Multiple Likelihood Models based Particle Filter for Long-term Full Occlusion

Chengjiao GUO, Ying LU, Takeshi IKENAGA (Member)
Graduate School of Information, Production and Systems, Waseda University

Summary
Object tracking is one of the most important applications in the field of computer vision. One of the common problems in object tracking is object occlusions. Especially in the presence of long-term full occlusion, or called long-lived full occlusion, during which the target remains invisible for tens of frames, the tracking is more difficult. This paper proposes an occlusion handling scheme based on particle filter. Compared with the conventional particle filter which usually utilizes color as tracking cue, multiple likelihood models: HSV color and gradient orientation likelihoods, are employed in the observation model during occlusion. The incorporation of these two features makes the target distinguishable even if it is occluded by a similar colored object in the background. Also, multiple state noises are introduced to ensure the redetection of the target at the end of full occlusion as well as keeping tracking accuracy under occlusion. Experimental results under different occlusion conditions show that the proposed particle filter achieves robust and accurate performance compared with the particle filter with appearance adaptive models and the color particle filter, even in the condition of long-lived full occlusion.

Keywords: Particle filter, object tracking, occlusion handling, multiple likelihood models, multiple state noises

1. Introduction

Moving object tracking has received much interest in the field of computer vision due to the increasing need for automated video analysis. It has an extensive range of applications in the aspects such as traffic monitoring, automated surveillance, video indexing, and human-computer interaction, etc. The problem of tracking can be defined as the issue of estimating the trajectory of an object over the video sequences.

Particle filter, also known as sequential Monte Carlo filters, the CONDENSATION algorithm, is one of the most popular visual tracking methods. It is a technique of nonlinear Bayesian filtering by Monte Carlo simulations, and the discrete approximation of the required posterior density function could be obtained with sufficient samples. In particle filter, the probability density of moving object is estimated by a set of weighted particles. Unlike the traditional Kalman filtering methods, particle filter can deal with non-linear and/or non-Gaussian noise problems with superior accuracy.

An inevitable problem in object tracking is the occlusion of the target object. When occlusion occurs in the video sequences, parts or the entire tracked target are invisible, due to the existence of static objects in the background, such as trees, walls, etc, or due to other moving objects, for example, two moving people cross each other. The occlusion problem makes the tracking complex since the target changes appearance. A number of approaches try to deal with occlusion with particle filter, such as in 3), 4) and in 5) an occlusion detector is developed from the observa-
tion of the weighting operation result by means of a dispersion criterion. Also, the masking technique is added for improving the tracking. Zhou et al.\(^6\) declare the occlusion when the number of outlier pixels in the image patch of interest exceeds a threshold compared with the appearance model. Also, they use robust statistics to reduce the large image differences on the estimation or measurement process caused by occlusion. When the occlusion is detected, the appearance model updating is stopped. Nummiaro et al.\(^7\) employ the similar occlusion handling principle, in which the target model is updated slowly and excludes the frames where the object is occluded or too noisy.

In the previous work\(^3\)–\(^7\), the partial and short-lived occlusion problem is focused on, in which the partial occlusion lasts for only several frames. However, as to the long-lived full occlusion in which the object state remains the same for a long period, the tracking is much more complex, since particles may propagate during the occluding period and converge to a local maximum. Consequently, it loses the target after occlusion. Also, when the target is occluded by an object with similar color in the background, the common color-based particle filter tends to fail since the color is not distinctive enough. In this paper, we try to solve these two kinds of occlusion problems.

An occlusion detector based on particle filter framework is proposed in this paper. The previous work on occlusion detecting is usually based on intensity or color information of pixels\(^5\)–\(^7\). Our proposal differs from those methods in that it not only utilizes color information but also the gradient orientation cue in the occlusion detection. And in order to improve the tracking performance in the occlusion, multiple likelihood models are employed in the observation model. Modification on state transition model is also introduced. Instead of single state noise, we apply multiple state noises in the transition model.

The remainder of the paper is organized as follows. Section 2 introduces the general algorithm of particle filter. The occlusion detecting method is described in Section 3. Section 4 presents the multiple likelihood models. And multiple state noises are proposed in Section 5. The experimental results of the proposed particle filter are demonstrated in Section 6. Conclusion is given in Section 7.

