



# Non-overlapping Distributed Tracking System Utilizing Particle Filter

F. L. LIM, W. LEOPUTRA AND T. TAN

*Department of Computer Science, Curtin University of Technology, GPO Box U1987, Perth, WA 6001, Australia*

*Received: 4 April 2006; Accepted: 9 February 2007*

**Abstract.** Tracking people across multiple cameras is a challenging research area in visual computing, especially when these cameras have non-overlapping field of views. The important task is to associate a current subject with other prior appearances of the same subject across time and space in a camera network. Several known techniques rely on Bayesian approaches to perform the matching task. However, these approaches do not scale well when the dimension of the problem increases; e.g. when the number of subject or possible path increases. The aim of this paper is to propose a unified tracking framework using particle filters to efficiently switch between visual tracking (field of view tracking) and track prediction (non-overlapping region tracking). The particle filter tracking system utilizes a map (known environment) to assist the tracking process when targets leave the field of view of any camera. We implemented and tested this tracking approach in an in-house multiple cameras system as well as using on-line data. Promising results were obtained which suggested the feasibility of such an approach.

**Keywords:** non-overlapping, tracking system, particle filter

## 1. Introduction

The cost of surveillance cameras are cheap, however, to install these cameras to monitor a reasonable size surveillance area require hundreds of cameras and sensors which can bring the cost of a surveillance system up to million of dollars. Nevertheless the efficiency in term of the ability of security personals to deduce a threat or even to follow up an event from these large amount of recording is almost impossible. Hence, many surveillance systems are used to gather data and to identify clues after an incident taken placed. Advances in tracking technology has opened up a possibility to detect event requiring attention as it happens to alert security personals to take further action.

In this paper we explore the task of tracking human subject across multiple cameras system. We advocate the use of a general framework, based on the particle filter, to resolve the problem of within field of view visual tracking and track prediction when the subject leave the field of view of a camera and enter another

camera resides in the non-overlapping region of the cameras system. The latter objective is accomplished using a map of the surveillance area, which provides information of possible path trajectories across the non-overlapping region. Another objective is the design of a unifying probabilistic framework that can seamlessly deal with both within field of view and out-of-view tracking.

The aims of this paper are to develop:

1. An indoor surveillance system that tracks the movements of a human subject with special emphasis on non-overlapping cameras.
2. A unifying framework to deal with both within field of view target tracking and non-overlapping targets association.

## 2. Background

There are a number of probabilistic models apply in surveillance for human tracking such as Kalman

filter, Bayesian and more recently particle filter. Kalman filter was first introduced by Kalman [14], which is a prediction model that provides an optimal solution when solving a linear problem [29]. Unfortunately, such constraint fails when Kalman filter is applied to non-linear problem with non-Gaussian noise. An example of an application uses non-linear equations is tracking irregular motions of human walking.

On the other hand, Bayesian and particle filter were emerged to solve systems with non-linear equations and non-Gaussian noise. Although Bayesian technique [10, 15, 23] is able to solve tracking problem which involves non-linear equations, for tracking problems which also involves non-Gaussian noise, a prior set of rules and conditions are required which get very complicated when dealing with large scale scenarios. Alternatively, particle filter, introduced by Gordon et al. [8], has been a preferred choice for tracking application due to its ability to solve both non-linear equations with non-Gaussian noise. Its main principle is derived from the sequential Monte Carlo method [3] that recursively generate random measurements to approximate the distribution of unknown variables.

Particle filter technique has been proved to be robust and is widely used in many applications such as robotics [2], human tracking [9, 21], vehicle tracking [20], sound detection [4], and bearing tracking [3].

### 2.1. Dynamic Model

In general, tracking using cameras is a non-linear problem. Thus, the dynamic model of a typical system can be expressed in Eqs. (1) and (2) [8]:

$$x_k = f_k(x_{k-1}, u_k) \quad (1)$$

and

$$z_k = h_k(x_k, v_k) \quad (2)$$

where  $k$  is a discrete time index,  $x_k$  is the state sequences, and  $z_k$  represents the observation measurement. The notation  $u_k$  and  $v_k$  are noise matrices. The first equation is the function that predicts the current state given the previous state and a noise vector. The second equation defines a measurement model, i.e. it determines how well the prediction of current state based on current observation. The main

objective is to recursively predict the state  $x_k$  based on historical observations of  $x_{1:n}$ , where  $n = 1..k$  [8].

### 2.2. Particle Filter

Particle filter is a probabilistic technique which is based on random measurement density approximated by a set of weighted particles. Each particle is a two-tuple consisting the state domains and its corresponding probability (weight), denoted by  $\{x_{1:k}^i, w_k^i\}_{i=1}^{N_s}$ , where  $i$  is the particle number and  $N_s$  is the number of particles. At each time instance,  $x_{1:k}^i$  represents the predicted state of the person by particle  $i$ . The predicted trajectories can be drawn from the density  $p(x_{0:k}|z_{1:k})$ , which is expressed in Eq. (3) [19]:

$$p(x_{0:k}|z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^{(i)} \delta(x_{0:k} - x_{0:k}^{(i)}) \quad (3)$$

Then, the estimation of the state observation,  $E(h(x_{1:k}))$  can be computed using Eq. (4)

$$E(h(x_{1:k})) = \sum_{i=1}^{N_s} w_k^{(i)} h(x_{1:k}^{(i)}) \quad (4)$$

The full algorithm of particle filter can be found in [8]. Basically, particle filter is a three-phase algorithm. A general framework of particle filter can be written as:

1. Sampling: the generation of particles.
2. Importance: weighting of each particle based on current observation.
3. Resample: a process to maintain the diversity of all particles according to their re-normalized weights.

**2.2.1. Stage 1: Sampling.** Sampling is a process of locating particles randomly to predict the next target's location. This is done using a "preferred" distribution, which is usually called a probability density function (pdf). The term "preferred" refers to any types of distribution that suits the application, for example normal distribution, uniform distribution, etc. and they are selected for tractability purposes. Mathematically, the sampling technique can be expressed as [19]:

$$x_k^i \sim q(x_k|x_{k-1}^i, z_k) \quad (5)$$

where  $q(x_k|x_{k-1}^i, z_k)$  is the pdf. In the context of tracking, Eq. (5) can be worded as: “how can the system predicts the target’s position, at current time period, given a knowledge of where the target might be at an instant before?” The answer is: “given an input particle at position  $(x, y)$  coordinate, the pdf is equal to the addition of some random numbers to the particle’s position of  $(x, y)$  coordinate.” This implies that the target may move randomly, but still be found by a number of particles [2].

**2.2.2. Stage 2: Importance.** In order to measure the wellness of the state’s prediction, all the particles are weighted based on how close they are to the observation. The weighting function refers to the computation on the previous weight times the probability of the real observation to the states divided by the sampling factor, which is defined in Eq. (6) and normalised in Eq. (7).

$$w_k^i = w_k^i \frac{p(z_k|x_k)p(x_k|x_{k-1})}{q(x_k|x_{k-1}, z_k)} \quad (6)$$

$$w_k^i = \frac{w_k^i}{\sum_{i=1}^{N_s} w_k^i} \quad (7)$$

In general, both *sampling* and *importance* are known as “sequential importance sampling” (SIS). The two methods are closely related, as the *sampling* procedure makes a random prediction, while the *importance* function weights the randomness into an accurate prediction based on observations.

**2.2.3. Stage 3: Resample.** Resampling procedure maintains large weight particles while removing small weight particles [19]. Over time, the weight of a number of particles will become very large while the remaining particles have value that is relatively small which are negligible. Thus, resampling stage ensure of keeping the larger-weighted particles. A suitable way to measure the need of resampling is to compute the effective sample size,  $\widehat{N}_{eff}$ , expressed in Eq. (8).

$$\widehat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_s} (w_k^i)^2} \quad (8)$$

During implementation, a threshold  $N_{TH}$ , was assigned to check the effective sample size. If

$(\widehat{N}_{eff} < N_{TH})$  then the majority particles carry relatively small weights, which means resampling is required.

### 2.3. Alternative Types of Particle Filter

Instead of making use of generic particle filter, some researchers modified the resampling method of the generic particle filter. An example is the auxiliary particle filter/sampling importance resampling (ASIR) [19]. The ASIR has an additional parameter in the generic sampling importance resampling algorithm. The filter can be derived as an importance density  $q(x_k, i|z_{1:k})$  which samples the pair  $\{x_k^i, i^j\}_{j=1}^{N_s}$ , where  $i^j$  refers to the index of a particle at time  $k-1$  [19]. The full algorithm of an ASIR can be found in [19].

The advantage of using ASIR filter as compared to generic particle filter is that it naturally generates points from the sample at time  $k-1$ , in conjunction with the current measurement. Whereas, in the case of generic particle filter, it randomly generate the sample at time  $k$ . Thus, ASIR makes more accurate prediction through its sampling method [19].

