# Particle Filter Tracking Method Using Graphical Model

Ming-Jie Zhang[1, 2] Bao-Sheng Kang[1]

[1] School of Information Science and Technology, Northwest University
Xi'an, 710127, China
75711650@qq.com
bskang@163.com

[2] School of Economics and Management, Xi'an University of Post and Telecommunications
Xi'an, 710121, China
75711650@qq.com

*Received 18 October 2014; Revised 27 January 2015; Accepted 9 March 2015*

**Abstract.** To solve tracking problem when objects' part are temporally occluded in complicated scene, a novel particle filter tracking method using graphical model is proposed. The method applies particle filter to graphical models. When the color histogram of the tracked object is made, there is not one region of whole object but the multi-part region of it, thus it is divided into some parts, which means that the multi-part region is taken as a graphical model. In the process of object tracking by using particle filter method, messages about the state of the parts are sent to other parts in the graphical model. As the result, when the tracked object has occluded part, the proposed method can track stably and infer the state of the occluded part. Finally, experimental results verify the proposed method effectively.

**Keywords:** particle filter, real-time tracking, graphical model, message propagation, observation model

## 1 Introduction

Visual object tracking has become a hot issue in computer vision, particularly in applications such as human computer interaction, automotive preventive safety, traffic monitoring, automated surveillance [1-4]. However, object tracking is extremely complex and time consuming especially in outdoor environments. Some problems of object tracking can be mentioned in outdoor environments such as occlusion, shadows, illumination changes and presence of clutter. To overcome these difficulties, a variety of tracking algorithms have been proposed. These methods for object tracking can be subdivided into two main groups [5]: deterministic methods and probabilistic methods. Deterministic methods typically track the object by performing an iterative search for a similarity between the template image and the current one. On the other hand, the probabilistic methods use the state space to model the underlying dynamics of the tracking system such as Kalman filter and particle filter. Particle filter methods are the most prominent.

In recent years, some scholars combine graph theory and probability theory, probabilistic graphical models (PGMs)[6] have become a popular and powerful mathematical tool to learn, represent, and compute complex probability distributions by explicitly defining statistical dependencies between the elements of the probability model. PGMs are able to provide an appropriate theoretical framework where object dynamics and appearance can be combined and the motion estimation problem can be efficiently solved. Graphical models can be divided into two general classes: directed acyclic graphs (DAGs) and undirected graphs (UGs). In both cases, the basic idea is to provide a graphical tool to decompose a multivariate probability distribution into a factored form by providing an intuitive and manageable visual description. The stochastic variables are associated to nodes and conditional dependencies between the variables are represented by links or edges. Nowadays, because particle filter can solve the estimation problem when the system is nonlinear and non-Gaussian, some scholars apply particle filters to graphical models [7, 8]. They combined belief propagation with particle filtering. E. B. Sudderth et al. [8] employed the Nonparametric Belief Propagation algorithm which can understand graphical models, and apply it to infer location and reconstruct occluded features of faces. However, they need high computational cost and can not apply to real-time tracking.

The object tracking problem under occlusion is always a difficulty in computer vision research. In recent years, Research on this problem is mainly focuses on two aspects below: (1) One is to consider a single object and to use different features to describe the object. (2) The other is to consider a single feature and to use it to describe different sections of the object within a single framework. For study of the first solution, A number of studies, e.g. [9, 10], employed a dynamic framework called the Democratic Integration to adaptively integrate different cues, addressing this problem. In their framework, each cue has an adaptive reliability value associated with it, and each cue contributes to the joint result according to its reliability. H. Wang et al. [11] proposed a color space Gaussian mixture model. By modeling in the color distribution and spatial information, the algorithm can more accurately track object. As for study of the second strategy, S. M. S. Nejhum et al. [12] split an object into parts introduces a kind of supplementary shape information regarding the object by providing the relative spatial arrangements of different sections. This offers an important advantage over the classical region-based trackers where the content of the region of interest is modeled with a single histogram with the loss of spatial information. This kind of part-based trackers mostly aims at tracking articulated/non-rigid objects (e.g. [13]), and generally requires the model of the object to be known or given a priori. Although researchers have done a lot of work in this area, however, the occlusion problem has not been solved well. In the aforementioned method, objects are lost when the occlusion is serious.