2. Particle Filter Overview

Particle filter is sequential Monte Carlo method that can be used for object tracking within a recursive Bayesian framework. It was originally developed to track objects in clutter. The key idea of particle filters is to give an approximation of the required state probability distribution by a set of discrete random samples, which are named as particles, with associated weights. Each sample represents a hypothesis of the state. And the weight defines the importance of each sample for finding the position of target. The approximation is more accurate when the number of samples becomes larger. However, the computational complexity is increased.

Particle filter provides a robust and efficient tracking framework, since multiple state hypotheses are considered simultaneously. One of the biggest advantages of particle filter is that not all parts of the state space are approximated with the same “resolution”. More samples can be concentrated in the more probable areas of the state space, and less computational time and samples are wasted to approximate the unimportant parts\(^8\).

There are three key steps in the framework of particle filter algorithm. These three steps are conducted recursively in tracking until there are no new observations. The first step is the prediction step. The states of particles are propagated according to the state transition model. Each sample of the state represents an image region and is labelled by a rectangle window in our proposal. We assign a state vector for each particle which is given like this:

\[
X_k = [x_k, y_k, \Delta x_k, \Delta y_k, S_k, \Delta S_k]^T
\]

where \((x_k, y_k)\) denotes the centroid of the particle region, and \((\Delta x_k, \Delta y_k)\) is the speed in \(x\) and \(y\) directions. \(S_k\) is the scaling factor which indicates the scaling changing from the initial frame. \(\Delta S_k\) is the scaling changing in two consecutive frames.

The state transition model predicts the motion between two consecutive frames. The transition model has two components - the deterministic component and the stochastic component. It is used to simulate the effect that a movement has in the set of particles with the noise. It can be expressed in the following
The Journal of the IIEJ vol. 39 no. 5(2010)

form:

\[ X_{k+1} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times X_k + \nu_k \]  

(2)

where \( \nu_k \) is the noise of the transition model which has a Gaussian distribution with zero mean.

The second step is the measurement (update) step. At this step, the particles are re-weighted using the information obtained from the measurements in order to describe the probability density of the moving object accurately. The particle weight, which signifies the quality of that specific particle, is usually calculated by a likelihood function. And a large weight means large probability for the presence of the target in the particle location.

The last step is the resampling step which replicates the samples in proportion to their weights. Thus, the particles with small weight are eliminated since they have little contribution to the approximation of the state probability function. The effective sample size \( N_{eff} \) is defined as

\[ N_{eff} = \frac{1}{\sum_{i=0}^{N-1} (L_k^i)^2} \]  

(3)

where \( L_k^i \) is the normalized weight. When \( N_{eff} \) falls below some threshold, significant degeneracy is observed which implies that a large number of particles have negligible weights.

3. Occlusion Detection

The standard particle filter can perform well with some simple sequences without occlusion or in some short-lived occlusions, since it considers multiple state hypotheses simultaneously. However, when the sequences are complex, for example, in the condition of long-lived full occlusion or when the target is occluded with a similar colored object, it is hard to achieve stable and accurate performance with the standard particle filter. Therefore, an occlusion handling scheme is proposed.

The principle of occlusion detecting in our proposal is evaluating the orientation similarity together with the color similarity of the best color-matched particle in each frame to the reference target model. The occlusion detecting condition is shown like this:

\[ \sqrt{L_{o,k}^0 \times L_{c,k}^0} < \pi_T \]  

(4)

where \( L_{o,k}^0 \) is the gradient orientation likelihood of the best matched particle which has the largest color likelihood \( L_{c,k}^0 \) among all the candidate samples, and \( \pi_T \) is the detection threshold. And we use half of the mean of the largest weights of previous frames to set the threshold.

By incorporating the gradient orientation information with the color, the mismatching caused by occlusion is able to be detected. When the object is fully occluded by other objects with similar color, the dissimilarity in gradient orientation can be detected; also, when the object is completely occluded by other objects with similar orientation feature, the color information can be used to declare the dissimilarity as well as the occlusion.

4. Multiple Likelihood Models

When occlusion is declared, two different likelihood models are utilized in the measurement step: the HSV color likelihood and the gradient orientation likelihood. In object tracking, color-based particle filter is widely used. Color can provide an efficient visual feature for tracking non-rigid objects in real-time. However, when the target is occluded by some objects with the similar color, the sole color likelihood is not sufficient to distinguish the target from the background. Consequently, it is not able to detect the occlusion and tracking failures might be induced. So other information for identifying the target is necessary. In our proposal, gradient orientation is utilized beyond the color feature.

