multiple objects,” in *Proc. Workshop of Vision Algorithms*, with ICCV 99, pp. 53-68, 1999.

[16] Csaba Beleznai, B. Fruhstuck, and H. Bischof, “Human Tracking By Mode Seeking,” in *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis (2005),*pp.1-6, Sept. 2005.

[17] I. Haritaoglu, D. Harwood, and L. S. Davis, “Hydra: Multiple People Detection and Tracking Using Silhouettes,” in *10th International Conference on Image Analysis and Processing (ICIAP’99)*, pp. 280-285, 1999.

[18] R. Eberhart and J. Kennedy. "A New Optimizer Using Particles Swarm Theory", in *Proc. Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan., IEEE Service Center, Piscataway, NJ, 39-43, 1995.

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

[20] Kennedy, J. and R. Eberhart, “A discrete binary version of the particle swarm algorithm,” in *Proceedings of the Conference on Systems, Man and Cybernetics*, pp. 4104-4109, Piscataway, New Jersey, IEEE Press, 1997.

[21] Brian V. Funt and Graham D. Finlayson, “Color Constant Color Indexing”, pp. 522-530, IEEE PAMI, Vol. 17, No. 5, May 1995.