In this paper, to solve tracking problem of temporal occlusion in complicated scene and reduce calculation cost, a novel particle filter tracking method using graphical model is proposed. Using simple graphs, the method apply graphical models to compensate for the lack of the information of the occluded part, and propagate messages from a node to other nodes of graph. On the one hand, in spite of simple graphs, the method can avoid losing information and obtain the accurate trajectory of the object motion. On the other hand, the state of occluded parts can be estimated from visible parts.

The rest of the paper is organized as follows. In section 2, particle filter method is briefly introduced. Following that, particle filter method based on graphical model is proposed in section 3. In section 4, the proposed algorithm is described. In addition, experimental results are presented in section 5. Finally, the conclusions are drawn in section 6.

## 2 Particle Filter Method

Visual tracking can be considered as an optimal estimation problem, the tracking purpose is that inference is performed on dynamic Bayesian networks by estimating the belief state using the Bayes' rule. That is to say, the optimal estimation of object state is obtained by calculating the posterior probability density $p(x_k \mid z_{1:k})$. $p(x_k \mid z_{1:k})$ is computed recursively using a two steps method, the two steps are as follows:

(1) Prediction step: $x_k$ is the state vector, the observations is denoted as $z_k$, $z_{1:k} = \{z_{1,}z_2 \ldots, z_k\}$. Assuming that the posterior $p(x_{k-1} \mid z_{1:k-1})$ is available, it is possible to make predictions using the state space model $p(x_k \mid x_{k-1})$.

$$p(x_k \mid z_{1:k-1}) = \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid z_{1:k-1}) dx_{k-1} \quad \textbf{(1)}$$

(2) Update step: Updating priori probability density using the observation $z_k$ and the likelihood probability $p(z_k \mid x_k)$ that is derived from the observation. The posterior probability density $p(x_k \mid z_{1:k})$ can be expressed by Eq. (2).

$$p(x_k \mid z_{1:k}) = \frac{p(z_k \mid x_k) p(x_k \mid z_{1:k-1})}{p(z_k \mid z_{1:k-1})} \quad \textbf{(2)}$$

The particle filter (PF) provides a framework for the recursive state estimation in nonlinear, non-Gaussian problems. In particle filter, the posterior probability density can be estimated by a set of random particles $S^{(i)}$, $S^{(i)} = \{x^{(i)}, \pi^{(i)}\}, i = 1, \cdots, N$, $x^{(i)}$ denotes the hypothetical state of the tracked object and $\pi^{(i)}$ denotes its weight. Here, the state is treated as the position of the object. The particle filter method can be described as follows:

Step1. Initialization: $k = 0$.

(1) Sampe N particles, $x_0^{(i)} \sim p(x_0), i = 1, 2 \cdots, N$.

(2) Compute the particle weights using Eq. (3).

$$\pi_0^{(i)} = p(z_0 \mid x_0^{(i)}) \quad \textbf{(3)}$$

(3) Normalize the importance weight using Eq. (4).

$$\tilde{\pi}_0^{(i)} = \frac{\pi_0^{(i)}}{\sum_{i=1}^{N} \pi_0^{(i)}} \tag{4}$$

Step2. Prediction and update:
(1) Sampe N particles, $x_k^{(i)} \sim q(x_k \mid x_{0:k-1}^{(i)}, z_{0:k}), i = 1, 2 \cdots, N$.
(2) Compute the particle weights using Eq.(5).

$$\pi_k^{(i)} = \pi_{k-1}^{(i)} \frac{p(z_k \left| x_k^{(i)}) p(x_k^{(i)} \right| x_{k-1}^{(i)})}{q(x_k^{(i)} \left| x_{0:k-1}^{(i)}, z_{0:k} \right.)}, i = 1, 2, ...N \tag{5}$$

(3) Normalize the importance weight using Eq. (6).

$$\tilde{\pi}_k^{(i)} = \frac{\pi_k^{(i)}}{\sum_{i=1}^{N} \pi_k^{(i)}} \tag{6}$$

Step3.Selective re-sampling:
1). Compute the effective particle size $N_{eff}$.