Particle filter has also been integrated with other techniques such as particle filter with colour analysis and particle filter with Adaboost. Okuma et al. [21] proposed a mixture technique of particle filter with Adaboost called “boosted particle filter” (BPF) to better solve the problem of splitting and merging. The proposed distribution function is implemented in Eq. (9).

$$\begin{aligned} \text{Newposterior} &= \alpha \times \text{AdaBoostproposal} + (1 - \alpha) \\ &\times \text{ParticleFilterdistribution} \end{aligned} \quad (9)$$

where  $\alpha$  is a parameter to weight dynamically between the particle filter and Adaboost. For example when  $\alpha = 0$ , particle filter distribution is used. As  $\alpha$  increases, the importance of Adaboost detections increases. As a result, the value of  $\alpha$  is adapted depends on the tracking situation, such as cross overs, collisions, and occlusions.

The proposed method has been applied in a hockey game environment. The goal is to track various number of hockey players on a sequence of digitized video. The results show BPF successfully detect objects after a cluttered individual split from the merging where Adaboost alone fails.

Nummiaro et al. [20] integrate a colour distribution into particle filter. The method employs a mixture of particle filter model with colour histo-

gram to perform multiple hypotheses. Initially, an RGB (red-green-blue) colour histograms of  $8 \times 8 \times 8$  bins was chosen as the colour distribution model. The colour measurement for an object is based on region weighting. That is pixels that are further away from the center of the region are assigned to a smaller weights, as compared to pixels that are closer to the center of the region. In order to measure the two distributions, Nummiaro et al. use the Bhattacharyya coefficient [20]. Overall, the colour based particle filter model can be expressed in Eq. (10):

$$\begin{aligned} \text{Newposterior} &= (1 - \alpha) \times \text{ParticleFilterdistribution} \\ &+ \alpha \times \text{ColourDistributionModel} \end{aligned} \quad (10)$$

Similar to the previous method proposed by [21],  $\alpha$  represents the weights contribution to adjust the importance between the use of particle filter and colour distribution. This system has applied in a traffic tracking surveillance system [20].

#### 2.4. Related Work

Recent studies show a great research attention on tracking moving objects using camera(s). For examples condensation method by Andrew and Isard [1] to track image contours, the mean-shift tracking by Comaniciu and Ramesh [5] to tracks objects based on its colour, the adaptive colour histogram model [28] to track object using histogram, and Boujou's method (<http://www.2d3.com/>) that tracks point features. In terms of human tracking, basically there are two different approaches in the literature. The first approach is on human tracking using overlapping cameras, while the second approach is tracking using non-overlapping cameras.

**2.4.1. Overlapping Tracking System.** A number of applications have used the technique of particle filter, due to its dynamic prediction model in solving non-linear equation and non-Gaussian noise. An example is on bearing tracking application [3]. In this application, Bolic [3] implemented particle filter by considering the *state* sequence in terms of angle. The ability to predict such parameter is then used to track the object position. In his Ph.D. thesis, he also proposed some improvement of the resampling method on particle filter.

Bererton [2] presented the implementation of particle filter in a shooter game application that consists of a map, player character(s) and non-player character(s). The aim is to have non-player character to search for the map and eliminate the player character. The prediction on finding objects is implemented using particle filter. The idea of using map information has been integrated into this project to perform non-overlapping region tracking.

A recent system, that uses particle filter for tracking application, is presented by Checka et al. [4]. The work combines both sound and vision to track multiple people. particle filter technique has been applied on the audio and video state components. The system is able to provide better prediction for multiple targets in a cluttered scene. In terms of human tracking, several systems were implemented [11, 12, 16] which successfully track people using multiple cameras with overlapping field of view.

#### 2.5. Non-overlapping Tracking System

As the scale of the environment increases, it is not feasible to have all areas monitored by cameras due to monitoring cost and tracking efficiency [17]. Hence, researches focus their attention into the non-overlapping region tracking. Several systems have been proposed to solve the non-overlapping cameras tracking, for example Kettner and Zabih [15]; Porikli and Divakaran [23] use Bayesian approach alone to tackle non overlapping region tracking. Kettner and Zabih [15] presented the use of Bayesian technique with prior information, such as the set of allowable paths and the transition technique, to track people using non-overlapping cameras. Furthermore, Porikli and Divakaran [23] presented a framework that includes Bayesian technique to perform an inter-camera target-correspondence. However, when the size of the environment gets larger, the complexity of the transition probability computation was dramatically increased.

Vehicle tracking is one example of the non-overlapping region tracking, where cars are identified by the camera based on their appearance and positional information. Huang and Russell [10]; Pasula et al. [22] tackle this problem using the appearance model. Huang and Russell [10] combined Bayesian technique and the appearance model to track the car, and the work was further extended by Pasula et al. [22] to solve the problem using

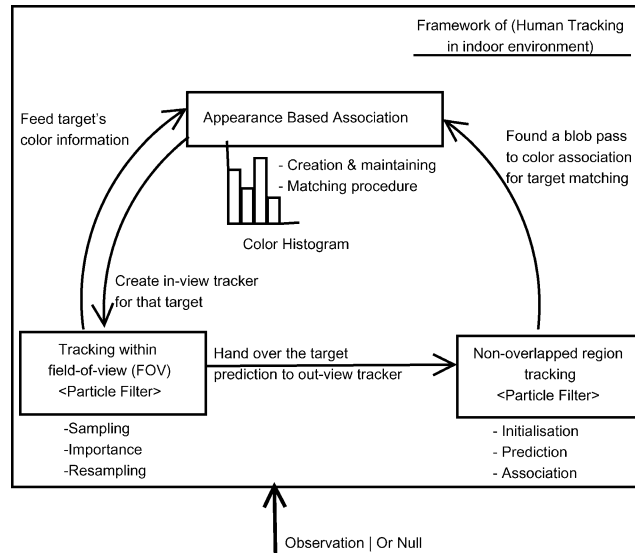


Figure 1. Graphical framework of the overall system.

Markov Chain Monte Carlo (MCMC) technique. In general, vehicle tracking follows a well-defined path that makes the problem easier to tackle.

When the probabilistic transition prior can be supplied into the system, Javed et al. [13] used that information to match the target moving model. On the other hand, Makris et al. [18] do not use prior information. Instead, they proposed a new technique for learning the accumulation information of histogram from the object coming in and out of a room, over a certain period of time which is then used to predict the target's motion.

Rahimi et al. [24, 25] proposed a technique on the non-overlapping region tracking by estimating the target's position. In the implementation, the distances between cameras are known, along with an assumption that the speed of the person is unknown but constant, which is not suitable for the real-world situation.

Recent technique, presented by Dick and Brooks [6, 7], uses hidden Markov model to model the motion of targets within and between all the cameras. Each camera, in the system, is initially trained to learn about the environment. During the operation, the system has been tested to track multiple targets under the non-overlapping situation.

### 3. Methodology

Tracking mainly involves two major steps: segmentation process (object detection) and object association

(identification). The focus of this project is on the second issue, i.e. to be able to identify and track the targets in an indoor environment. A typical indoor environment is characterised with a number of rooms, of which some are installed with cameras. The challenge of this project is to track the target using cameras that the field of view are non overlap to each others.

Figure 1 shows the overview of the project framework. The three main procedures are:

1. The non-overlapping region tracking (NORT)
2. Tracking within field of view and
3. Appearance based association

Each procedure will be discussed in the following subsections.

#### 3.1. Non-overlapping Region Tracking

A new approach to the non-overlapping region tracking using particle filter is proposed. This technique composes of three major phases: initialisation, prediction and association. In the initialisation phase, a prior map information is loaded into the system. In the second phase, when a target leaves the camera field of view, the prediction model predicts where the person might move under the non-measurement area, based on the map information. When the prediction model catches a person on one of the camera field of view, then the association

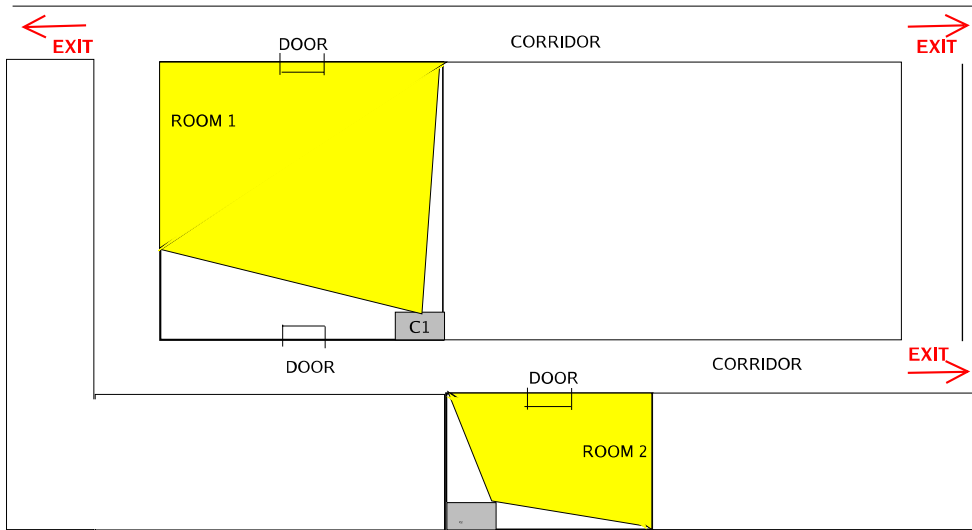


Figure 2. A typical indoor environment that consists of rooms, corridors and two cameras.

phase is performed, i.e. to identify the person with a set of people recorded in the database.