The combination of color and edge orientation features is also used for tracking in the work of Yang et al. The color information is represented by a mean color vector of each sub-region inside the target, and the edge orientation histogram is computed by using horizontal and vertical derivatives to index the edge instead of computing the orientations explicitly. The two features are cascaded for several stages of likelihood evaluation, while our proposal evaluates them concurrently. In our proposal color and gradient histograms are computed in simpler and more
ensures that $\sum (x, y)$ where $I$ is defined as conducted with the Bhattacharyya coefficient, which and the target model. A similarity measure is usually Gaussian weighted color histograms of the particles is 110 in our tracking system for the color histograms.

To increase the reliability of the color distribution, we use the Gaussian weighting function to weight the pixels inside each particle region

\[
g(x, y) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left\{ -\frac{1}{2\sigma^2} \left[ \frac{(x-c)^2}{w^2/2} + \frac{(y-r)^2}{h^2/2} \right] \right\}
\]

where $(x, y)$ denotes the location of one pixel in a particle region. $(c, r)$ is the region center, and $w, h$ are the width and height of the region. $(x-c)/(w/2), (y-r)/(h/2)$ are the normalized distance to the center in x-axis and y-axis. With the Gaussian weighting function, the pixels closer to the center contribute more to the color distribution. When they belong to the background or get occluded, the boundary pixels will have less effect on the color distribution.

The color histogram is calculated as follows

\[
p_c^{(u)} = k \sum_{i=1}^{m} g(x_i, y_i) \delta[h(x_i, y_i) - u]
\]

where $h(x_i, y_i)$ is the quantized index of the HSV color histogram. $g(x_i, y_i)$ is the Gaussian weighting function. $I$ is the number of pixels in the particle region. $\delta$ is the Kronecker delta function and the normalization factor

\[
k = \frac{1}{\sum_{i=1}^{m} g(x_i, y_i)}
\]

ensures that $\sum_{u=1}^{m} p_c^{(u)} = 1$. The number of bins $m$ is 110 in our tracking system for the color histograms.

The measurement step is carried out based on the Gaussian weighted color histograms of the particles and the target model. A similarity measure is usually conducted with the Bhattacharyya coefficient, which is defined as

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

\[
p = \{p^{(u)}\}_{u=1,\ldots,m}, q = \{q^{(u)}\}_{u=1,\ldots,m}
\]

considering discrete densities such as two color distribution histograms $p$ and $q$.

Suppose that $p_c^{(u)}$ is the color histogram of the $i$th particle at the $k_{th}$ frame and $q_c$ is the color histogram of the target model. Then the color likelihood of each particle is obtained based on the Bhattacharyya coefficient:

\[
L_{c,k}^i = \frac{1}{\sigma \sqrt{2\pi}} \exp\left\{ -\frac{1 - \rho[p_c^{(u)}, q_c]}{2\sigma^2} \right\}
\]

### 4.1 HSV Color Likelihood

The color likelihood is calculated in the HSV color space in our proposal. And color histogram is used to represent the color distributions of the target and the samples. To increase the reliability of the color distribution, we use the Gaussian weighting function to weight the pixels inside each particle region

\[
G(x, y) = \frac{1}{\sqrt{2\pi \sigma^2}} \exp\left\{ -\frac{1}{2\sigma^2} \left[ \frac{(x-c)^2}{w^2} + \frac{(y-r)^2}{h^2} \right] \right\}
\]

where $(x, y)$ are the width and height of the region. $\delta(x, y)=(x-c)/(w/2), (y-r)/(h/2)$ are the normalized distance to the center in x-axis and y-axis. With the Gaussian weighting function, the pixels closer to the center contribute more to the color distribution. When they belong to the background or get occluded, the boundary pixels will have less effect on the color distribution.

The color histogram is calculated as follows

\[
p_c^{(u)} = k \sum_{i=1}^{m} g(x_i, y_i) \delta[h(x_i, y_i) - u]
\]

where $h(x_i, y_i)$ is the quantized index of the HSV color histogram. $g(x_i, y_i)$ is the Gaussian weighting function. $I$ is the number of pixels in the particle region. $\delta$ is the Kronecker delta function and the normalization factor

\[
k = \frac{1}{\sum_{i=1}^{m} g(x_i, y_i)}
\]

ensures that $\sum_{u=1}^{m} p_c^{(u)} = 1$. The number of bins $m$ is 110 in our tracking system for the color histograms.