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} (\pi_k^{(i)})^2} \tag{7}$$

2). If $N_{eff} \le N_{th}$ multiple weighted particles to generate N equal weighted particles.
Step4. Output: State estimation.

$$\tilde{x}_k = \sum_{i=1}^{N} \pi_k^{(i)} x_k^{(i)} \tag{8}$$

## 3 Particle Filter Method Based on Graphical Model

In this paper, we compensate for the lack of the information (e.g. the occluded part) using graphical models. In the conventional method, the region of the color histogram is often obtained from the whole tracked object. In our proposed method, the regions are obtained from the parts of divided object. The parts are set to nodes of the graph and propagate messages from a node to other nodes of graphs. By this means, when the part of object is occluded, the method can avoid losing information and obtain the trajectory of object motion stably. In this section, particle filter method based on graphical model is explained.

### 3.1 Observation Model

Color features have mainly RGB and HSV, RGB is an uneven color space, moreover HSV is an even color space, it can better response person visual sense about color. So we select HSV color space. Target color distribution use color histograms of positional information. Assuming the color information in the HSV space is divided into the levels, which is $m = 16 \times 4 \times 4$.

Calculating the likelihood, Bhattacharyya distance is used to measure similarity between two color distributions. Obtain the object color distribution $p$ of the object template and the color distribution $q$ of the candidate object.

Using the Bhattacharyya distance, the similarity is measured between object template color distribution which are denoted by $p$ and the candidate object color distribution which are denoted by $q$. The Bhattacharyya distance is given as:

$$d = \sqrt{1-\rho[p,q]} \tag{9}$$

$$\rho[p,q] = \sum_{u=1}^{m} \sqrt{p^{(u)} q^{(u)}} \tag{10}$$

Where, $\rho$ is Bhanacharyya coefficient. The color of the object observation likelihood function is defined as:

$$p(z_k \mid x_k) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(1-\rho[p(x^{(i)}),q)]}{2\sigma^2}) \tag{11}$$

$\sigma^2$ is Gaussian variance, the value is 0.2 in our experiment.

### 3.2 State Space Model

State space model describes the dynamic process of object state with time change, in the particle filter algorithm, the object state is represented with particles. Therefore, the state space model describes the spread process of the particle. In this paper, a rectangle represents the object, the state vector is defined as $X_k = (x_k, y_k, h_k, w_k)$, $x_k, y_k$ are the center coordinate of rectangle, $h_k, w_k$ are the height and width of the rectangle respectively. Second order auto regressive model is used to describe the object state:

$$X_k - X_{k-1} = X_{k-1} - X_{k-2} + U_{k-1} \tag{12}$$

$U_k$ is random noise, it obeys the normal random distribution, and it is denoted by $U_k \sim N(0,\sigma)$.

In Eq.(12), state space model is as follows:

$$X_k = 2X_{k-1} - X_{k-2} + U_{k-1} \tag{13}$$

This state space model is used in the sampling step of particle filters.

### 3.3 Graphical Model

A graph G is defined by a set of nodes V, and a corresponding set of edges $\varepsilon$ .The neighborhood of a node $\alpha$ ( $\alpha \in V$ ) is defined as $\Gamma(\alpha) = \{\beta \mid (\alpha,\beta) \in \varepsilon\}$. $x_{\alpha,k}$ and $z_{\alpha,k}$ are respectively the state and the observation of the node $\alpha$ at time $k$ . $m_{\alpha\beta}$ is the message from the node $\alpha$ to the node $\beta$, This message works to transfer the information of nodes to neighbor nodes.

To simplify, we consider models with potential functions for each node, it is as follow:

$$p(x_{\alpha,k,} z_{\alpha,k}) = \frac{1}{Z}\{\phi_\alpha(x_{\alpha,k,} z_{\alpha,k}) + \sum_{\beta\in\Gamma(\alpha)} \phi_{\alpha,\beta}(x_{\alpha,k,} x_{\beta,k})\} \tag{14}$$

Where, Z is the normalization factor. We calculate the posterior density $p(x_{\alpha,k} \mid z_k)$ based on these potential functions, and approximate the posterior density using messages. The state $x_{\alpha,k}$ is estimated by these approximated densities.