**3.1.1. Initialisation (Maps).** Initially, the map of the environment is loaded into the system. This forms the basis input for the particle filter. Figure 2 shows a typical indoor surveillance environment, where two cameras are mounted at Room 1 and Room 2. This environment will be used throughout the rest of this paper. The map defines the possible trajectories of the target after it has left the camera's

field of view. Two entities that come along with this definition are the “area definitions” and the “door regions”. The area definitions determine how the particles are to be sampled at a particular region while the door regions determine the entry/exit points that particles may flow through when approached. The “EXIT” marked target possible exit points leaving the surveillance area. Figure 3 illustrates the meaning of area definitions and door regions. Door regions are denoted by numbers 6, 7, and 11 whereas other numbers denoted area definitions.

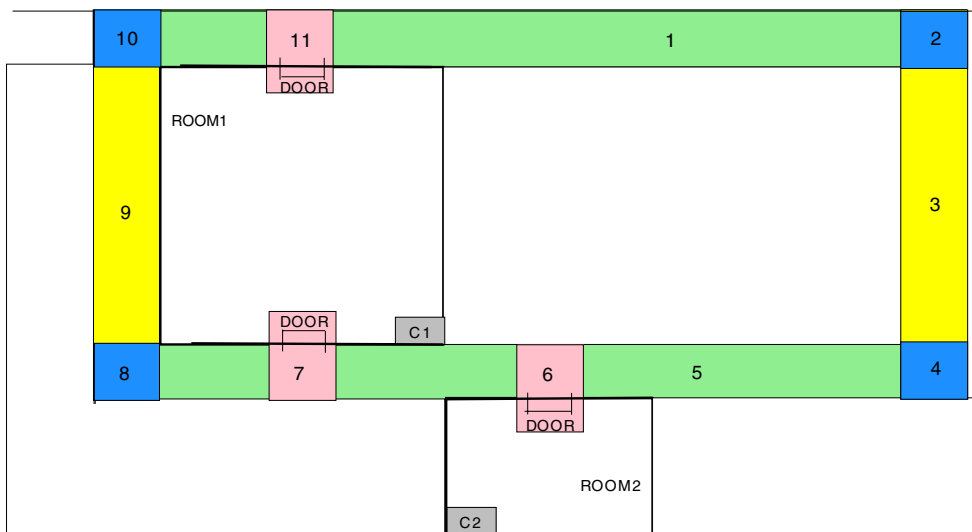


Figure 3. Area definition and door region are shown.

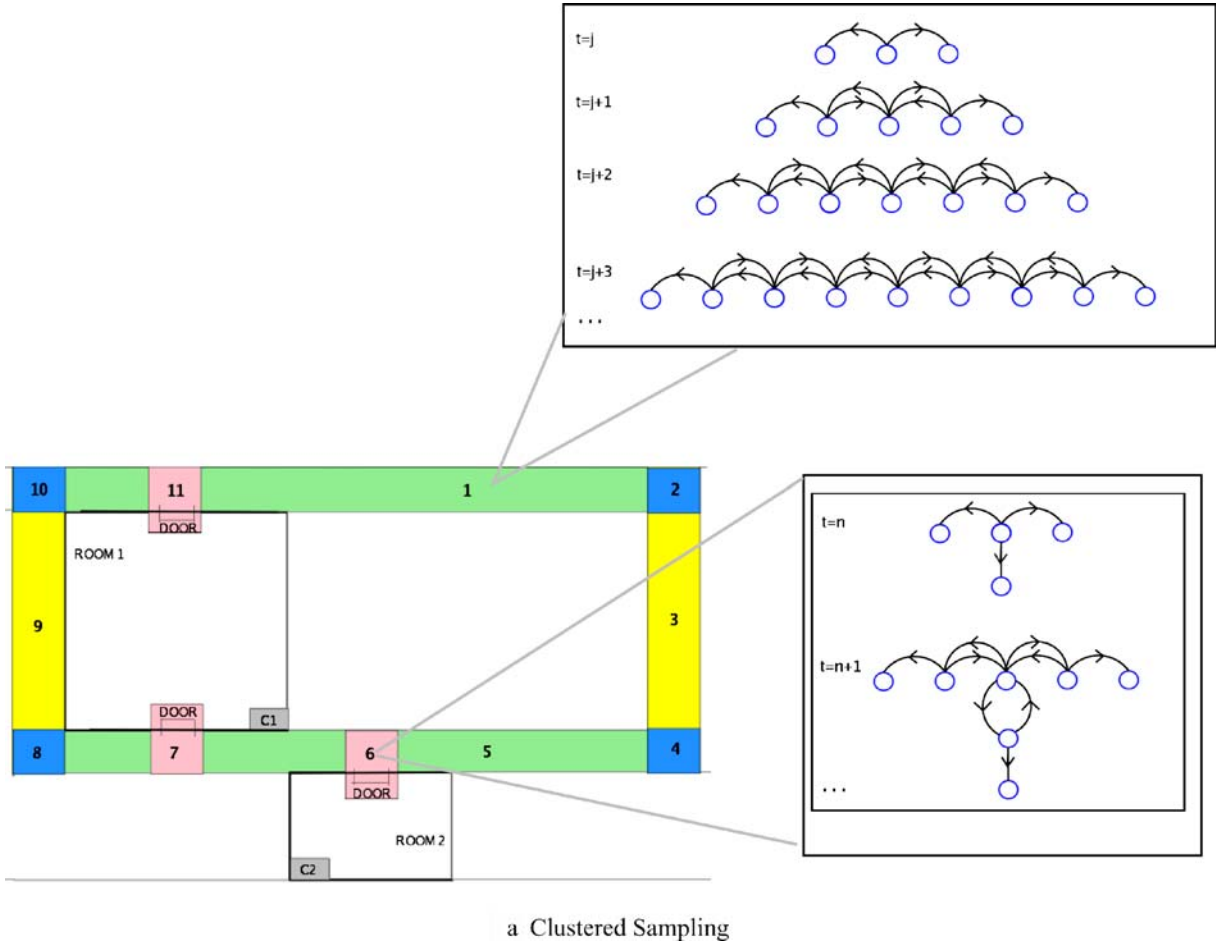


Figure 4. An example of clustered sampling, using example of area definition 1 and door region 6.

**3.1.2. Prediction.** This section describes the concept of using the map information to better predict the capability of the particle filter. The sampling method is done through a concept called cluster (grouping). Particles that are located in roughly the same position are considered as one cluster. At each time instance, particles in a cluster will be sampled in different directions based on their position in the map.

In term of mathematical notation, Let  $C_k$  be a set of cluster  $C_k = \{C_k^1, C_k^2, \dots, C_k^n\}$ , where  $n$  is the total number of clusters at time  $k$ . Each cluster,  $j(C_k^j)$ , will contain a set of particles that resides in the same position, and the number of the elements inside the cluster can be defined as a function of  $size(C_k^j)$ . The new sampling procedure can be expressed in Eq. (11):

$$x_k^i \sim q(x_k | x_{k-1}^i, m) \sim q(x_k | x_{k-1}^i \in C_k^j, m) \quad (11)$$

where  $m$  is the map information and  $C_k^j$  is the  $j^{th}$  cluster that contains particle  $i$ . Note that in the sampling method of Eq. (11), there is no measurement  $z_k$  and the addition of  $m$  presents, as compared to the generic particle filter. This implies that no measurement is available and the particles are sampled based on the cluster and the map. Figure 4 shows the particle movement which cover all possible paths a target could take within the surveillance area.

**3.1.3. Association.** There are two conditions to stop the prediction of the particle filter. First, over a long period of time, the prediction of the particle filter on the non-overlapping area becomes less accurate, due to different possibilities of the person walking trajectories. For example a target may have left the surveillance area. A *timeout* is used to control the allowable prediction duration.

Secondly, to deal with positional errors in the tracking, we define the “gradient boundary” that we position in all entry/exit of a room. In our implementation, a gradient boundary of 100 cm in radius is used to deal with uncertainty in the tracking results. When a target enters this boundary region, the system will proceed to detect the target by analysing the video output of the associated camera. When a target is detected by the system, target association will be performed.

During target association phase, particle filter is used to keep track of the target trajectory. The tracking system uses mixture of Gaussian (MoG) to perform segmentation [26]. The target’s colour information, in term of RGB values, are stored in a 3D colour histogram. Over time, the colour histogram of a target is used to represent the appearance profile of the target. The tracking system uses histogram intersection techniques [27] to associate currently detected target’s colour histogram against stored models’ colour histogram in the database. A matching score is computed where models in the database whose score is above a nominated threshold will be considered. The models with the highest score is used to confirm the target. In cases where there is no model matching the target, the target will be assigned as “new target”.