The measurement step is carried out based on the Gaussian weighted color histograms of the particles and the target model. A similarity measure is usually conducted with the Bhattacharyya coefficient, which is defined as

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

\[
p = \{p^{(u)}\}_{u=1,\ldots,m}, q = \{q^{(u)}\}_{u=1,\ldots,m}
\]

considering discrete densities such as two color distribution histograms $p$ and $q$.

Suppose that $p_c^{(u)}$ is the color histogram of the $i$th particle at the $k_{th}$ frame and $q_c$ is the color histogram of the target model. Then the color likelihood of each particle is obtained based on the Bhattacharyya coefficient:

\[
L_{c,k}^i = \frac{1}{\sigma \sqrt{2\pi}} \exp\left\{ -\frac{1 - \rho[p_c^{(u)}, q_c]}{2\sigma^2} \right\}
\]

### 4.2 Gradient Orientation Likelihood

In the gradient orientation likelihood, we use Sobel gradient operator to extract the gradient orientation. Sobel is a simple and efficient edge detecting operator. Two $3 \times 3$ kernels are convolved with the image patch of each sample to calculate the approximate derivatives of each particle region. The convolution process is expressed with the following equations:

\[
G_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \times A
\]

\[
G_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \times A
\]

where $\times$ denotes the two-dimensional convolution, and $A$ is the gray-scale image of the image patch of one specific particle. $G_x$ and $G_y$ are the horizontal and vertical derivatives. So the gradient orientation for the pixel at the position $(u, v)$ can be calculated as

\[
\theta(u, v) = \arctan \frac{G_y(u, v)}{G_x(u, v)}
\]

The gradient orientation ranges from $-\pi/2$ to $\pi/2$. It is discretized and assigned into the 96 bins, so the orientation histogram $p_o = \{p_o^{(v)}\}_{v=1,2,\ldots,n}$ can be produced with

\[
p_o^{(v)} = k \sum_{i=1}^{n} g(x_i, y_i) |G(x_i, y_i)| \delta[\hat{\theta}(x_i, y_i) - v]
\]

where $\hat{\theta}(x_i, y_i)$ is the quantized orientation. $g(x_i, y_i)$ is the Gaussian weighting function (5), and $|G(x_i, y_i)|$ is the gradient magnitude obtained by this equation

\[
|G(x_i, y_i)| = \sqrt{G_x^2(x_i, y_i) + G_y^2(x_i, y_i)}
\]

Then, the similarity test of orientation histogram is conducted based on the Bhattacharyya coefficient. And the orientation likelihood is calculated as

\[
L_{o,k}^i = \frac{1}{\sigma \sqrt{2\pi}} \exp\left\{ -\frac{1 - \rho[p_o^{(u)}, q_o]}{2\sigma^2} \right\}
\]
Therefore, during occlusion, the particles are weighted by the squared likelihood

\[ L_k^i = \sqrt{L_{c,k}^i \times L_{o,k}^i} \]  

(15)

On the other hand, when the target is not occluded, the particles are weighted by the color likelihood.

5. Multiple State Noises

In the general configuration of particle filter, the identical state model with a state noise is utilized in the prediction stage of all the particles. A good state transition model has small prediction noise, which implies that samples are concentrated in a small range and make the model precise. On the other hand, a large prediction noise means a poor prediction but can cover irregular and abrupt fast object motion.

In the presence of complete occlusion, particles will converge to the outliers where the region is not the target. And under a long-lived occlusion, the particles might converge to outliers which are not close enough to the reoccurring position of the target. In the condition, the particles with a small noise tend to find a local maximum and fail tracking.

In our proposal, when the occlusion is declared, two state noises are applied in the prediction stage. Half of the particles are assigned with a small prediction noise and the others with a large noise, which has a larger set of values than that of the small noise. The large noise is used to cover a large range of samples and ensure the probable target region is detected by the particles after full occlusion; while the small one provides an accurate locating of the target in the partial occlusion.