### 3.4 Update and Prediction of Particles

In the paper, the particles are used in two stages. One is the update, and another is the prediction. Updated and predicted particles of the node $\alpha$ at time *K* are defined as:

$$\bar{x}_k = E[x_k \mid z_k, ..., z_1] \tag{15}$$

$$\hat{x}_k = E[x_k \mid z_{k-1}, ..., z_1] \tag{16}$$

These particles are used properly in the iterative computation of the tracking algorithm. In the improved particle filter, each particle has state and weights. The *i*-th updated and predicted particles at time $k$ are represented by $\bar{s}_k^{(i)} = \{\bar{x}_k^{(i)}, \bar{\pi}_k^{(i)}\}_{i=1,\ldots,N}$ and $\hat{s}_k^{(i)} = \{\hat{x}_k^{(i)}, \hat{\pi}_k^{(i)}\}_{i=1,\ldots,N}$ .

### 3.5 Posterior Density

In general the particle filter, the posterior density is approximated according to assumption that the temporal transition of the state is simple Markov process. The relation term between neighbor nodes is added to the posterior density of the state of the node $\alpha$ at time $k$ :

$$p'(x_{\alpha,k} \mid z_{\alpha,k}, z_{\Gamma(\alpha),k}) = \frac{1}{Z}\{p(x_{\alpha,k} \mid z_{\alpha,k}) + \sum_{\beta \in \Gamma(\alpha)} \phi_{\alpha,\beta}(x_{\alpha,k}, x_{\beta,k}) p(x_{\beta,k} \mid z_{\beta,k})\} \tag{17}$$

Where, $\phi_{\alpha,\beta}$ is the related potential function between the node $\alpha$ and the node $\beta$ . The second term of above equation is approximated by message particles. The details about message particles are shown in the following subsection.

From the posterior density $p'$ , the expected value of a function $f(x_{\alpha,k})$ is given as:

$$E[f(x_{\alpha,k})] = \int f(x_{\alpha,k}) p'(x_{\alpha,k} \mid z_{\alpha,k}, z_{\Gamma(\alpha),k}) dx_{\alpha,k} \tag{18}$$

### 3.6 Message Propagation

The particles of general particle filter propagate through the Markov chain temporally. In addition, the message particles propagate through the edges of the graph. In Fig. 1, the graphical model has two nodes, it is connected by the Markov chain. The message flows are shown as arrows in Fig. 2.

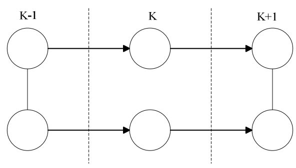



**Fig. 1.** Graph models which have two nodes connected by Marcov chains

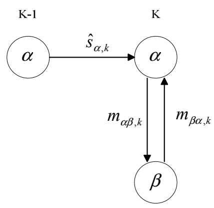



**Fig. 2.** Propagation of predicted particles and message particles

In Eq. (17), The message particles play the role of the approximation of the second term. Here we reinterpret the second term of it as:

$$\phi_{\alpha,\beta}(x_{\alpha,k}, x_{\beta,k}) p(x_{\beta,k} \mid z_{\beta,k}) = \frac{1}{Z} p(z_{\alpha,k} \mid x_{\beta\alpha,k}) p(x_{\beta\alpha,k} \mid z_{\beta,k}) \tag{19}$$

Where, $Z$ is the normalization factor. In Eq. (19), we approximate it by using the message particle-set $m_{\beta\alpha,k}^{(i)} = \{x_{\beta\alpha,k}^{(i)}, \pi_{\beta\alpha,k}^{(i)}\}$ from the node $\beta$ to node $\alpha$ . The weight of the message particle is defined as:

$$\pi_{\beta\alpha,k}^{(i)} = p(z_{\alpha,k} \mid x_{\alpha,k} = x_{\beta\alpha,k}^{(i)}) \tag{20}$$

In fact, we make the message particles by propagating between the nodes as below:

$$x_{\beta\alpha,k}^{(i)} = A_{\beta\alpha,k} x_{\beta,k}^{(i)} + b_{\beta\alpha,k} \quad \textbf{(21)}$$

$A_{\beta\alpha,k}$ and $b_{\beta\alpha,k}$ are respectively the propagation matrix and vector from the node $\beta$ to the node $\alpha$ at time k. The state of the node $\alpha$ is estimated according to the distribution of updated particles:

$$E[x_{\alpha,k}] = \sum_{i=1}^{N} \overline{\pi}_{\alpha,k}^{i} \hat{x}_{\alpha,k}^{(i)} + \sum_{\beta\in\Gamma(\alpha)} \sum_{i=1}^{N} \overline{\pi}_{\beta\alpha,k}^{i} |\hat{x}_{\beta\alpha,k}^{(i)} \quad \textbf{(22)}$$

$$\sum_{i=1}^{N} \overline{\pi}_{\alpha,k}^{i} + \sum_{\beta\in\Gamma(\alpha)} \sum_{i=1}^{N} \overline{\pi}_{\beta\alpha,k}^{i} = 1 \quad \textbf{(23)}$$

