People Counting System Based on Particle Filter with Memory States for Improvement

[1]J. M. Wang (王俊明), [2]S. W. Chen (陳世旺), and [1]C. S. Fuh (傅楸善)

[1] Computer Science and Information Engineering, National Taiwan University, Taiwan

[2] Computer Science and Information Engineering, National Taiwan Normal University, Taiwan

Abstract

Tracking people is a big challenge because of the human's unpredictable shape and moving changes. To overcome such problems, Bayesian estimation has been suggested in filtering. Among the implementations of the Bayesian estimation, particle filter has been proposed to solve the nonlinear distribution. In this paper, we give a framework for people detection and tracking using particle filter. Memory states are introduced here as the prior knowledge for the measurement and transition model in particle filter. This consideration helps to track the individual person in crowded case.

1. Introduction

Since getting the people number is very useful, some methods have been proposed: First, infrared sensors are set up in the roadside, and used to detect the people passing while that has been blocked. It is easy to construct, but may be fail to count those people walking side by side. Second, setting a gate in the passageway can help to count pedestrian more accurately. However, the gates may be hard to set up and will influence people's walking. Third, we may dispatch some persons to count people, which is not efficient and not economical.

A vision-based counting system can be applied to count people based on the monitoring system. Such system has more advantages than the other methods addressed before. First, cameras are easy to set up and easy to maintain. Second, it would not influence the people's moving. Third, by applying some appropriate image processing method, occluded pedestrians can be separated and would be counted correctly. At last, monitoring image could provide more information, such as the directions or the human status, than the other systems.

Since a vision-based counting system has so many advantages, many researches have been proposed for counting vehicle [1]. However, developing a people counting system will face more challenges because of our unpredictable behavior. Such unpredictable behavior includes of two phases: In the first one, human shape is not fixed because of their variable postures. Besides, one pedestrian may be connected with others or carries some objects so that it will form many kinds of shape. The second phase is the pedestrian's unstable moving state. The moving state includes the moving trajectory and the moving gait, which may be changed at will or by accident. The above challenges need to be considered while developing such system.

A vision-based counting system often consists of three steps: detecting, tracking, and counting. In detecting step, many detection methods, such as model matching [3], temporal differencing [2], and background subtraction [6], have been developed. After obtaining the foreground, we have to separate that if some human figures are connected with each other. Using a contour (ellipse or rectangular) to represent a pedestrian figure is a common idea [13], but it is too rough to detect while they are occluded. Level set [4] and snake model [5] can be applied to model a human's silhouette dynamically [15], but they are too precise to tolerate the imperfect observation. Detecting each part of a human body and combing them according to a predefined architecture can solve the above problems [18][14], but there are many constrain required to define the human architecture and the combinations may require many computation time to prove.

In the tracking step, we want to obtain the moving trajectory of each pedestrian. This can be done by matching the pedestrian silhouette between sequential frames [16], or changing the model's state to achieve the current state [15]. Because the detecting step may not be reliable, some prediction or filtering models are applied here to compensate that. Kalman filter [10] is a common filtering while the target has a stable moving. Since a pedestrian often moves at will, this method would have some problem in tracking people. Some updating versions of Kalman filter [11] have been proposed to solve the nonlinear system that could be approximated by a Gaussian distribution. Particle filter [7], unlike Kalman filter, is a non-parametric filtering model without predefining the distribution of the prior and posterior knowledge. Recently, some researches have shown that it is robust to track people [13].

After detecting and tracking, the people number can be obtained by counting the trajectory number. This counting can be applied in two kinds of area. In the first one, the area is closed and we want to count the people number in it. Such area often has an entrance that we can monitor that and it often requires to define a cross line for counting [12]. In an open area, since we cannot define inside or outside part, only the passing number can be counted. The passing number means the number of the people that have been detected [10], which will be the same as the trajectory number in the scene.

In this paper, we consider the people moving system is a nonlinear dynamic system, that is, their states are assumed to be generated by the nonlinear functions. Particle filter is applied here to solve this problem. The remainder of this paper is organized as follows: In Section 2, the nonlinear dynamic system and particle filter are introduced. Four issues, initial state, transition, measurement, and determination are introduced in Section 3, 4, 5, and 6. In Section 7, the experiments are performed on some real video sequence. Finally, a conclusion and discussion are addressed in Section 8.

2. Sequential Monte Carlo Method

In our application, we want to know the human behavior (hidden states) according to the monitoring image sequence (observations). Since the actual human behavior can not be known, its distribution of the state given the passing observations, $z_{0:t} = (z_0, \cdots, z_t)$, is defined as $p(s_t|z_{0:t})$, where s_t is the human behavior shooting at time t. In this system, each image frame is computed to the distribution of the human behavior $p(s_t|z_{0:t})$, and the behavior s_t with a highest probability is determined as the pedestrian detecting result. The number of the behaviors along time, $s_{0:t} = (s_0, \cdots, s_t)$, is regarded as the detection result of one pedestrian.

Using Bayesian rule, this prior probability can be updated to

$$p(s_t \mid z_{0:t}) = \frac{p(z_t \mid z_{0:t-1}, s_t) p(s_t \mid z_{0:t-1})}{p(z_t \mid z_{0:t-1})}, \qquad (1)$$

where $p(z_t \mid z_{0:t-1})$ is the predictive distribution of z_t given the past observation $z_{0:t-1}$, and it is a normalizing term in most case. Assume that $p(z_t \mid z_{0:t-1}, s_t)$ depends only on s_t through a predefined measurement model $p(z_t|s_t)$. Equation (1) can be rewritten as

$$p(s_t \mid z_{0:t}) = \alpha p(z_t \mid s_t) p(s_t \mid z_{0:t-1}), \qquad (2)$$

where α is a constant.

Now, suppose s_t is Markovian, then its evolution can be described through a transition model, $p(s_t|s_{t-1})$. Based on this model, $p(s_t|z_{0:t-1})$ can be calculated using the Chapman-Kolmogorov equation:

$$p(s_t \mid z_{0:t-1}) = \int p(s_t \mid s_{t-1}) p(s_{t-1} \mid z_{0:t-1}) ds_{t-1}. \qquad (3)$$

Equations (2) and (3) show that we can obtain $p(s_t|z_{0:t})$ recursively if we have the following requirements: the measurement model $p(z_t|s_t)$, the transition model $p(s_t|s_{t-1})$ and the initial distribution $p(s_0|z_0)$.

The main problem is how to represent the distribution of the transition model $p(s_t|s_{t-1})$ and the unknown state given the observation $p(s_0|z_0)$. This problem also induces the problem of the calculation of the Equation (3). In Hidden Markov Model, these models are discrete, and all calculations can be combined into matrices form. In Kalman filter, Gaussian distribution is assumed, and then the parameters (mean and standard deviation) of $p(s_t|z_{0:t-1})$ can be calculated.

Particle filter is a Monte Carlo method that uses m particles, $s^{(i)}, i = 1...m$, and their corresponding weights, $w^{(i)}$, to simulate the distribution $p(s_t|z_{0:t})$. This simulation also can be applied to Equation (3):

$$p(s_t \mid z_{0:t-1}) = \sum_{i=1}^{m} p(s_t \mid s_{t-1}^{(i)}) p(s_{t-1}^{(i)} \mid y_{0:t-1}). \qquad (4)$$

The distribution $p(s_t \mid z_{0:t-1})$ also can be simulated using the these particles if we have a new transition equation $s_t^{(i)} = f(s_{t-1}^{(i)}, u_{t-1})$ matching the transition model $p(s_t|s_{t-1})$, where u is a noise sequence with zero mean. The distribution of the propagating result

$$f(s_{t-1}^{(i)}, u_{t-1}) p(s_{t-1}^{(i)} \mid y_{t:t-1}), i = 1...m, \qquad (5)$$

will be the same as $p(s_t|z_{0:t-1})$ if the particle number is infinite [7].