### 3.2. Tracking Within Field of View

The tracking within FOV is implemented using a generic particle filter. Algorithm 1 shows the implementation of SIS for tracking within FOV.

```

Import : Previous state  $X_{k-1}^i = [x, y, Vx, Vy]$ , Current Observation  $Z_k = [z_x, z_y]$ 
Export : Current state  $X_k^i$ 
for  $i \leftarrow 1$  to  $N_s$  do
   $x_k^i = x_{k-1}^i + Vx_{k-1}^i + Nx_k^i$ 
   $y_k^i = y_{k-1}^i + Vy_{k-1}^i + Ny_k^i$ 
end
for  $i \leftarrow 1$  to  $N_s$  do
   $pZ_i X_i = \text{EuclideanDistance}(x_k^i, y_k^i, z_x, z_y)$ 
   $w_k^i = w_{k-1}^i \times e^{(-1 \times pZ_i X_i)}$ 
end
for  $i \leftarrow 1$  to  $N_s$  do
   $w_k^i = \frac{w_{k-1}^i}{\sum_{j=1}^{N_s} w_{k-1}^j}$ 
end

```

**3.2.1. Stage 1: Sampling.** The parameters in algorithm 1 can be interpreted as follows: let’s define  $i$  as the particle number,  $k$  as the time, and the state as  $X$  that consists of four tuples  $[x, y, Vx, Vy]$ , i.e. the  $x$  and  $y$  coordinate position, and  $Vx, Vy$  are the velocity respectively. The length of a step that a

target can take is assumed to range from 0 (stop) to 150 cm. These values are obtained by measuring a normal height person (assuming 1.5 m in height) has a walking step of 50 cm and running step of 100 cm. Hence, it is reasonable to have an upper range of 150 cm to cover all types of walking patterns. The  $Nx_k^i$  and  $Ny_k^i$  are the random number noise with a specific value of either negative (move backward) or positive (move forward). The noises are used to specify the state randomness, including the direction of the particle. The observation is defined as  $Z_k$  which contains two tuple i.e. the observation position  $(z_x, z_y)$ . Note that the use of  $X$  as the state sequences is to avoid the confusion with the  $x$  coordinate.

Refer to page 4 of this paper, the *Sampling* in Eq. (5) requires a  $q(x_k | x_{k-1}^i, z_k)$  function that sampled the particles at current point to arrive at the next one. This can be put into question as: “Where will the target be at current time period, given that we have knowledge on its position at previous time frame?” The answer is implemented in algorithm 1. The first loop represents the sampling step, i.e. by assigning the new state of each particle as the addition of previous state with the velocity and random noise. As a result, the new state are scattered randomly, in term of speeds and directions.

**3.2.2. Stage 2: Importance.** After each sampling, the particle will be assigned a new weight accordingly to the observation. In the implementation, the input observation is the target that is closest to all the particles. The reason is that the possibility of the current target’s position should be close enough to the previous target’s position, except when *occlusion* occurs. In the implementation, we will first process the foreground objects that are obtained from segmentation process. That is, to filter out between the objects and the noise. The assumption we made is: objects that are less than  $\frac{1}{4}$  of a normal height (as before, assuming 1.5 meters in height) person are considered as noise. Another assumption is we only process object when they enter the surveillance area. Furthermore, we will wait until the object is fully seen to enhance our association once it appears in the scene.

The importance can be expressed as: *given what I observed, update my current belief state*. The states closer to the observation will have higher weights than those far away. This is done through the use of an exponential function,  $e^{(-x)}$  (for  $x \leq 0$ ). In algorithm 1, first, we calculate the *EuclideanDistance* between each

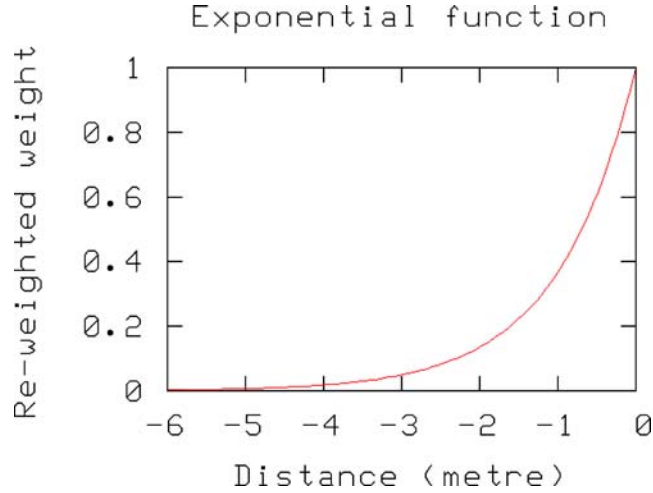


Figure 5. Exponential function for modeling the weighting of particles.

particles' position with comparison to the observed location. Then, the current weight of a particle is derived from the previous weight multiplied by the exponential function of the distance (for  $x \leq 0$ ).

The behaviour of an exponential function with respect to ( $x \leq 0$ ) is shown in Fig. 5. From this figure, the  $y$  value range between 0 and 1 which indicates the weight of a particle. We implement the  $x$  value as the distance between the particles to the observation. This leads to the further the distance, the smaller the weight (as the  $y$  value reaches 0). On the other hand, particles that are close to the observation blob will carry higher weight, i.e. the  $y$  value close to 1.

**3.2.3. Stage 3: Resampling.** Recall that the resampling process is to maintain the diversity of the particles, i.e. to keep only the particle with the higher weights. The following discusses a way to measure the need of resampling and the resampling procedure.

The need of resampling is computed through the effective sample size,  $N_{eff}$ . The equation of the effective sample size can be expressed as  $\widehat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_s} (w_i^k)^2}$ . From this equation, we observe that the value of  $\widehat{N}_{eff}$  gets larger when the particles are of equal size. On the other hand,  $\widehat{N}_{eff}$  will employ a small value when all particles are scattered, which implies that the particles need to be resampled.

During the implementation, a value of  $N_{TH}$  is chosen as the threshold. A value of ( $\widehat{N}_{eff} < N_{TH}$ ) means the majority of the particles are carrying

relatively small weights where resampling is required. The larger the value of  $N_{TH}$ , the higher the resampling rate. Note that the value of  $N_{TH}$  depends on the number of particles. In the implementation, we choose to use 500 particles to track one person for non-overlapping region tracking while  $N_{TH}$  is set to 25 for within FOV tracking. These values are chosen through trial and errors during experiments.

### 3.3. Appearance Based Association

The third procedure of this paper consists of a colour matching technique. The features used in the tracking process are velocity and colour. The colour matching technique has the following three major phases:

- Histogram creation
- Histogram matching
- History management

When a target is in the FOV, particle filter is used to keep track of its trajectory. At the same time, the target's colour information is obtained, which is represented in RGB values. This value is stored in a 3D colour histogram (CHist[r,g,b]), i.e. an  $8 \times 8 \times 8$  histogram. Over time, these colour histograms is used to represent the appearance profile of the person.

A "main database" is used to store all the target's color histograms. In the experiment, up to 20 histograms were kept for each person. The main



Figure 6. The two rooms under surveillance.

database is divided into three sub databases. The first sub database contains all the targets currently in the camera field of view. The second sub database contains the target that is under particle filter's prediction phase. The entries in the second sub database will be used when a new target reappears in one of the camera field of view. The third sub database stores only the log of the target that has left the surveillance environment.

Recall that the appearance based association will only be used under two conditions. That is when a target leaves one room and reappears in another camera field of view, or when a number of individual objects split after merging within the field of view of the camera. A histogram intersection technique is implemented [27], which returns a matching score determining the similarity or dissimilarity between two histograms. This matching score has a value between 0 (dissimilar) to 1 (similar) inclusively. We set a threshold of 0.7 for the matching histograms. When a target is checked against the model in the second sub database, the highest probability of matching score between all the models in database is used to confirm the target.

#### 4. Experimental Environment

The environment under surveillance consists of two rooms and corridors. Figure 2 (page 9) shows the layout of the experimental environment, while Fig. 6 are images captured by the cameras C1 and C2 respectively. Note that, room 1 has two entry/exit doors while room 2 has one entry/exit door.

The cameras C1 and C2 have the following characteristics:

- Each camera is a static calibrated camera, without pan/tilt/zoom functionality.
- Each camera is connected to one computer from which it can be controlled.
- The cameras were setup to record a video stream of size  $320 \times 240$  at 25 frames per second (fps).

##### 4.1. Experimental Objectives

The main objectives of the experiments are to test the:

- Performance of particle filter on non-overlapping region tracking.
- Performance of particle filter within field of view tracking.
- Accuracy of the implemented framework, i.e. to associate targets at different locations at different time frame within the surveillance area.

##### 4.2. Image Acquisition, Processing and Output

The tracking algorithm is implemented using C++ programming language on an Intel Pentium(R) 4 CPU 3.0 GHz computer, under the Linux environment. At the end of the program execution, an output video is obtained. This video comprises the map (top left part of Fig. 7), the two rooms (bottom left part of Fig. 7), and a list of targets' activities showing transactions of targets entering and leaving room 1 and 2 (right hand side of Fig. 7). Each transaction records the location of the targets (which room the targets are currently

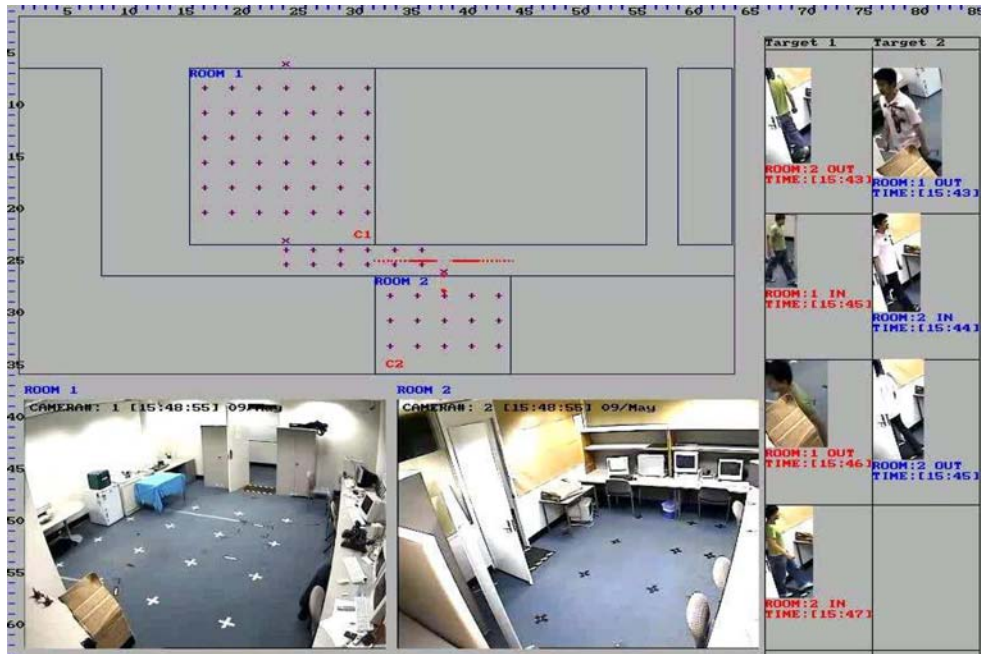


Figure 7. An example of a snapshot after program execution.

in), the type of activities (leaving (OUT)/appearing (IN)), and the time these activities were carried out. In addition, a snapshot of the targets was captured, showing targets leaving/appearing in room 1 and room 2. Each transaction is saved into a log file. The log file provides a way to thoroughly check the individual target activities, which can be used to evaluate the reliability of the tracking system.

### 5. Experimental Data Sets

The experimental data sets consist of a series of people walking patterns. Basically, it consists of people moving from one room to another room passing different paths. In addition, irregular motions of merging and splitting were introduced in within

field of view tracking. Also, appearance based tracking is demonstrated. The following subsections present the three sets of experiments.

#### 5.1. First Experiment: Non-overlapping Region Tracking

The first set of experiment involves tracking two people (*Person-1* and *Person-2*) moving from one room to another. The sequence of movements is shown in Fig. 8. Initially, Person-1, wearing a green shirt, entered room 1 using door 2. Twenty frames later, Person-2 went into room 2. Person-1 wondered for several seconds in room 1 and left the room using door 1. At the mean time, Person-2 stayed in room 2 for 210 frames. After a few seconds, Person-1

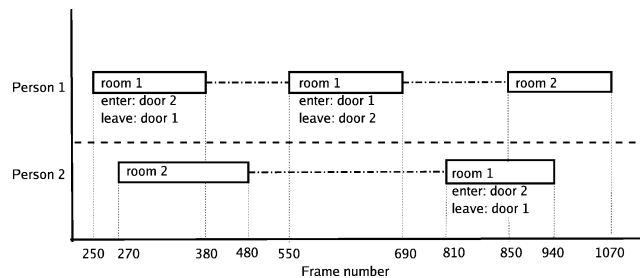


Figure 8. Overview of non-overlapping region tracking.

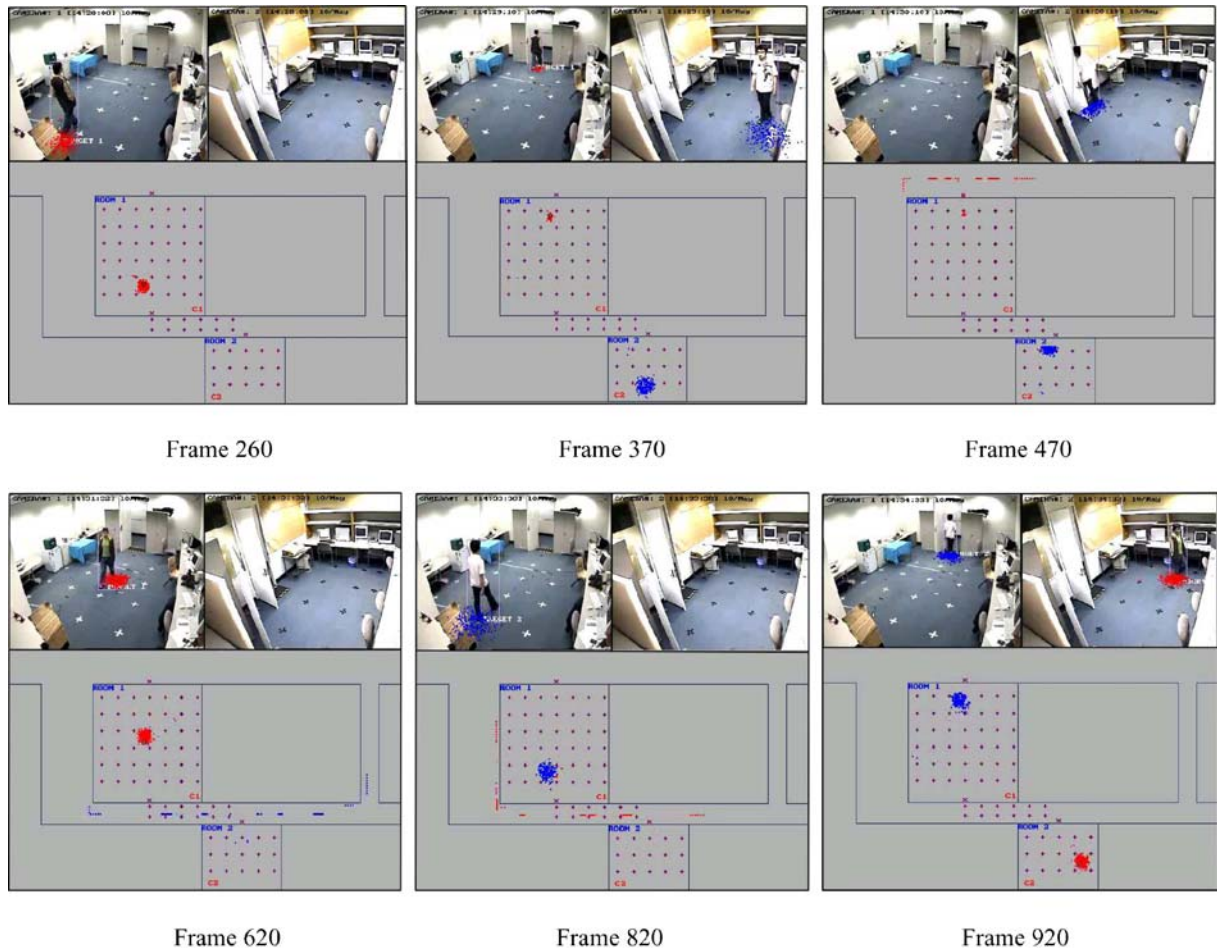


Figure 9. A selected number of frames for non-overlapping region tracking.

walked back to room 1 using door 1. He then walked straight and left the room using door 2. At frame 810, Person-2 walked into room 1 using door 2. At frame 850, Person-1 went into room 2. Person-1 and Person-2 left the camera FOV at frame 940 and 1,070 respectively.

A selected number of frames from the experiment is shown in Fig. 9 while the complete sequence is available at <http://www.computing.edu.au/limfl/journal.shtml>. This experiment shows that particle filter with known environment (map) is able to track a number of targets for non-overlapping region tracking.

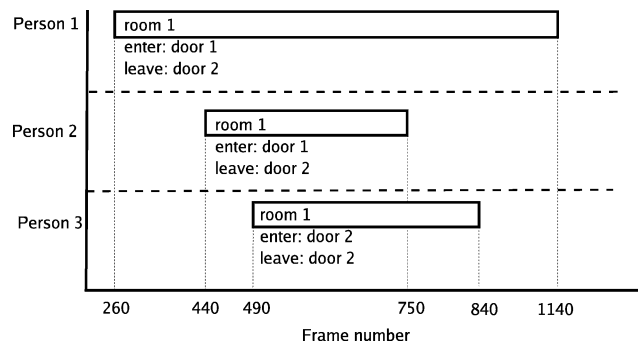


Figure 10. Overview of the merging and splitting for scenario 1.

5.2. Second Experiment: Tracking Within FOV

The second set of experiments on within FOV tracking involve walking pattern of merging and splitting. The first example was recorded in an in house surveillance area while the second example is taken from on-line data (<http://www.groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA/>).

5.2.1. Merge and Split Scenario 1. This experiment was conducted to deal with walking pattern of

merging and splitting involving three people. The challenge of this experiment is to show a successful scenario when tracking multiple targets that occlude one another for a short period of time. The sequence of movements is shown in Fig. 10. Figure 11 shows a number of frames and the complete sequence of the experiment is available at <http://www.computing.edu.au/limfl/journal.shtml>. Note that merging occurs from frame 500 to frame 660, while splitting occur at frame 680. During merging, as for non-overlapping region tracking, particles were scattered around

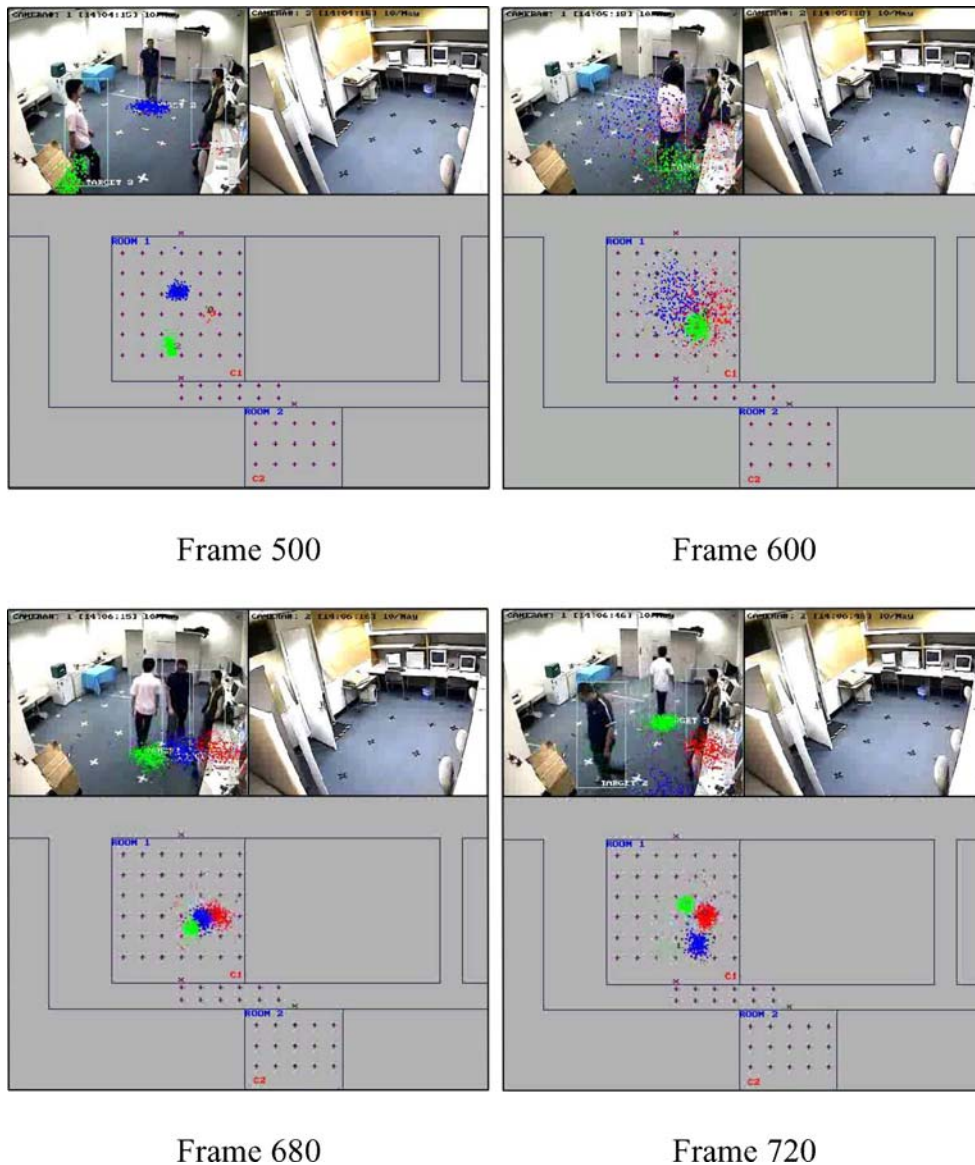


Figure 11. A number of frames extracted from the sequence of Merging and Splitting for Scenario 1.

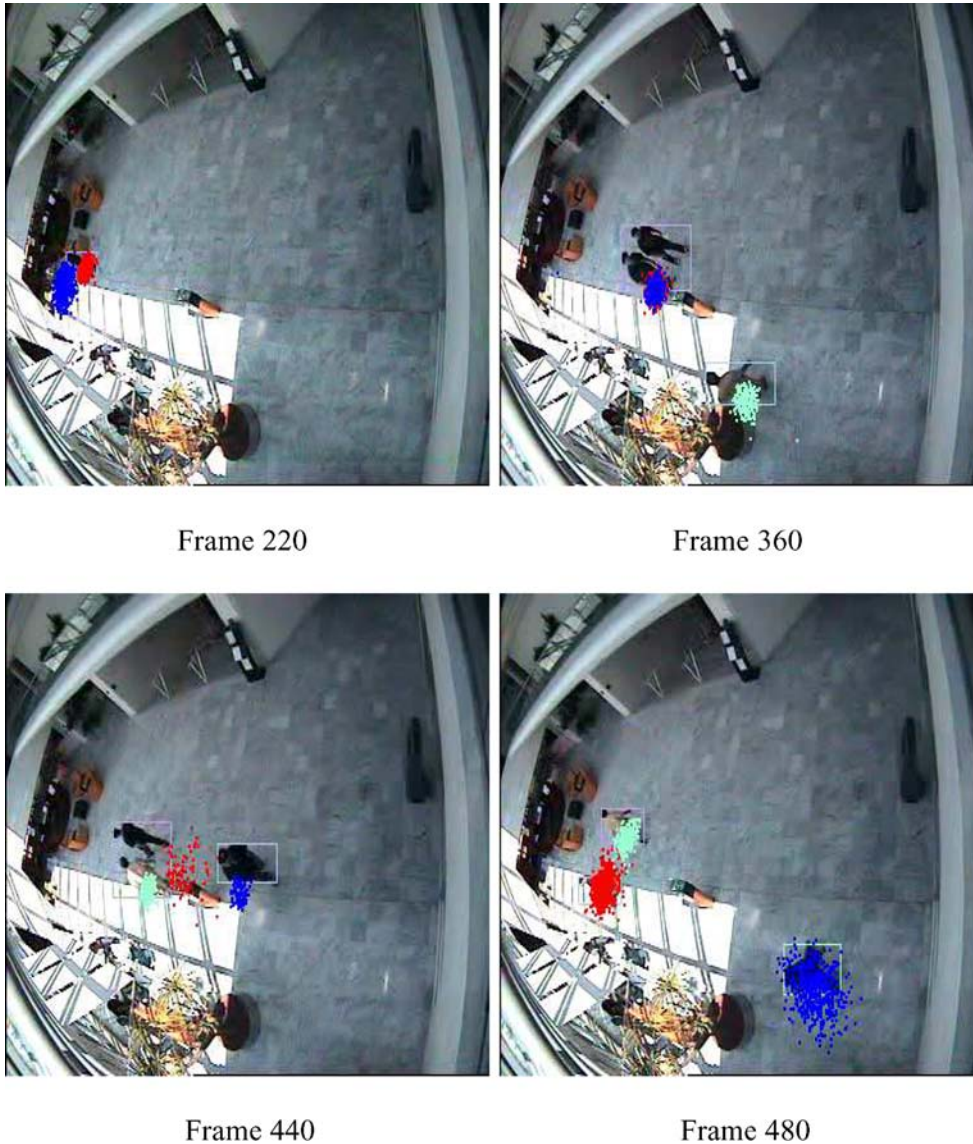


Figure 12. A number of frames extracted from the sequence of Merging and Splitting for Scenario 2.

targets without performing prediction as shown in frame 600 (see Fig. 11). When the targets split, then an appearance based association is used to identify each individual target, as shown in frame 680 (see Fig. 11). This experiment shows that particle filter is able to handle cases of merging and splitting which involve non-linear equations.

**5.2.2. Merge and Split Scenario 2.** This experiment was conducted to further justify the efficiency of the tracking system in handling multiple targets tracking

where targets may occlude one another for a short period of time. Figure 12 shows a number of frames while the complete sequence of the experiment is available at <http://www.computing.edu.au/limfl/journal.shtml>.

### 5.3. Third Experiment: Appearance Based Association

The third experiment investigates the tracking of two people who have similar colour profile. The main

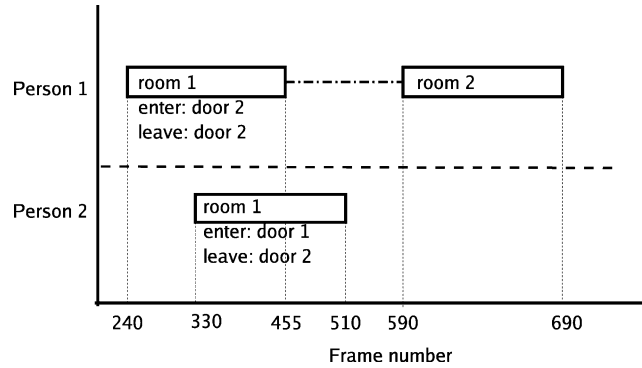


Figure 13. Overview of appearance based association.

objective of this experiment is to show the need of integrating particle filter and appearance based association technique in order to successfully track two targets with similar colour profile. Note that by using appearance based association technique alone in this scenario would fail. Figure 13 shows the sequence of movements while Fig. 14 shows a number of frames extracted from the experiment. The complete sequence of the experiment is available at <http://www.computing.edu.au/limfl/journal.shtml>.

### 6. Performance Measures

In order to evaluate the project performances of our system, we present a quantitative measurement (precision and recall) as the metric, i.e. by calculating the success and un-success rates for the non-overlapping region tracking. Success and un-success rates refer to the success of the system in matching targets’ “leaving” and “entering” the surveillance area. Table 1 shows how the metric is calculated.

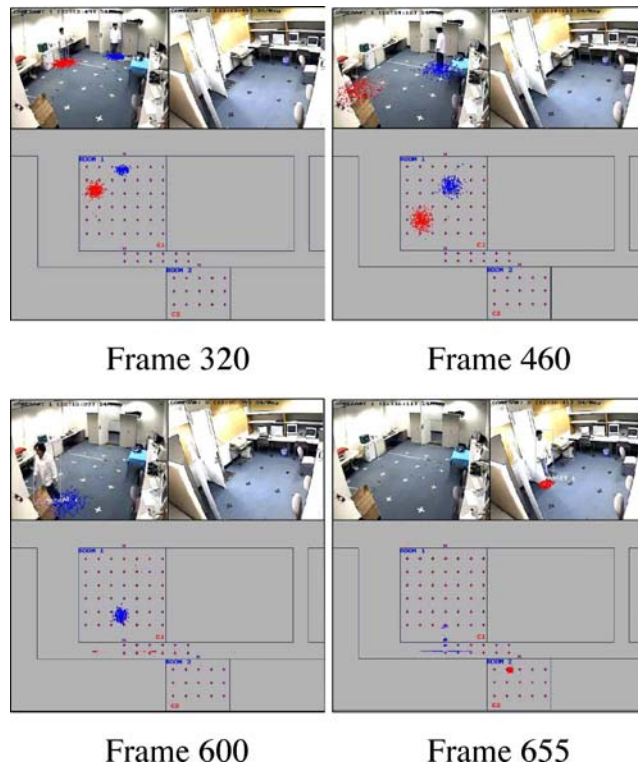


Figure 14. A number of frames extracted from the appearance based association scenario.

Table 1. Target matching procedure.

Missed	False alarm (subset of Missed)	Hit
$L$	$K$	
$N$		$M$

Figure 15 illustrates a pictorial definitions of the terms *Hit*, *Missed*, and *False Alarm*. Figure 15a shows an example of a Hit. Initially, Person-1 entered Room 1, which was captured by camera C1. After leaving Room 1, Person-1 then went to Room 2. If the system is capable of identifying the target as Person-1, then it is considered as a hit. Next, Fig. 15b shows the same scenario as the previous example, except when Person-1 entered Room 2, the system identifies the person as a new person (Person-2) which illustrates a missed case. Finally, Fig. 15c demonstrates a false alarm example. In this case, Person-1 entered Room 1 and its profile is updated on the system. After leaving the room, Person-3 with roughly the same colour profile

entered Room 2. If camera C2 identifies the person as Person-1, then it is classified as a false alarm. The assurance of the quality measurement depends on:

$$\begin{aligned}
 \text{SuccessRate} &= \frac{M}{M+N} (\% \text{ of success}) \\
 \text{FalsealarmRate} &= \frac{K}{M+N} (\% \text{ of falsealarm}) \\
 \text{Un - successRate} &= \frac{K+L}{M+N} (\% \text{ of un - successrate})
 \end{aligned}
 \tag{12}$$

The following subsections calculate the performance measure for both non-overlapping region tracking and tracking within FOV.

### 6.1. Performance Measure for Non-overlapping Region Tracking

For non-overlapping region tracking, the numbers of particles required are computed according to the size of

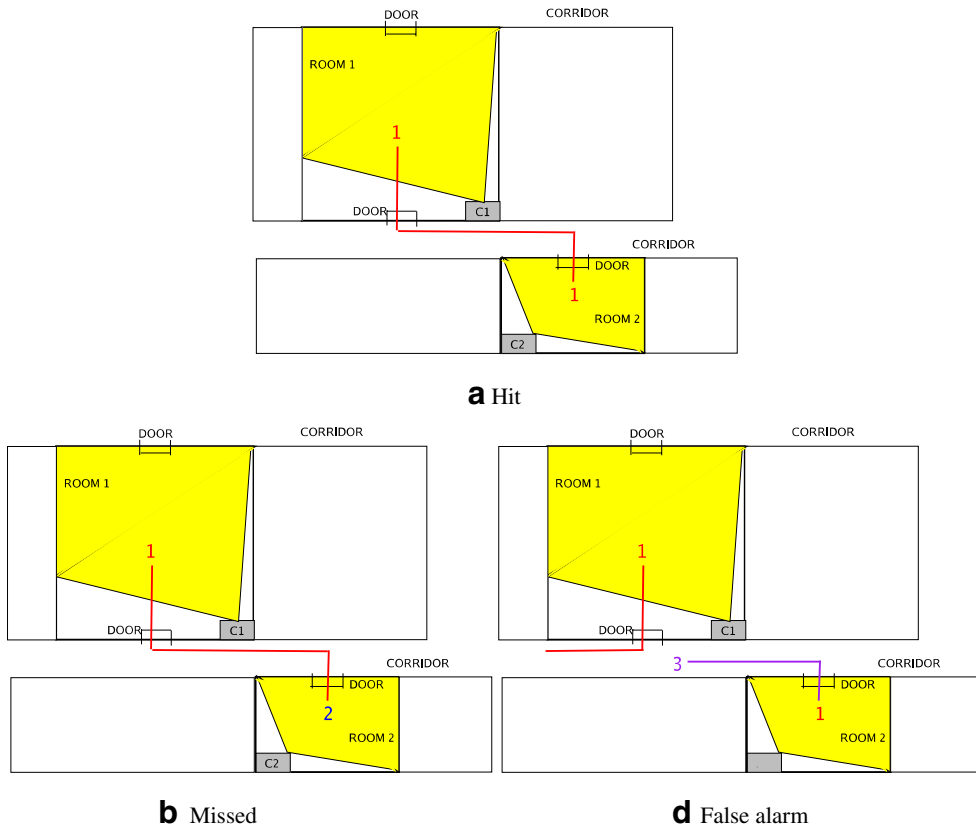


Figure 15. Matching procedure. a, b, and c illustrate the “Hit,” “Missed,” and “False alarm” definition.

Table 2. System performance.

Missed	False alarm	Hit	Total
2	7		
9		44	53

the environment. As the scale of the environment increases, the number of particles required increases. This is because a large number of particles are needed to follow different possible paths of targets' movements.

For the testing purposes, 25 sets of video sequences were collected. Each video consists of 1,500 to 2,200 frames and in total of 53 transactions (1 to 5 transactions for each video sequence). The outcome is the system has successfully tracks and associate targets for 20 sets of video sequences. These video sequences include various kind of walking pattern such as merging and splitting, target turns back, time out, and occlusion. The remaining five video sequences where the system fails for a number of tracking situations are mainly due to the system false positively identifying some similar colour profile targets. Another reason was when targets spend a large amount of time on the non-overlapping area. This leads to the prediction of particle filter becomes less accurate. Table 2 outlines the system performance on non-overlapping region tracking.

## 6.2. Performance Measure for Tracking Within FOV

The performance on tracking within the camera's field-of-view, as for non-overlapping region tracking, improves as the number of particles increases. In the experiment, eight video sequences were tested which contain objects occlusion (merging and splitting). The tracking system achieves 100% accuracy rate for these sequences of video.

We also develop a prior testing on the number of particles required for tracking one target within FOV. The test was done through testing an individual person walking inside a room. Initially, 200 particles were used to track the person, which led to the system failed to track the person. However, the system successfully tracks the person with 500 particles or more. The evaluation on the number of particles used is based on the error produce by a number of experiments, which is shown in Table 3. The average error rate is

computed based on the difference between particles position and observation (ground truth) at each time frame, which can be expressed in Eq. (13).

$$TotalErrorRate = \sum_{k=0}^T \left( \frac{\sum_{i=0}^{N_s} (x_k^i - z_k)}{N_s} \right) \quad (13)$$

$$AverageErrorRate = \frac{TotalErrorRate}{T}$$

where  $k$  is the time unit from 0 to  $T$  (the end of the observations),  $N_s$  is the number of particles. In other words, the average error rate is the average of total error during object tracking. A smaller average error rate means particles prediction is closer to the real observations.

The numbers on Table 3 represent the average error rate in millimeters (after rounding up the value). When we use 500 number of particles, the average error rate reduces to approximately 126 mm, which means that the average error of particle filter prediction is "two walking steps away" from the current observation (as a normal single foot step of a person walking requires 500 mm long). This amount of error rate can be considered as acceptable. Other reason of using this amount of particles is due to memory consideration. The process of the whole system includes reading frames from the video file, extracting each foreground object from background, tracking the object in non-overlapping and within camera FOV through particle filter, and storing the colour profile of each agent in database. Thus, throughout our testing, 500 number of particles are used to track a target within FOV.

## 7. Conclusions

We have proposed a unified technique, based on particle filter that is able to track subjects when they leave the view of a camera and when they re-enter in

Table 3. Error rate.

Number of particles	Average error rate (mm)
200	344
400	212
500	126

another location. As the focus of this project is to investigate the problem of tracking people in a network of distributed camera system, we did not consider other pertinent issues like background adaptation, crowded scene tracking and object interactions. In order to make this applicable to real-world outdoor conditions, an integrated approach which combines these considerations as well as the tracking tasks should be further investigated.

It is well known that object association techniques based on appearance model alone would experience high failure rates which make them impractical for implementation. We have found out that by implementing a form of track prediction based on a modified particle filter together with map information, we are able to keep the failure rate down thereby making it a viable approach. Since the performance of the tracking system is measured by how well the system performs the (exit–entry) association task for a particular subject, one can consider producing a ranked order of possible associations instead of reporting only the first ranked result as in the case here. This feature is practical for a semi-automated system. We have proposed, implemented and tested a unified framework of tracking human subject across a distributed camera system. A particle filter-based approach was used that is able to switch between visual tracking, when the subject is seen by the camera and track prediction, when the subject has left the field of view of the camera and enters into the non-overlapping region of the system. The latter prediction function is accomplished using a map of the surveillance area. The map acts as the main input to the particle filter, which control the movement of the particle filters in the surveillance space. The use of a map provides a scalable solution to deal with an expandable area of interest (e.g. scaling from building level to street level tracking) typical of real-world surveillance applications. Experiments conducted so far have been very encouraging. One of our near-term task is to increase the number of cameras in the network and measure its performance under real-world conditions.

### Acknowledgments

The data for Merging and Splitting Scenario 2 are obtained from the EC funded CAVIAR project/IST 2001 37540, found at URL: <http://www.homepages.inf.ed.ac.uk/rbf/CAVIAR/>.

### References

1. B. Andrew and M. Isard, "Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion," Springer, London, 1998. ISBN: 3-540-76217-5.
2. C. Bererton, "State Estimation for Game AI Using Particle Filters," in *AAAI Workshop on Challenges in Game AI*, Pittsburgh, PA, USA, 2004.
3. M. Bolic, "Architectures for Efficient Implementation of Particle Filters," Ph.D. thesis, Electrical Engineering, Stony Brook University, August 2004. <http://www-sigproc.eng.cam.ac.uk/smc/smcpapers.html>.
4. N. Checka, K.W. Wilson, M.R. Siracusa and T. Darrell, "Multiple Person and Speaker Activity Tracking with a Particle Filter," in *Proc. (ICASSP '04). IEEE International Conference on Acoustics, Speech, and Signal Processing*, Cambridge, MA, vol. 5, May 2004, pp. V881–V884.
5. D. Comaniciu and V. Ramesh, "Mean Shift and Optimal Prediction for Efficient Object Tracking," in *International Conference on Image Processing*, Princeton, NJ, 2000.
6. A.R. Dick and M.J. Brooks, "Issues in Automated Video Surveillance," in *Proc. 7th International Conference on Digital Image Computing: Technique and Applications*, Macquarie University, Sydney, NSW, Australia, 2003, pp. I195–I204.
7. A.R. Dick and M.J. Brooks, "A Stochastic Approach to Tracking Objects Across Multiple Cameras," in *Proc. Australian Joint Conference on Artificial Intelligence*, Mawson Lakes, SA, 2004, pp. 160–170.
8. N.J. Gordon, D.J. Salmond and A.F.M. Smith, "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation," *IEE Proc.-F on Radar and Signal Processing*, vol. 140, no. 2, 1993, pp. 107–113.
9. R.D. Green and L. Guan, "Tracking Human Movement Patterns Using Particle Filtering," in *Proc. International Conference on Multimedia and Expo*, Hongkong, vol. 3, July 2003, pp. 117–120.
10. T. Huang and S.J. Russell, "Object Identification in A Bayesian Context, in *International Joint Conference on Artificial Intelligence*, Nagoya, Japan, 1997, pp. 1276–1283.
11. M. Isard and A. Blake, "Condensation-Conditional Density Propagation for Visual Tracking," *International Journal of Computer Science*, vol. 29, no. 1, 1998, pp. 5–28.
12. M. Isard and J. MacCormick, "Bramble: A Bayesian Multiple-Blob Tracker," in *Proc. Eighth IEEE International Conference on Computer Vision*, Palo Alto, CA, vol. 2, July 2001, pp. 34–41.
13. O. Javed, Z. Rasheed, K. Shafique and M. Shah, "Tracking Across Multiple Cameras with Disjoint Views," in *International Conference on Computer Vision*, Nice, France, 2003, pp. 952–957.
14. R.E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transaction of the ASME-Journal of Basic Engineering*, 1960, pp. 35–45.
15. V. Kettmaker and R. Zabih, "Bayesian Multi-Camera Surveillance," in *Proc. IEEE Computer Vision and Pattern Recognition*, Miami, FL, USA, 1999, pp. 253–259.
16. S. Khan and M. Shah, "Consistent Labeling of Tracked Objects in Multiple Cameras with Overlapping Fields of

- View,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, 2003, pp. 1355–1360.
17. A.J. Lipton, C.H. Heartwell, N. Haering and D. Madden, “Critical Asset Protection, Perimeter Monitoring, and Threat Detection Using Automated Video Surveillance,” in *Proc. 36th Annual 2002 International Carnahan Conference*, 11600 Sunrise Valley Dr, Suite 290 Reston, VA 20191, Oct 2002.
  18. D. Makris, T. Ellis and J. Black, “Bridging the Gaps Between Cameras,” in *Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, vol. 2, 2004, pp. 205–210.
  19. S. Maskell and N. Gordon, “A Tutorial on Particle Filters for On-line Nonlinear/Non-Gaussian Bayesian Tracking,” *IEE Target Tracking: Algorithms and Applications (Ref. No. 2001/1174)*, vol. 2, Oct 2001, pp. 2/1–2/15.
  20. K. Nummiaro, K.M. Esther and L.J.V. Gool, “Object Tracking with an Adaptive Color-based Particle Filter,” in *DAGM-Symposium*, Zürich, Switzerland, 2002, pp. 353–360.
  21. K. Okuma, A. Taleghani and N.D. Freitas, “A Boosted Particle Filter: Multitarget Detection and Tracking,” *European Conference on Computer Vision*, vol. 1, no. 2, Feb 2004, pp. 28–39.
  22. H. Pasula, S.J. Russell, M. Ostland and Y. Ritov, “Tracking Many Objects with Many Sensors,” in *Proc. of the Sixteenth International Joint Conference on Artificial Intelligence*, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., 1999, pp. 1160–1171.
  23. F. Porikli and A. Divakaran, “Multi-Camera Calibration, Object Tracking and Query Generation,” in *Proc. International Conference on Multimedia and Expo*, Baltimore, Maryland, vol. 1, July 2003, pp. 653–656.
  24. A. Rahimi, B. Dunagan and T. Darrell, “Simultaneous Calibration and Tracking with a Network of Non-overlapping Sensors,” in *Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, vol. 1, 2004, pp. 187–194.
  25. A. Rahimi, B. Dunagan and T. Darrell, “Tracking People with a Sparse Network of Bearing Sensors,” in *European Conference on Computer Vision*, Prague, Czech Republic, vol. 4, 2004, pp. 507–518.
  26. C. Stauffer and W.E.L. Grimson, “Learning Patterns of Activity Using Real-Time Tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 22, no. 8, August 2000, pp. 747–757.
  27. M.J. Swain and D.H. Ballard, “Indexing Via Color Histograms,” in *Proc. of the DARPA Image Understanding Workshop International Conference on Computer Vision*, Pittsburgh, PA, USA, 1990, pp. 390–393.
  28. M.C. Tse, R.C. Luo and H.H. Tsu, “Visual Tracking Using Adaptive Color Histogram Model,” in *Industrial Electronics Society, 1999. IECON '99 Proc. The 25th Annual Conference of the IEEE*, San Jose, USA, vol. 3, 1999, pp. 1336–1341.
  29. G. Welch and G. Bishop, “An Introduction to the Kalman Filter,” *Technical Report, University of North Carolina at Chapel Hill*, Chapel Hill, NC, USA, 1995, <http://www.ncstrl.org:8900/ncstrl/servlet/>.



**FEE LEE LIM** is a Research Fellow in the Department of Computing at the Curtin University of Technology, Western Australia. Her research interests are security and surveillance, space variant sensors, illumination consideration for face recognition system, model based object recognition and multi-modal systems.



**WILSON LEOPUTRA** graduated from Curtin University of Technology, Western Australia in 2005 with First Class honours in B Sc (Computer Science) Hons. He is currently a PhD student in the Department of Computing at Curtin University of Technology. His research interests include artificial intelligence, computer vision, pattern recognition and surveillance.



**TELE TAN** is a Senior Lecturer in the Department of Computing at the Curtin University of Technology, Western Australia. His research interests are in human motion analysis, security and surveillance, multi-modal system considerations and technology commercialization. Dr Tan helped contribute to the original commercialization plan of a biometrics start-up company, XiD Technologies (<http://www.xidtech.com>) in late 2002. The biometrics software developed by the company was nominated for the 2004 World Technology Awards (software category) that was held in conjunction with the World Technology Summit 2004. He was made Technical Adviser to Miltrade Technologies in 2003 and was appointed International Reader with the Australian Research Council (ARC) in mid 2004.