## 4. Algorithm Design

In this section, our proposed tracking algorithm is explained as follows:

Step 1: Prepare the graph $G$ of the tracked object and the node set $\alpha$ of the graph and initialize the updated particle-set $\overline{s}_{\alpha,0}^{(i)} = \{\overline{x}_{\alpha,0}^{(i)}, \overline{\pi}_{\alpha,0}^{(i)}\}_{i=1,\dots,N}$

Step 2: Predict the particle-set at the next time-step using Eq. (13). Make the predicted particle-set $\hat{s}_{\alpha,k}^{(i)}$, using selected previous updated particle-set $\overline{s}_{\alpha,k\text{-}1}^{(i)}$

Step 3: Color observation likelihood function value is calculated using Eq. (11) .Calculate the weight of each particle as below:

$$\hat{\pi}_{\alpha,k}^{(i)} = \frac{1}{\sqrt{2\pi\sigma}} \exp(-\frac{(1-\rho[p(x_{\alpha,k}^{(i)}, q_{\alpha})]}{2\sigma^2})] \quad \textbf{(24)}$$

Step 4: Make message particle sets. Calculate the state and the weight of message $m_{\beta_l\alpha,k}^{(i)} (\beta_l \in \Gamma(\alpha), l = 1, \cdots, n)$ from neighbor nodes $m_{\beta\alpha,k}$ as below:

$$x_{\beta_l\alpha,k}^{(i)} = A_{\beta_l\alpha,k} \hat{x}_{\beta_l,k}^{(i)} + b_{\beta_l\alpha,k} \quad \textbf{(25)}$$

$$\pi_{\beta_l\alpha,k}^{(i)} = \hat{\pi}_{\beta_l,k}^{(i)} \quad \textbf{(26)}$$

Step 5: Select $N$ samples from the predicted particle-set $\hat{s}_{\alpha,k}^{(i)}$ and message particle-set $m_{\beta_l\alpha,k}^{(i)}$ .The normalized cumulative probability $c_{\alpha,k}^{\prime(i)}$ is calculated as below:

$$c_{\alpha,k}^{(0)} = 0 \quad \textbf{(27)}$$

$$c_k^{((n+1)i-n)} = c_k^{((n+1)i-n-1)} + \widetilde{\pi}_{\alpha,k}^{(i)} \quad \textbf{(28)}$$

$$c_k^{((n+1)i-n+l)} = c_k^{((n+1)i-n+l-1)} + \overline{\pi}_{\beta_l\alpha,k}^{(i)} \quad \textbf{(29)}$$

$$c_k^{\prime(j)} = \frac{c_k^{(j)}}{c_k^{(N)}} \quad \textbf{(30)}$$

Generate a uniform distribution random number $r \in [0,1]$. Find the smallest $j$ satisfied as $c_k^{\prime(j)} \geq r$ .Set $\overline{s}_{\alpha,k}^{(i)} = \hat{s}_{\alpha,k}^{(j)}$ or $m_{\beta_l\alpha,k}^{(j)}$.

Step 6: Estimate the position $x_{\alpha,k}$ of each node of the object using Eq.(22).

Step 7: Set $k = k + 1$, and go to step 2.

## 5 Experimental Results

In this section, the experimental results are presented by two groups of experiments to illustrate the performance and the effectiveness of the proposed tracking algorithm. The first set of experiments demonstrates the effectiveness of the proposed algorithm on illustrative tracking sequences, and the second set of experiments compares the runtime and tracking accuracy of the two algorithms in different number of particles. In these experiments, the trackers in comparison are initialized by manually marking the image region surrounding the object in the first frame of the sequence.

The proposed algorithm and the conventional particle filter algorithm [14] have been implemented in MATLAB 2009 on a PC with a 2.6 GHz Intel Core2 Duo processor.

### 5.1 Effectiveness Analysis

The effectiveness of our proposed method experiments have been confirmed on three video sequences. The first video sequence contains 384×288 pixel image frames, and the second video sequence contains 720×576 pixel image frames. All the two sequences come from the CAVIAR database [15]. The third video sequence which consists of 352×288 pixel images is provided in [16].