Fig. 1 illustrates how the idea of multiple state noises work under long-lived full occlusion. The black stuffed circles denote particles at previous time \( k-1 \), and gray ones at time \( k \). Green crosses represent the true position of target. And the dashed circles give the probable particle region. The occluded target positions are denoted by blue curves. Suppose that the target is fully occluded for a while before time \( k \). Thus, the particles are focused on background at time \( k-1 \). If the target occurs at \( S_k \) as shown in the figure, the particles are unable to locate it when a small noise is applied, since the target is out of the region of small noise. On the contrary, a large noise denoted with red circle will enable successful detection of the target. In our experiments, the small noise is Gaussian distributed with mean value \( \mu = 0 \), and variance \( \sigma = 1 \). The range of the large noise distributions is set to four times of the range of small noise.

As long as the occlusion is detected, this occlusion handling scheme works. Therefore, in principle, there is no restriction on the time that the occlusion lasts for. On the other hand, in the frames without occlusion, the state transition with one state noise is utilized, which will assign a small prediction region for particle propagation. In general, the object motion between two frames is not remarkable. As a result, accurate tracking result can be achieved.

6. Experiment Result

The proposed particle filter is implemented in C environment. It was tested with some representative real video sequences. All the sequences shown here have the frame rate of 30fps. Each has a different environment, with the target being completely occluded by different objects. Here, occlusion lasting for more than 30 frames is considered as long-lived full occlusion in order to distinguish from the common short-lived full occlusion which lasts for about 10 frames.

To evaluate the performance of the proposed approach, we compared the performance with the color particle filter\(^9\) and particle filter with appearance-adaptive models (AAMs)\(^6\). In 6), occlusion is handled using robust statistics as well as robust likelihood measurement and the adaptive velocity estimate. Accurate and robust tracking results can be achieved as it also adopts online updated appearance model and adaptive state transition model with adaptive velocity and adaptive noise as well as adaptive particle number. Here the number of particles in
AAMs based particle filter is adaptive between 100 and 200. And for the color based particle filter and our proposal, the number of particles are 100 if not specified.

The first sequence is an outdoor video sequence in which a girl is tracked. This video sequence is composed of 468 frames of $320 \times 240$ pixels. The girl is fully occluded by a pole near Frame 180 and Frame 400 when she walks back and forth. The short-lived full occlusions last for about 10 frames. Fig. 2 shows an example of tracking failure with the color particle filter. As shown in Frame 170, it has good performance when the occlusion is not severe. However, in the presence of full occlusion, the performance is quite unstable. For example, particle filter succeeds in Frame 189 but fails in Frame 406. In some other trials, it even fails in Frame 189.

Fig. 3 shows the tracking performance with our algorithm and particle filter with AAMs for the same sequence. The AAMs based particle filter employs image intensities for tracking and it fails tracking in Frame 170 and 338 due to the rich texture in the background. And after the second occlusion, it still focuses on the background. In contrast, thanks to the multiple likelihood functions and multiple state noises, our proposal manages to track the girl even after the full occlusion, although the number of particles is set to be half of the number in Fig. 2.

Fig. 4 shows the results of the proposal, the color particle filter and the AAMs based particle filter for sequence II. This sequence is complex for tracking since the girl is occluded by a boy wearing the similar colored coat, and the occlusion lasts as long as over 40 frames. This sequence is composed of 269 images of $320 \times 240$ pixels. From the above two rows, we can note that the color particle filter converges to the local maximum and do not return on the target at the end of full occlusion, since it utilize color distribution to identify the target. The AAMs based particle filter can track the target at first. However, as shown in Frame 184 and 225, the tracker also fails finding the target since the adaptive state transition model loses effectiveness during the long-lived full occlusion. On the other hand, in our proposal, multiple likelihood
Fig. 4 Comparison between (a) the color particle filter (PF), (b) the AAMs based particle filter and (c) the proposed particle filter with sequence II.

Fig. 5 Comparison with the Ground truth for sequence II.

models help to distinguish the target from the boy, so the occlusion is detected. And thanks to the multiple state noises used in our proposal manages to track the girl soon when she reoccurs after the long-lived occlusion.

Fig. 5 shows the locations of the target every 10 frames during the tracking. Compared with the Ground truth which is defined manually, the pro-
posed particle filter locates the target more accurately than the other two algorithms. The red points between Frame 140 and 170 indicate the full occlusion period from Frame 135 to Frame 172. During the full occlusion, the Ground truth is determined by inference according to the target locations before occlusion. Although the inference will induce error in Ground truth, it can be neglected since the purpose is to track the target when it appears in the visual field. The Y-axis error of the proposal between Frame 140 and Frame 180 is due to the full occlusion in which the target is invisible, and the particles focus on the background.

The locating error estimate that accounts for both the x and y errors is plotted in Fig. 6. The locating error is defined with the equation:

\[
LE = \sqrt{\left(\frac{x - x_g}{W_g}\right)^2 + \left(\frac{y - y_g}{H_g}\right)^2} \tag{16}
\]

where \((x_g, y_g)\) denotes the Ground truth, and \((W_g, H_g)\) is the target width and height respectively. From the curves, it can be noted that our proposed method tracks the target successfully and achieves better accuracy. The full occlusion frames are denoted with red points. Moreover, our proposal has larger locating errors from Frame 140 to Frame 170 than the frames without occlusion, since the target is completely invisible in this period and the particles focus on the background. And in the frames without occlusions, the locating error is kept below 0.5. Therefore, the superior performance of the proposal is evident.

In order to handle the occlusion, effective occlusion detection is needed. Fig. 7 shows the occlusion detection curves for Sequence II according to (4). In sequence II, the occlusion lasts from Frame 110 to 190, during which full occlusion lasts from Frame 135 to 172. The dotted line displays the values of detection threshold, and the solid one plots the occlusion detection score according to the left-hand of (4). The full occlusion frames are indicated with the red line. From the occlusion detection curve, we can see that the occlusion between Frame 127 and Frame 174 can be detected during which the target is severely or completely occluded.

As mentioned in Section 5, when increasing the state noise, particle filter with sufficient samples is likely to cover the probable object region. However, the accuracy of motion model will be lost, and tracking accuracy will be depressed. In contrast, the proposal of multiple state noises keeps the tracking accuracy as well as handles the long-lived full occlusion problem. To demonstrate the accuracy of our proposal, the tracking errors of the color particle filter
Tracking results of (a) the color PF, (b) the AAMs based particle filter and (c) the proposed PF (number of particles is 50) with sequence III with a large state noise and the proposal are displayed in Fig. 8 for every twentieth frame. To consider the locating accuracy and the scaling accuracy as well, the tracking error is defined as

\[ TE = \sqrt{LE \times \frac{|s - s_g|}{s_g}} \]  

(17)

where \( s_g \) is Ground Truth of the scaling value compared to the target size in the first frame. \( s \) is the target scaling obtained from tracking results in each frame, and \( LE \) is the locating error defined in (16). Since the ratio of particle width and height is kept the same in the tracking, the scaling ratio equals the ratio of the height or width. As shown in Fig. 8, the tracking error of the proposal is not more than 0.2 except in Frame 400 where the full occlusion occurs in sequence I. In contrast, the color PF with an increased state noise achieves much worse tracking accuracy than our proposal for most of the cases. And the tracking with large noise tends to fail tracking when the number of particles is small, for example, at 50. Therefore, our proposal has superior performance compared to the conditions of large state noise or conventional small state noise.

Tracking results for another video sequence are shown in Fig. 9. In this sequence, the girl walks to the back of the boy and stays for a while. The full occlusion lasts from Frame 65 to Frame 130. Both the color particle filter and the AAMs based particle filter fail tracking after the full occlusion. It shows that the AAMs based particle filter is not capable of dealing with this full occlusion although it employs robust appearance model and state transition model. However, our algorithm tracks the target successfully with 50 particles. As a result, it can be concluded that our proposal achieves efficient and robust performance.

7. Conclusions

This paper presented an occlusion handling approach with particle filter suitable for object tracking. The proposal is robust and efficient as it can handle long-lived full occlusions. Also, when the target is occluded with the similar colored object, the proposed particle filter can track successfully. The proposed particle filter is tested in different occlusion conditions. And the results are compared with the classical color particle filter and the AAMs based particle filter. The experiment results demonstrate that our proposal achieves superior tracking performance. However, the algorithm might fail if the target appearance changes largely after the full occlusion.
sion, the proposed particle filter might fail since the proposal employs a fixed target model. Further improvement can be achieved by using an adaptive appearance model to solve this problem.

References


(Received Mar. 31, 2010)
(Revised July 31, 2010)