After propagating particles, each particle's weight is rescaled with the Equation (3),

$$\begin{aligned} \tilde{w}_t^{(i)} &= w_{t-1}^{(i)} p(z_t \mid x_t^{(i)}) \\ w_t^{(i)} &= \tilde{w}_t^{(i)} \Big/ \sum \tilde{w}_t^{(t)} \end{aligned} . \qquad (6)$$

The distribution of the samples $S_t = \{s_t^{(i)}, w_t^{(i)}\}_{i=1}^{m}$ can show the distribution of $p(s_t|z_{0:t})$. This is often our goal, and one may calculate the expected value as the detection result as the traditional particle filter method [11]:

$$E[S_t] = \sum_{i=1}^{m} s_t^{(i)} w_t^{(i)}. \qquad (7)$$

Expected value, however, is not a good solution, because the solution may be located at the state value with lower probability. The posterior distribution may have multiple maxima, and the best solution would be not the expected value in most case. To estimate the maximum a posteriori (MAP), one may calculate the derivative of $p(s_t|z_{0:t})$ respect to s_t as the research in [8] or approximate the distribution as a mixture of Gaussians [17]. In this paper, we apply mean shift method [19] to locate the maxima, $E'[S_t]$, and this state is defined as our detecting result.

Before the next round, particles are often resampled according to $p(s_t|z_{0:t})$ (i.e. the distributions of $w_t^{(i)}$), and their weights are reset to $1/m$ (that is also the new value of $p(s_t^{(i)} \mid z_{0:t})$). In our practice, those particles with weight less than $1/m$ are eliminated at first, and the remains with weight value greater than c/m will be repeated c-1 particles, where c is an integer. After that, all of the particle weights are set as $1/m$. If the final particle number is not m, some particles will be added or eliminated randomly to correct that.

For each pedestrian moving in the monitoring range, we detect his behavior and denote the detecting result as $s_{0:t}$. State s_t is calculated to the MAP of $p(s_t|z_{0:t})$ using the mean shift method. The distribution of $p(s_t|z_{0:t})$ can be obtained if we have the following requirements: The first requirement, initial state s_0, is defined according to the foreground object extracted using the method proposed in [9]. The second requirement, measurement model, is computed by comparing the current image with our predefined human model. The final requirement, transition model, would be defined by some prior knowledge, and updated according to the detecting result. People number is then calculated to the number of the moving trajectory $s_{0:t}$.

3. Initial Distribution

Initial distribution $p(s_0|z_0)$ is the base of the sequential Bayesian estimation. We can detect the appearing objects and encode them as our initial states, since one pedestrian must move into the monitoring range before the following tracking. Foreground object detection can be applied here to detect the appearing objects. After the foreground object detection, those foreground objects that are not covered by any $E'[S_t]$ are called as the appearing objects. In fact, we do not require the perfect foreground region in our application, so that we can detect the inconsistent region instead of the foreground objects to reduce the processing time.

In this step, we focus on giving a new particle set, S_t, to model the distribution of the appearing object's state. The state of a particle is set as the following features: shape, intensity histogram, and moving velocity. Among these features, the shape should be designed to match a pedestrian's silhouette. However, a pedestrian's silhouette can be modeled complexly by parts [14][8] or simply just by one ellipse [13]. Because of the huge computation in particles, we use one ellipse to model one pedestrian, and its axes (major and minor), rotation angle and centroid are set as the shape features. The second feature, color histogram is constructed using the values of the pixels in the ellipse. The velocity includes the moving direction and moving distance that can be computed while we have two successive states.

In an image, we denote the region of an appearing object as R, and set the features of each particle, centroid, axes, and rotation angle, according to that. Suppose the centroid of R is $(\bar{x},\bar{y})$, we define the centroid (x_c, y_c) of a particle as $x_c = \bar{x} + N(l_R/2)$ and $y_c = \bar{y} + N(l_R/2)$, where l_R is the size of R, and $N(\sigma^2)$ is a Gaussian noise with a standard deviation σ^2. l_R is the width or the height of R depending on which one is longer. All of the particles will be distributed in the range of R by this definition. The axes, a and b, are defined as $a = b = l_R + N(l_R/4)$.

If b is greater than a, we swap them to ensure that the major axis is greater then the minor one. Finally, the rotation angle, θ, is defined as a random number between -π/2 to π/2.

If we set the weight of each particle as $1/m$, these particles will give a normal distribution with the feature values of R as the mean value. Normal distribution cannot represent the actual state of the appearing objects, so we apply Equation (6) to this original distribution to obtain a more suitable distribution. After resampling, the new particle set will model the initial distribution, and the color histogram of the pixels enclosed by one particle are then computed.

4. Transition Model

The transition model $s_t^{(i)} = f(s_{t-1}^{(i)}, u_{t-1})$ defines the transition of one state from time t-1 to time t. It involves two kinds of information, the memory states and the previous transition. We will explain the memory states before the model construction.

4.1 Memory States

In the beginning, the monitoring range is divided into $w \times h$ blocks. For each block (i, j), we set a memory state m_{ij} to record the detected states in this block. These states $\{m_{ij} \mid i = 1..w, j = 1..h\}$ are called as the memory states. To let the recording data being correct, the memory state will be updated until we have a reliable state. A detected state is reliable while it has the following criteria: First, there is only this state detected in the monitoring range. Second, this state has high probability $p(s_t|z_{0:t})$. Third, this state transits smoothly. If the detected state achieves the above criteria, we may suppose that there is only one pedestrian in the scene, and this pedestrian has no occlusion and separating. This leads the memory states to be constructed automatically.

All parameters in a memory state are represented as random variables. As those features in one state, these parameters include of shape and moving parameters. The shape parameters include of **a**, **b**, and **θ**, corresponding to the major axis, minor axis, and rotation angle; the moving parameters are the moving vector $\mathbf{v} = (\dot{x}, \dot{y})$ defined in the image coordinates.

In computing the parameter values of m_{ij}, we can update all values except **v**. Because the moving vector $(\dot{x}, \dot{y})$ that we can obtain at time t would belong to the state at time t -1, that is, $\dot{x}_{t-1} = x_{c,t} - x_{c,t-1}$ and $\dot{y}_{t-1} = y_{c,t} - y_{c,t-1}$. We must update the memory state m_{kl} in the block (k, l) where the position $(x_{c,t-1}, y_{c,t-1})$ locates on. After all, each memory sate will represent the distribution of the detected states in the block along time.

After a period of time, the memory states will record the historical trajectory in this scene. This information can be applied to predict and to evaluate the pedestrian's moving path. In crowded case, one individual person would be matched with the other one in the neighborhood. To make a more precise tracking, we need a better transition rather than the Gaussian transition as those developed in other researches [13]. We will give more details of our transition model later.

4.2 Construction Using Historical Path

A pedestrian may have an expectable moving path in an environment. This expectance is caused by the following two characteristics: First, a pedestrian often have a sudden velocity change because of inertia. Second, most of the pedestrians will have some similar trajectories in the same environment. Those trajectories are called as the historical paths. A historical path can be constructed according to the memory states, if we can start from any memory state and connect to the next one on the expected moving vector $E[\mathbf{v}]$. In our application, we use the probability of the historical transition, **v**, to predict the next moving.

Assume $(\Delta x_{t-1}, \Delta y_{t-1})$ be the moving vector of the state, this vector can be calculated using the following equations:

$$\begin{aligned} \Delta x_{t-1} &= \alpha \times \hat{x} + (1-\alpha) \times \Delta x_{t-2} + N(\max\{a_{t-1}, b_{t-1}, \Delta x_{t-2}\}/2) \\ \Delta y_{t-1} &= \alpha \times \hat{y} + (1-\alpha) \times \Delta y_{t-2} + N(\max\{a_{t-1}, b_{t-1}, \Delta y_{t-2}\}/2)' \end{aligned} \quad (12)$$

where $(\hat{x}, \hat{y})$ is a moving vector selected under **v** of the memory state in the corresponding block, and $\alpha \in [0,1]$ is a weight value for adjusting the trust degree between the memory state and the previous change (α will be 0 if there is no value in the memory state in this block). This definition let the transition not only refer to the previous state but also consider the historical path. This consideration especially can be applied to the crowded case.

The parameter values of the ellipse in the state $s_t^{(i)}$ are then defined using the moving vector $(\Delta x_{t-1}^{(i)}, \Delta y_{t-1}^{(i)})$ of particle *i*:

$$\begin{aligned} &x_{c,t}^{(i)} = x_{c,t-1}^{(i)} + \Delta x_{t-1}^{(i)} \quad , \quad y_{c,t}^{(i)} = x_{c,t-1}^{(i)} + \Delta y_{t-1}^{(i)} \\ &a_t^{(i)} = a_{t-1}^{(i)} + N(a_{t-1}^{(i)}/4)\,, b_t^{(i)} = b_{t-1}^{(i)} + N(b_{t-1}^{(i)}/4)\,. \\ &\theta_t^{(i)} = \theta_{t-1}^{(i)} + N(\pi/4) \end{aligned} \tag{13}$$

In these definitions, all of the particles will simulate the distribution of Equation (3). We will have a good simulation result if the particle number is infinite. However, the more particles we have, the more computations we need. In our practice, the particle number is 200, and the processing result is good enough.

5. Measurement Model

Measurement model, $p(z_t|s_t)$, is calculated to the probability of the observation, z_t, given the unknown state, s_t. It can be considered as the degree of the state matching to the given image. In our assumption, a pedestrian could be represented as an ellipse, so the high degree means the more confidence of the state. The matching degree of state *s* is defined as

$$p(z\,|\,s) = d_h(s)^{g_h} \times d_f(s)^{g_f} \times d_e(s)^{g_e} \times d_m(s)^{g_m}, \tag{14}$$

where the $d_i(s)$ and g_i are the matching degree and the weighting value according to the feature *i*. Each measure, $d_i(s)$, is defined in the following contents.

$d_h(s)$ measure the similarity between the current and the previous intensity histograms of the state *s*, which means that this state represents a same object along time. We construct the histogram by counting the number of pixels falling into an intensity bin. If the bin size is *c* and the range of the intensity value is 0...255, there will be 256/*c* bins in one histogram. The distance between the two histogram h_1 and h_2 can be defined based on the Bhattacharyya coefficient.

$d_f(s)$ is defined as the ratio of the foreground region in the region of *s*, which means that the state *s* cover more foreground region will have larger $d_f(s)$ value. We design this to require the whole particle set cover more foreground regions. This will help the tracking model to pay more attention on the moving objects. Besides, we define the foreground region size be 1 at least, so that the state with small size will be enhanced while there is noting that can be located. This would be happened when the object moving to the outside of the monitoring range.

$d_e(s)$ is designed to let the state *s* locate on the target. In our application, the target is the pedestrian in the scene. We assume that one pedestrian's silhouette can be shaped as an ellipse, and the shaping degree is decided by evaluating the edge point along the boundary of the ellipse. In the image, the boundary points (*x*, *y*) of the ellipse are those point satisfied these equations:

$$\begin{aligned} x &= x_c + a \times \cos(\phi) \times \cos(\theta) - b \times \sin(\phi) \times \sin(\theta) \\ y &= y_c + a \times \cos(\phi) \times \sin(\theta) + b \times \sin(\phi) \times \cos(\theta)' \end{aligned} \tag{16}$$

where $\phi = [0,2\pi]$ is the parameter along the ellipse boundary. We may say that there is an edge point on the boundary while this boundary point and its near point (x_n, y_n) have great different intensity values. We calculate (x_n, y_n) by

$$\begin{aligned} x_n &= x_c + (1 \pm d) \times a \times \cos(\phi) \times \cos(\theta) - (1 \pm d) \times b \times \sin(\phi) \times \sin(\theta) \\ y_n &= y_c + (1 \pm d) \times a \times \cos(\phi) \times \sin(\theta) + (1 \pm d) \times b \times \sin(\phi) \times \cos(\theta) \end{aligned} ,\tag{17}$$

where *d* is the minimum distance from the changing point to the boundary. Figure 5 shows the meaning of *d*. Suppose d_ϕ, $\phi = [0,2\pi]$ means all edge distances measured along the boundary, $d_e(s)$ can be defined as

$$d_e(s) = \min_\phi\{g(d_\phi)\}, \phi = [0,2\pi], \tag{18}$$

where *g*(.) is a decreasing function (ex. Gaussian with zero mean). Under this definition, the ellipse that encloses the human shape will have high shaping degree. Figure 3(b) and (c) show the examples with high and low shaping degree

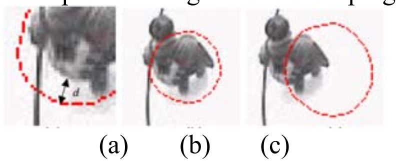

Figure 3. (a) *d* means the distance between the ellipse boundary point to the edge point. (b) shows the example with high shaping degree, and (c) shows the example with low shaping degree.

The final measurement, $d_m(s)$, is calculated based on the memory states using the following equation

$$d_m(s) = \min\{\frac{a_s}{a_m}, \frac{a_m}{a_s}\}, \tag{19}$$

where a_s and a_m are the major axes in the state *s* and the memory state corresponding to the location. Under this definition, those detected object has similar size with the memory state will have high degree.

6. Detection Result Determination

After we have the distribution of the state given observation, $p(s_t|z_{0:t})$, we need to decide a state value to represent the detecting result. As mentioned before, expected value is calculated as the detecting result in particle filter, but it is not the best value in some kind of distribution (ex. multimodal distribution). In this framework, we use mean shift to locate the peak state value. One may refer to [19] to obtain the detail of the mean shift.

In mean shift, the main problems are the start point and the Parzen window size. In our application, they are defined according to the expected value calculated by Equation (7). Given the expected state, we set its centroid as the start point and its major axis length as the Parzen window size. Since the transition model has a noise within the state size, these setting will let the shift considering most of the states. The final location is supposed to be better than the expectation.

Figure 4 shows the detection result defined as the expected state and located using our mean shift method. In this figure, the crosses mark the positions of the particles, and its size means the particle weight. While there are more than two pedestrians close to each other, the distribution of $p(s_t|z_{0:t})$ will have more than one peak values. Expectation value (marked as +) of such distribution cannot mean the good state value, so we use mean shift method to locate that.

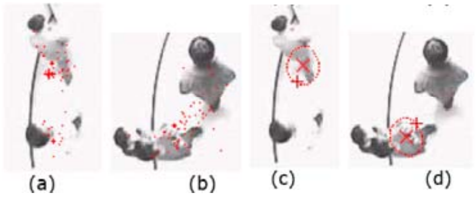

Figure 4. (a) (b) The +'s mark the particles with their weights. (c) (d) The centroids of the expectation state is denoted as "+", and our detection result is denoted as "X".

7. Experiments

We test our method on an indoor monitoring system. In this system, a fisheye camera is mounted on the ceiling over a walk. The video sequence is saved and processed offline. If we detect the inconsistent region instead of the foreground region, our method can process 4 frames in one second using the computer with Pentium 4 3.0GHz CPU. Figure 5 shows the monitoring image. In this figure, detected result is marked using an ellipse with two numbers, where the upper number is a sequential number of this pedestrian, and the lower one is its stay time.

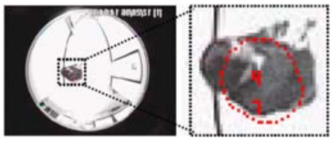

Figure 5. In the monitoring image, the detected result is marked using an ellipse with two numbers.

Our method is designed to handle the variances of the human state. The variances include the moving and shape change. Figure 6 shows the examples. In the upper sequence, a pedestrian moved forward and stopped for a while. We can locate his position well. In the lower sequence, the pedestrian did not stop, but its shape is really different along time. We can detect that without missing.

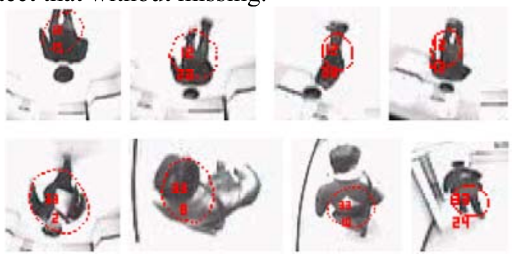

Figure 6. In the upper sequence, the pedestrian move and stop at will. In the lower sequence, the shape of the pedestrian may be changed along time.

In the second case (Figure 7), we show some people passing interlaced. People may pass in the different ways or in the same way. After the processing, they are labeled as the same number after passing. Notice that the detecting result still marked in the rough position even if they are occluded (ex. the number 35 pedestrian in the lower sequence).

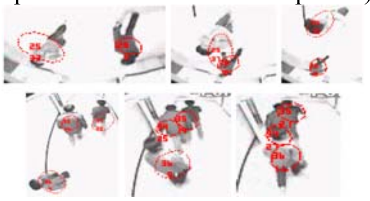

Figure 7. The upper sequence shows two pedestrians passing in the different ways. The lower sequence shows three pedestrians passing in the same way.

The upper sequence of Figure 8 shows the pedestrian moving to behind the door. Unlike those people move to outside the monitoring range, its size is large and it may disappear suddenly. The detection result would locate on the position where this pedestrian disappears for a while. This behavior like our expectation for a disappear object. Similarly, a pedestrian may appear on any position. We can locate such pedestrian without any problem, because our method did not set the region of interest.

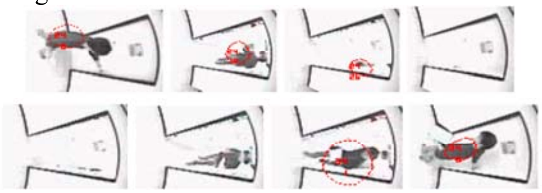

Figure 8. There is a pedestrian moving to behind the door (the upper sequence) and front the door (the lower sequence).

Table 1 shows the counting numbers of two sequences. We count the people number in a long sequence to show the precision. The precision rate is defined as $1-\frac{n_e}{n_a}$, where n_a is the actual number, n_e is the counting error number using our method. Counting error includes miss counting and over counting. Miss counting is often happened while a pedestrian connected with the others in all time; over counting is happened while a non-human object is detected.

Table 1: The counting results

	Sequence 1	Sequence 2
Time length	5 min	10 min
Actual number	52	66
Counting number	48	56
Miss counting	5	12
Over Counting	1	2
Precision rate	88.5%	79.8%

We count the people number in the video for 5 minutes. The actual people number is 52, and the number counted using our method is 48. There are 5 missing and 1 over counting, so the precision rate is 88.5%. Figure 9 shows some examples of the miss counting. More than two pedestrians may be connected with each other by occlusion or shadow in all time, so they are counted as one pedestrian. Over counting is happened while non-human object is detected. This can be solved if we could give a threshold value for $p(E'[S_t]|z_{0:t})$, but it is a dilemma of counting precision or tracking adaptation number. The experiments show that our method can adapt to the human's unpredictable shape and moving changes.

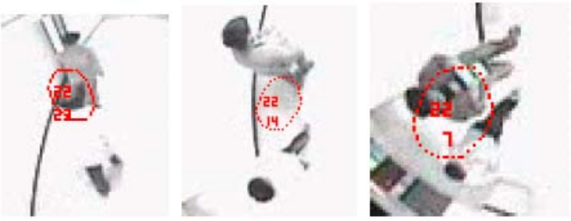

Figure 9. Two pedestrians are connected with each other for some reasons, such as (a) occlusion, (b) shadow, and (c) side by side.

The length of the second video is about ten minutes. In this video, the walk is crowed and most of the people move in the same way. The actual people number is 66 and the counting number is 56. There are many missing, and the precision rate is about 80%. Most error caused by the problem of the occlusion and the error measurement in the crowded people. While the observation given one particle $p(z_t \mid s_t^{(i)})$ is measured, each particle may have high probability to locate some other pedestrian in the scene. Our memory states play an important role right here. To avoid the particles to locate some other pedestrian on the previous location, those particles will be transited to a new location according to the memory states. Figure 10 shows a sequence with crowed people and the detection result.

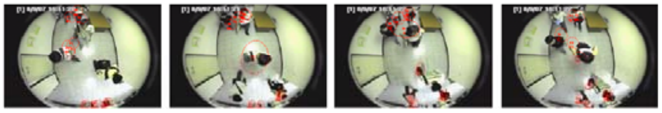

Figure 10. Crowded people will cause more locating errors.

8. Conclusion

Counting people cannot avoid detecting and tracking people. In this paper, we propose a people counting system based on the particle filter applied to track people. When the particle filter is applied, the initial state, measurement model and transition model need to be defined before hand. We give a processing method to define the initial state automatically and to construct the prior knowledge for measurement and transition model. Among those requirements, we define the initial state according to the foreground object detection. The measurement model combines the variance information of a human, such as the color and shape, which helps to locate the pedestrian more precisely. Since people may have the same trajectory in an environment, we define the memory states to express the historical moving path. The memory states will give the prior knowledge of the human's transition model.

The final calculation result is the probability of the human state; mean shift is applied here to locate the peak value instead of the expectation value used before. The detection results along time will show the human's trajectories, and we can obtain the people number by counting the trajectory

References

1. J. M. Wang, Y. C. Chung, S. C. Lin, S. L. Chang, S. Cherng, and S. W. Chen, "Vision-Based Traffic Measurement System," *Int'l Conf. on Pattern Recognition*, vol. 4, pp. 360-363, Cambridge, 2004.
2. S. Jehan-Besson, M. Barlaud, and G. Aubert, "A 3-Step Algorithm Using Region-Based Active Contours for Video Objects Detection," *Journal on Applied Signal Processing*, vol. 2002, no. 1, pp. 572-581, 2002.
3. A. Broggi, M. Bertozzi, A. Fascioli, M. Sechi, "Shape-based pedestrian detection," *Proc. of the IEEE Intelligent Vehicles Symposium*, pp. 215-220, Dearborn, 2000.
4. J. A. Sethian, *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science (2nd ed.).* Cambridge University Press. ISBN 0-521-64557-3.4., 1999.
5. F. Buccolieri, C. Distante, and A. Leone, "Human Posture Recognition Using Active Contours and Radial Basis Function Neural Network," *IEEE Conf. on Advanced Vide and Signal Based Surveillance*, pp. 213-218, 2005.
6. J. W. Kim, K. S. Choi, B. D. Choi, and S. J. Ko, "Real-time Vision-based People Counting System for the Security Door," *Int'l Technical Conf. on Circuits/Systems Computers and Communications*, pp. 1416-1419, 2002.
7. N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation", *IEE Proc.-F Radar and Signal Processing*, vol. 140, no. 2, pp. 107-113, 1993.
8. E. Poon and D. J. Fleet, "Hybrid Monte Carlo Filtering: Edge-Based People Tracking", *Proc. Workshop on Motion and Video Computing*, pp. 151-158, Orlando, 2002
9. J. M. Wang, S. Cherng, C. S. Fuh, and S. W. Chen, "Foreground Object Detection Using Two Successive Images", *IEEE International Conf. on Advanced Video and Signal-based Surveillance*, pp. 301-306, Santa Fe, 2008.
10. O. Masoud and N. P. Papanikolopoulos, "A Novel Method for Tracking and Counting Pedestrians in Real-Time Using a Single Camera," *IEEE Trans. on Vehicular Technology*, vol. 50, no. 5, pp. 1267-1278, 2001.
11. O. Cappe, S. J. Godsill, and E. Moulines, "An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo," *Proceedings of the IEEE*, vol. 95, no. 5, 2007
12. A. Albiol, I. Mora ,and V. Naranjo, "Real-Time High density people counter using morphological tools," *IEEE Trans. on Intelligent Transportation Systems*, vol. 2 , no. 4 , pp.204 – 218, 2001.
13. E. Maggio, F. Smerladi, and A. Cavallaro, "Adaptive Multifeature Tracking in a Particle Filtering Framework*," IEE Trans. On Circuits and Systems for Video Technology*, vol. 17, no. 10, pp. 1348-1359, 2007
14. D. Ramanan, D. A. Forsyth, A. Zisserman, "Tracking People by Learning Their Appearance," *IEEE. Trans. On Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 65-81, 2007
15. Y. Alper, L. Xin, and S. Mubarak, "Contour-Based Object Tracking with Occlusion Handling in Video Acquired Using Mobile Cameras*," IEEE trans. On Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1521-1536, 2004.
16. J. M. Wang, S. W. Chen, S. Cherng, and C. S. Fuh, "People Counting Using Fisheye Camera," *Proc. of the IPPR Conf. on CVGIP*, Mauli, Taiwan, 2007.
17. M. W. Lee and I. Cohen, "A Model-Based Approach for Estimating Human 3D Poses in Static Images", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 6, pp. 905-916, 2006
18. T. Zhao, R. Nevatia, and B. Wu, "Segmentation and Tracking of Multiple Humans in Crowded Environments", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1198-1211, 2008.
19. Y. Cheng, "Mean Shift, Mode Seeking, and Clustering", IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 17, no. 8, pp. 790-799, 1995.