1) The first video sequence is an occlusion clip. In this sequence, two persons walk pass each other and the person being tracked is partially occluded. Comparing the conventional method and the proposed method, we evaluate the tracking algorithm on the video sequences. First, we select an initial object area where 30 particles are set for conventional method in the first frame. In our method, two initial areas are selected to build the graphical model which has two nodes. Each node and each message have 15 particles respectively.

Fig. 3 shows the result of the conventional particle filter algorithm, and the tracking results of our proposed method are shown in Fig. 4. The results show that the tracker drifted away using the conventional method when the object has the partial occlusion, at the same time the position of the occluded part is estimated stably in the proposed method, and the object can be tracked correctly before and after occlusion.

**Fig. 3.** Object tracking experiment using conventional method for sequence 1

**Fig. 4.** Object tracking experiment using our proposed method for sequence 1

Fig. 5 shows the position error of our tracker and the conventional particle filter color-based tracker. In this experiment, the object has severe occlusion between the frames 30 and 70. The estimation of the conventional method fluctuates more widely than the proposed method. In summary, our method performs better than the conventional method.

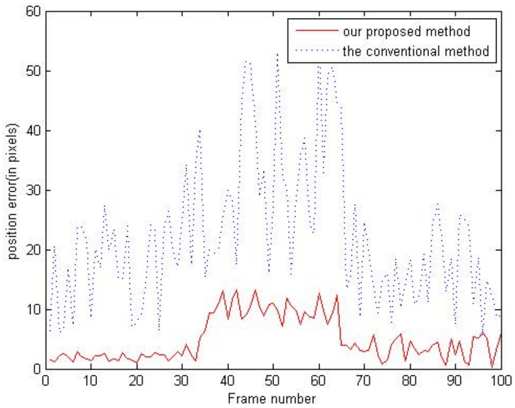



**Fig. 5.** Position error plots for the sequences 1

2) The second video sequence shows object scale changes and partial occlusion. In this experiment, an object is tracked by the conventional method and our proposed method. For initialization, we select an initial object area where 30 particles are set for the conventional method. In the new method, two initial areas are set to build the graphical model that has 2 nodes. Each node and each message have 15 particles respectively.

Fig. 6 and Fig. 7 give the different result of the two algorithms. In Fig. 6, the position of the estimation is affected widely by another pedestrian when the object has the occluded part, and the tracker lose the object afterward. Fig. 7 shows the result of the proposed method. In this case, the position of the pedestrian is estimated stably when partial occlusion occurs. Fig. 8 shows the position error in this experiment, the partial occlusion occurs between the frames 60 and 80. The results confirm that our proposed method demonstrates the better performance than the conventional method. Our method provides the better satisfactory tracking results.

**Fig. 6.** Object tracking experiment using conventional method for sequence 2

**Fig. 7.** Object tracking experiment using our proposed method for sequence 2

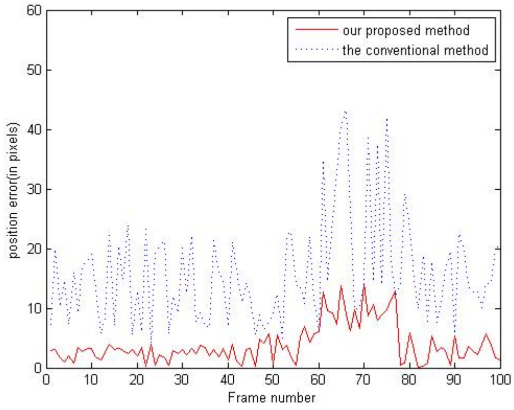



**Fig. 8.** Position error plots for the sequence 2

3) The third video sequence is a tracking of woman's face, and it contains 300 frames. The main difficulty in this sequence is that the face is often occluded by a magazine throughout the sequence. In this experiment, we select an initial object area where 30 particles are set for the conventional method. Three initial areas are selected for the proposed method to build the graphical model which has three nodes. Each node and each message have 10 particles respectively.

Fig. 9 and Fig. 10 show the results of the two different particle filter method. The results show that our method coped with these kinds of occlusions better than the conventional method when the partial occlusion occurs. Our method is able to accurately track the object throughout the whole sequences.

**Fig. 9.** Object tracking experiment using conventional method for sequence 3

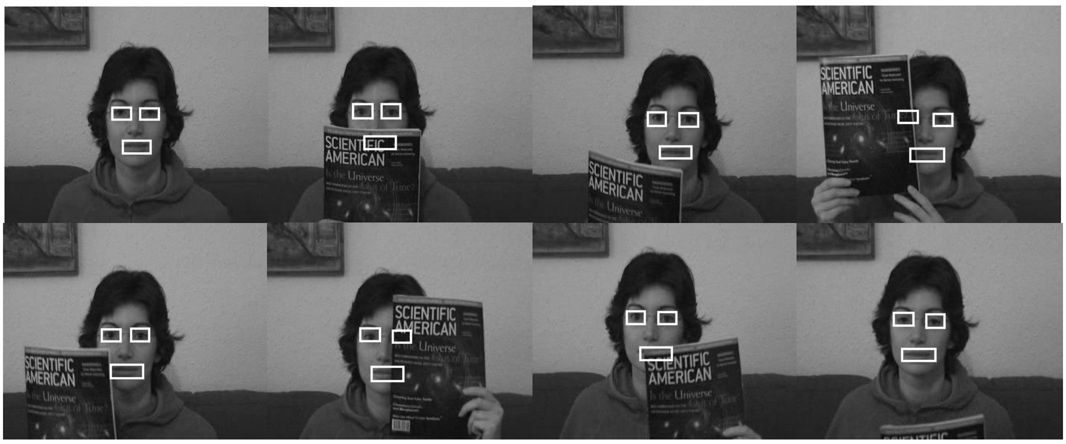


**Fig. 10.** Object tracking experiment using our proposed method for sequence 3

### 5.2 Runtime and Tracking Accuracy Analysis

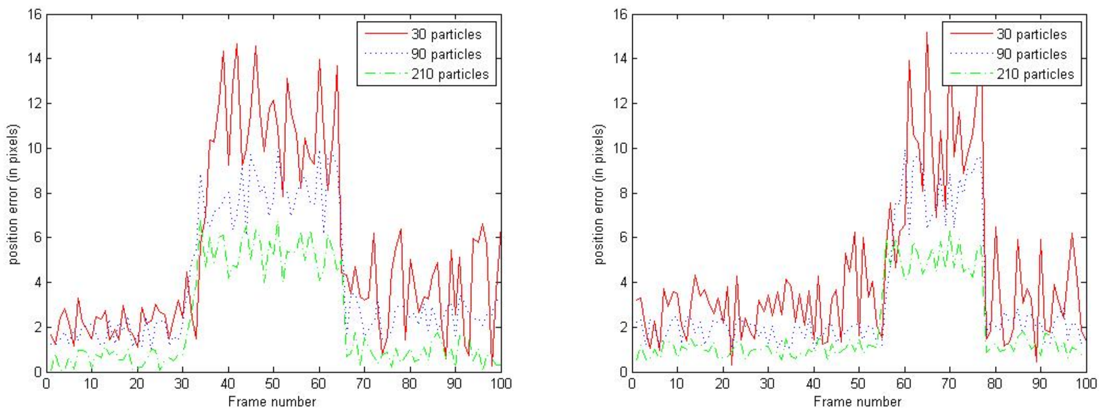
This experiment compares the runtime and tracking accuracy of the two algorithms in different particle numbers. It can be seen from table 1 that the proposed algorithm outperforms the conventional particle filter method in terms of runtime with the same particle numbers. With the increase of the particle numbers, the runtime of the conventional particle filter method is obviously increased.

In Fig. 11, the position error of our tracker is plotted in different particle numbers on two sequences. The results show that the increase of the particle numbers can improve tracking accuracy of the algorithm, but the runtime increases.

From these experiments, our proposed method is effective. When the object has the occluded part, the object can be reliably tracking.

**Table 1.** Comparisons of runtimes from three video sequences (unit: second)

| Sequence | Frame number | Particle number | Runtime | |
|---|---|---|---|---|
| | | | Proposed method | Conventional particle filter method based on color |
| 1 | 100 | 30 | 11.16 | 13.12 |
| | | 90 | 12.51 | 15.56 |
| | | 210 | 14.59 | 19.69 |
| 2 | 100 | 30 | 11.65 | 13.31 |
| | | 90 | 12.96 | 16.12 |
| | | 210 | 15.12 | 20.06 |
| 3 | 300 | 30 | 30.12 | 35.23 |
| | | 90 | 31.67 | 38.35 |
| | | 210 | 34.55 | 44.67 |




(a) Sequence 1 (b) Sequence 2

**Fig. 11.** Position error plots using our proposed method in different particle numbers

## 6 Conclusion

To track moving object reliably and obtain accurate object trajectory, the particle filter method and graph model method are studied, and particle filter is applied to graphical model. To realize reliable tracking, especially when the object has the occluded part, we use graphical models to compensate for the lack of the information of the occluded part. Experimental results have demonstrated that the proposed tracking algorithm is effective. This method not only can be reliably tracking moving object, but also the position of the occluded part can be estimated from visible parts.

## Acknowledgment

This work was supported by the National Nature Science Foundation of China (61272286) and the Ministry of Education in China Project of Humanities and Social Sciences (13YJCZH251).

## References

[1] D.-C. Huang, K.-D. Hung, Y.-K. Chan, "White blood cell nucleus segmentation based on adaptive threshold detector," *Journal of Computers*, Vol. 23, No. 4, pp. 20-29, 2012.

[2] R. Hoseinnezhad, B.-N. Vo, B.-T. Vo, "Visual tracking in background subtracted image sequences via multi-Bernoulli filtering," *IEEE Transactions Signal Processing*, Vol. 61, No. 2, pp. 392-397, 2013.

[3] M. Cristani, R. Raghavendra, A. Del Bue, "Human behavior analysis in video surveillance: A Social signal processing perspective," *Neurocomputing*, Vol. 100, No. 1, pp. 86-97, 2013.

[4] S. Reuter, B. Wilking, J. Wiest , M. Munz, "Real-time multi-object tracking using random finite sets," *IEEE Transactions on Aerospace and Electronic System*, Vol. 49, No. 4, pp. 2666-2679, 2013.

[5] A.Yilmaz, O. Javed, M.Shah, "Object tracking: A survey, " *ACM Computing Surveys*, Vol. 38, No. 4, pp. 1-46, 2006.

[6] A. Dore, M. Soto, C. Regazzoni, "Bayesian tracking for video analytics," *IEEE Signal Processing Magazine*, Vol. 27, No. 5, pp. 46-55, 2010.

[7] M. Isard, "PAMPAS: Real-valued graphical models for computer vision," in *Proceedings of 2003 IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2003)*, Vol. 1, pp. 613-620, 2003.

[8] E. B. Suddenth, A. T. Ihler. W. T. Freeman, A. S. Willsky, "Nonparametric belief propagation," in *Proceedings of 2003 IEEE International Conference on Computer Vision and Pattern Recognition(CVPR 2003)*, Vol. 1, pp.605-612, 2003.

[9] K. Nickel, R. Stiefelhagen, "Dynamic integration of generalized cues for person tracking," in *Proceedings of the 10th European Conference on Computer Vision (ECCV 2008)*, pp. 514-526, 2008.

[10] P. Brasnett, L. Mihaylova, D. Bull, N. Canagarajah, "Sequential Monte Carlo tracking by fusing multiple cues in video sequences, " *Image Vision Computer*, Vol. 25, No. 8, pp. 1217-1227, 2007.

[11] H. Wang, D. Suter, K. Schindler, C. Shen, "Adaptive object tracking based on an effective appearance filter," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No.9, pp.1661-1667, 2007.

[12] S. M. S. Nejhum, J. Ho, M.-H. Yang, "Visual tracking with histograms and articulating blocks," in *Proceedings of 2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, pp. 1-8, 2008.

[13] P. Chockalingam, N. Pradeep, S. Birchfield, "Adaptive fragments-based tracking of non-rigid objects using level sets," in *Proceedings of 2009 IEEE International Conference on Computer Vision (ICCV 2009)*, pp. 1530-1537, 2009.

[14] K. Nummiaro, E. Koller-Meier, L. J. V. Gool, "An adaptive color-based particle filter," *Image and Vision Computing*, Vol. 21, No.1, pp. 99-110, 2003 .

[15] Caviar datasets: http://groups.inf.ed.ac.uk/vision/caviar/caviardata1.

[16] Video datasets: http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml.