Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Monocular Vision Tracking Based on Hybrid Particle Filters for a Person Following Robot

Yan Zhuang* Wei Wang* Yisha Liu* Yang Liu*

** Research Center of Information and Control, Dalian University of Technology, Dalian, China, (e-mail: zhuang@dlut.edu.cn)*

Abstract: A person following behaviour for a mobile robot with a new vision tracking algorithm is presented in this paper. According to the different characteristics of particle filter and Kalman filter, a novel approach of target tracking based on hybrid particle filters is applied to process the target object's position and shape component respectively, whose state updating is on the basis of data fusion between these two filter algorithms. The proposed method can not only pave the way for a low-complexity particle filter algorithm in dealing with higher dimensional tracking problem, but also cover the shortage of Gaussian restriction in Kalman filter. With the result of hybrid particle filter and the projective model of camera, the distance between the target and the robot can be calculated in real-time so that the robot can decide its own action to follow the target autonomously. A series of experiments on the Pioneer 3 robot show the method's validity and practicability.

1. INTRODUCTION

Fixating and following persons or objects with the eyes is a basic function that is essential for human perception. It implies that this skill is important to an autonomous mobile robot. Nowadays, many researches on person following robot are studied in different ways. Schulz and Montemerlo proposed a method of person following by using a distance sensor (Schulz et al., 2003; Montemerlo et al., 2002), but it is difficult to find out which is person and which is obstacle. Information obtained from vision sensors is more abundant as compared with distance sensors and it is easy to implement the target's identification. With the image sequence grabbed by a vision sensor, many algorithms are proposed to achieve a specific target's recognition and tracking.

A person following robot using vision sensors must has two fundamental abilities. The first one is to detect and track a person in a sequence of images. Many vision tracking algorithms have been proposed in recent years, such as optical flow, Kalman filter, particle filter, etc. For many methods that exploit optical flow, computational time for flow segmentation is long, and this problem is far from being solved; Kalman filter (KF) and extended Kalman Filter (EKF) are two common approaches for dealing with target tracking in probabilistic framework, but they can not resolve the tracking problem with multi-modal mode since their motion and measurement models both rely on Gaussian approximation. To resolve this challenging tracking problem in non-Gaussian measurement environment, particle filter (PF) has been introduced into the practice of computer vision over the past several years, where it is known as the CONDENSATION algorithm (Isard and Blake, 1998) and has been applied to great effect for tracking moving objects in image sequences (Katja *et al*., 2003). However, when the dimension of the state vector is high, the computational burden becomes a problem for a mobile robot with limited processing capacity. The second necessary ability for a person following robot is that the robot can calculate the person's position in the world coordinates based on the tracking result and make a decision in behaviour planning.

The aim of our work is to solve the problems mentioned above. In most situations, the shape of the target object is always unchangeable but its size may vary remarkably in different frames of the image sequence. Therefore when particle filter is used, the state vector must include the target's shape information. Obviously, it will increase the computational burden. In order to design an effective and practical colour target tracking system for our mobile robot, a hybrid particle filter (HPF) algorithm is proposed to track the target object's position and shape components respectively, which not only overcomes the linear and Gaussian restriction in Kalman filter, but also reduces the computational burden caused by the extension of system state's dimension in particle filter. Real-time data fusion between KF and PF is implemented to ensure the efficiency of the hybrid algorithm.

The remainder of this paper is organized in the following manner. After introducing Kalman filter and Particle filter in section 2, a foreground measurement method and a tracking algorithm named Hybrid Particle Filter are presented in detail in section 3. Section 4 shows how to implement HPF with a robot platform, section 5 presents some experiment results and finally section 6 concludes the paper.

2. RELATED WORK

2.1 Kalman Filter

KF is a recursive procedure that estimates the state of a dynamical system with the linear minimum error of mean square principle. If the state vector and the system noise are both independent Gauss process, KF is the square optimal

978-1-1234-7890-2/08/$20.00 © 2008 IFAC **10.3182/20080706-5-KR-1001.1415**

estimation. The recursive algorithm mainly depends on the equation of system state

$$X_{k+1} = \Phi_k X_k + W_k \tag{1}$$

and the equation of observation

$$Z_k = H_k X_k + V_k \tag{2}$$

where X_k is the N-dimensional state vector of the system; Z_k is the M-dimensional observation vector; the N × N-dimensional state transfer matrix Φ_k shows the relationship between the state vectors at time k and time k+1; H_k is a M × N-dimensional transfer matrix, which defines the transfer relationship between the system state and the observation vector when there is no system noise. W_k and V_k are random disturbance vectors for the system state equation and the observation equation respectively, on the assumption that both disturbance vectors are independent sequences of zero-mean Gauss white noise; the covariance matrixes are:

$$Q_k = E\left[W_k W_k^T\right], \quad R_k = E\left[V_k V_k^T\right] \tag{3}$$

The recursive procedure of KF to obtain the posterior estimation of X_k can be described as follows:

$$\hat{X}_k = \Phi_{k-1} X_{k-1} \tag{4}$$

$$X_k = \hat{X}_k + K_k (Z_k - H_k \hat{X}_k) \tag{5}$$

Here, K_k is the gain matrix of KF, which can be calculated by the following equations:

$$\hat{P}_k = \Phi_{k-1} P_{k-1} \Phi_{k-1}{}^T + Q_{k-1} \tag{6}$$

$$K_k = \hat{P}_k H_k{}^T (H_k \hat{P}_k H_k{}^T + R_k)^{-1} \tag{7}$$

$$P_k = \hat{P}_k - K_k H_k \hat{P}_k \tag{8}$$

2.2 *Particle Filter*

Particle filter (PF) is first proposed by Gordon in 1993 (Gordon *et al.*, 1993). It provides a practical way for solving non-Gaussian and nonlinear tracking tasks. It is an effective technique for implementing a recursive Bayesian approximation by Monte Carlo simulations. The kernel of this theory is exploiting a series of weighted particles $\{S_k^{(i)}, \omega_k^{(i)}\}_{i=1}^{N_s}$ to describe the posterior probability density function (PDF) at time *k* and updating the weights of the particles with the observation information. Every particle $S_k^{(i)}$ shows an assumed state of the system, $\omega_k^{(i)}$ is the discrete sample probability for $S_k^{(i)}$, and $\sum_{i=1}^{N_s} \omega_k^{(i)} = 1$. When the number of the particles is large enough, this discrete weighted estimation for the posterior state approximates to the Bayes optimum solution, therefore PF is a solution to the problem of state estimation. At time k, the prior PDF of the state is represented by a series of particles $\{S_k^{(i)}, \omega_k^{(i)}\}_{i=1}^{N_s}$. Each particle's weight is updated according to the observation at time k, and the particles close to the observed peak value are approximate to the real state of the system, so they are endued with bigger weight. After updating the weights, the particles represent the posterior distribution of the system state. To reduce the disturbance caused by the particles with small weights, resample is required. The particles with small weight will be ignored while the ones with big weight will be copied. Now, a loop of this iteration process is done.

The advantage of particle filter is that it can express arbitrary distribution and work well in nonlinear and non-Gaussian systems. But in practical application, a large number of particles are needed when the background is complicated. Since every particle is an assumed state of the system, the high computing cost obstructs the implementation of real-time tracking, especially when the dimension of the state vector is high. The hybrid particle filter algorithm presented in this paper can solve this problem and the detailed steps of this algorithm will be introduced in section 3.3.2.

3. FOREGROUND MEASUREMENT AND HYBRID PARTICLE FILTER ALGORITHM

Vision algorithm is the kernel of a person following robot. The target object could be identified by certain colour and shape, and these objective traits provide a physical basis for foreground measurement, which consists of pixel classification and target region segmentation in our research. After accomplishing foreground measurement, HPF is used to update the system state and track the target.

3.1 *Pixel Classification*

In order to identify the candidate objects from the background, pixels classification is the fundamental technique which includes threshold method, Gaussian-mixture model method and clustering method, etc. Considering the real-time and other constraints in practical application, we assume that the person who is tracked wears single colour clothes so that we can choose the constant threshold and reduce the computing cost in our method.

The output of a CCD camera is the RGB image. In order to enhance the algorithm's adaptability to the light intensity, the HSI colour space is used instead of RGB and the component of Intensity is not taken into account in our research. Pixel classification includes two steps. The first step is colour learning. The colour samples of the target are extracted and the thresholds of H and S are obtained through learning. The second step is pixel matching. All the pixels are compared with the thresholds of H and S, then the pixels are classified into two types: the one belong to the object and the one not.

3.2 *Target Region Segmentation*

After pixel classification, it is necessary to analyse the connectedness of the pixels which belong to the target. The target pixels are separated into several independent regions. A tree-based run length encoded (RLE) method (Bruce *et al.*, 2000) is employed in this paper. The region segmentation procedure includes two steps. The first step is to compute a

RLE version for the classified image. By grouping adjacent target pixels as a single "run", the efficiency of the algorithm is improved because subsequent users of data can operate on entire "runs" rather than individual pixels. In the second step, the merging method employs a tree-based "union find" with path compression. The merging is performed in place on the classified RLE image. Initially, each "run" labels itself as its parent, resulting in a completely disjoint forest, and then the merging procedure scans adjacent rows and merges "runs" which are connected to each other. This yields a disjoint forest, and each tree represents an independent connected region. In our experiment, the connected regions are expressed by the rectangles. The observation characteristic vector of the rectangle $Z_k = [x_k \quad y_k \quad w_k \quad h_k]^T$ is provided for the tracker, where (x_k, y_k) is the rectangle's center and (w_k, h_k) is the rectangle's width and height.

3.3 *Hybrid Particle Filter Algorithm*

The final goal of a vision tracking algorithm is to estimate the target's position in each frame accurately and robustly, nevertheless the factor in target's shape also draws our attentions. The system state vector constructed in our work includes both the object's position and the object's shape, which is very helpful in the preprocessing of weight updating in HPF algorithm. Object's shape component in the system state not only guarantees the accuracy in observation's matching and robustness in target tracking, but also plays an important role in the procedure of target locating.

3.3.1 Target Dynamic Model

We consider the motion of target as the discrete-time 2-dimensional nearly constant velocity (NCV) motion model (Coraluppi and Grimmett, 2001). In our task, the state of the motion model is expanded to be a 4-dimensional vector and the state propagation equation is:

$$X_{k+1} = \Phi_k X_k + W_k \tag{9}$$

and

$$X_k = [S_k \quad B_k]^T = [x_k \quad y_k \quad \dot{x}_k \quad \dot{y}_k \quad w_k \quad h_k \quad \dot{w}_k \quad \dot{h}_k]^T \tag{10}$$

$$\Phi_k = \begin{bmatrix} \Phi'_k & O_4 \\ O_4 & \Phi'_k \end{bmatrix} = \begin{bmatrix} I_2 & \Delta t_k I_2 & O_2 & O_2 \\ O_2 & I_2 & O_2 & O_2 \\ O_2 & O_2 & I_2 & \Delta t_k I_2 \\ O_2 & O_2 & O_2 & I_2 \end{bmatrix} \tag{11}$$

The position component in X_k is $S_k = [x_k \quad y_k \quad \dot{x}_k \quad \dot{y}_k]$, where (x_k, y_k) is the center of target region at time k, while $(\dot{x}_k, \dot{y}_k)$ represents the motion; Correspondingly, the object's shape component in X_k is $B_k = [w_k \quad h_k \quad \dot{w}_k \quad \dot{h}_k]$, where w_k, h_k represent the width and height of object's rectangular boundary in foreground measurement respectively, while $\dot{w}_k, \dot{h}_k$ is their corresponding scale change; I_n and O_n are n×n identity and zero matrices; $\Delta t_k = t_{k+1} - t_k$ is the time interval between frames. $\{W_k, k \geq 1\}$ is a discrete-time white Gaussian noise with $W_k \sim N(0, Q_k)$, where, in this case, $Q_k = \begin{bmatrix} Q_{11} & O_4 \\ O_4 & Q_{22} \end{bmatrix}$, $Q_{11} = \begin{bmatrix} \frac{1}{3}\Delta t_k^3 diag(q_{x_k}, q_{y_k}) & \frac{1}{2}\Delta t_k^2 diag(q_{x_k}, q_{y_k}) \\ \frac{1}{2}\Delta t_k^2 diag(q_{x_k}, q_{y_k}) & \Delta t_k diag(q_{x_k}, q_{y_k}) \end{bmatrix}$, $Q_{22} = \begin{bmatrix} \frac{1}{3}\Delta t_k^3 diag(q_{w_k}, q_{h_k}) & \frac{1}{2}\Delta t_k^2 diag(q_{w_k}, q_{h_k}) \\ \frac{1}{2}\Delta t_k^2 diag(q_{w_k}, q_{h_k}) & \Delta t_k diag(q_{w_k}, q_{h_k}) \end{bmatrix}$ and q_{x_k}, q_{y_k}, q_{w_k}, q_{h_k} are noise covariance gains of x_k, y_k, w_k, h_k, respectively. The target measurement equation at time k is described by

$$Z_k = H_k X_k + V_k \tag{12}$$

The measurement vector Z_k can be expressed as $Z_k = [U_k \quad Y_k]^T = [x_k \quad y_k \quad w_k \quad h_k]^T$, where $U_k = [x_k \quad y_k]$ and $Y_k = [w_k \quad h_h]$ are position and shape components respectively. The transfer matrix H is

$$H_k = \begin{bmatrix} H'_k & O_{2\times4} \\ O_{2\times4} & H'_k \end{bmatrix} = \begin{bmatrix} I_2 & O_2 & O_2 & O_2 \\ O_2 & O_2 & I_2 & O_2 \end{bmatrix} \tag{13}$$

where $O_{2\times4}$ is 2×4 zero matrix; $\{V_k, k \geq 1\}$ is a zero-mean independent Gaussian noise with $V_k \sim N(0, R_k)$, where $R_k = diag(\sigma_x^2(k), \sigma_y^2(k), \sigma_w^2(k), \sigma_h^2(k))$.

3.3.2 Algorithm

Considering the high dimension of the system state in our task and the computational cost of PF algorithm in complex non-Gaussian and nonlinear target tracking, how to reduce the computational burden and improve tracking algorithm's practicability are the crucial problems in our research. In this paper, a novel algorithm, hybrid particle filter, is presented to solve the problems mentioned above. HPF algorithm combines Particle filter and Kalman filter to process system state's position component and shape component respectively, and the data fusion between these two filter algorithms makes system state updating more reasonable and effective.

System state is composed of position component and shape component. PF algorithm is used to estimate object's position recursively; KF algorithm is used to estimate the width and height of object's rectangular boundary in current time, the estimating result of which is also the basis of particle weight's updating in PF. Correspondingly, the state in KF is updated by using the measurement information derived from the posterior estimation of target's position. The using of KF in HPF reduces the computational burden greatly and the combination of PF and KF makes the real-time and robust target tracking practicable in non-Gaussian measurement environment.

In PF algorithm, $S^{(i)} = [x^{(i)} \quad y^{(i)} \quad \dot{x}^{(i)} \quad \dot{y}^{(i)}]^T$ is a sample (a particle) which represents an assumed position state of the target object and is described by a rectangular boundary line. The predicted measurement vector $\hat{U}^{(i)} = [\hat{x}^{(i)} \quad \hat{y}^{(i)}]^T$ of particle $S^{(i)}$ can be got from the system state propagation equation and the target measurement equation. The predicted measurement vector $\hat{Y} = [\hat{w} \quad \hat{h}]^T$ of current target object's rectangular boundary can also be got from Kalman filter. $\left\{Z^{(j)} = [U^{(j)} \quad Y^{(j)}]^T = [x^{(j)} \quad y^{(j)} \quad w^{(j)} \quad h^{(j)}]^T\right\}_{j=1}^{M}$ are M measurement vectors totally which is the result of foreground measurement. In order to update particle's weight, it is necessary to find out each particle's optimum matching in practical observation. Mahalanobis rule is used to preprocess the candidates from foreground measurement according to the predicted shape measurement vector, and the candidates that have the similar rectangular boundary are reserved. Then the minimum Euclidian distance is used to achieve particle's least uncertainty measurement among the remainder candidates. For a single particle, the reserved candidate which has the minimum Euclidian distance to it is its least uncertainty measurement.

In our HPF algorithm, the position component $U^{(m)}$ of particle's least uncertainty measurement $Z^{(m)}$ is used to update particle's importance weight. According to the model of two-dimensional Gaussian joint distribution, the weight updating formula of sample $S^{(i)}$ is

$$\omega^{(i)} = \frac{1}{2\pi \left|H_k R_k H_k^T\right|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\hat{U}^{(i)} - U^{(m)})^T (H_k R_k H_k^T)^{-1} (\hat{U}^{(i)} - U^{(m)})\right\} \quad (14)$$

From (14) we can conclude that the particles with smaller distance in the least uncertainty measurement will be assigned larger weights. During the resampling step in PF, samples having a high importance weights may be multiplied several times, leading to identical copies, while others with relatively low importance weights may be eliminated since they have little contribution to the posterior estimation of the target's state.

The output of position component can be expressed by MAP (Maximum A Posteriori) estimation $E(S) = S_{MAP}$. Compared with weighted mean estimation $E[S] = \sum_{i=1}^{N_s} \omega^{(i)} S^{(i)}$ in other work, the state's MAP estimation can suppress interference from spurious measurement and the convergent point of particles' distribution is selected as the posterior estimation, which makes the tracking result with HPF algorithm more objective to true value.

Suppose target object's state posterior estimation at time k-1 is $X_{k-1} = [S_{k-1} \quad B_{k-1}]^T$, where the position component S_{k-1} is represented by a set of N weighted samples $\{S_{k-1}^{(i)}, \omega_{k-1}^{(i)}\}_{i=1}^{N_s}$. Given that P_{k-1} is the posterior error covariance of shape component B_{k-1}, the recursive update in HPF is realized in the following 6 steps:

(1) *Resampling in PF:* Resample with replacement N particles $\{S_{k-1}'^{(i)}, 1/N_s\}_{i=1}^{N_s}$ from the set $\{S_{k-1}^{(i)}, \omega_{k-1}^{(i)}\}_{i=1}^{N_s}$ according to the probability $\Pr(S_{k-1}'^{(i)} = S_{k-1}^{(j)}) = \omega_{k-1}^{(j)}$.

(2) *State Prediction in PF and KF:* The predictive output of particle set $\{S_{k-1}'^{(i)}, 1/N_s\}_{i=1}^{N_s}$, posterior estimation of shape component B_{k-1}, and posterior covariance estimation P_{k-1} are given by $\hat{S}_k^{(i)} = \Phi_{k-1}' S_{k-1}'^{(i)} + [I_4 \quad O_4] W_{k-1}$, $\hat{B}_k = \Phi_{k-1}' B_{k-1}$ and $\hat{P}_k = \Phi_{k-1}' P_{k-1} \Phi_{k-1}'^T + [O_4 \quad I_4] Q_k [O_4 \quad I_4]^T$ respectively. The new particle set of shape component is $\{\hat{S}_k^{(i)}, 1/N_s\}_{i=1}^{N_s}$; the state prediction of shape component is $\hat{B}_k$; the corresponding prediction of state covariance is $\hat{P}_k$.

(3) *Measurement Prediction in PF and KF:* The predictive measurement vector set $\{\hat{Z}_k^{(i)}\}_{i=1}^{N_s}$ of current particle set $\{\hat{S}_k^{(i)}\}_{i=1}^{N_s}$ and the measurement prediction $\hat{Y}_k$ of $\hat{B}_k$ are computed by $\hat{Z}_k^{(i)} = H_k' \hat{S}_k^{(i)}$ and $\hat{Y}_k = H_k' \hat{B}_k$.

(4) *Weight update and output in PF:* Based on the methods presented above, we can find the corresponding least uncertainty measurement $Z_k^{(m_i)}$ for each particle $S_k^{(i)}$ from the current foreground measurement vector set $\{Z_k^{(i)}\}_{i=1}^{M}$. Update the weights of particle set $\{\hat{S}_k^{(i)}, 1/N_s\}_{i=1}^{N_s}$ by using (16) and normalize the weights with $\omega_k^{(i)} = \omega_k^{(i)} / \sum_{i=1}^{N_s} \omega_k^{(i)}$, then we can get $\{S_k^{(i)}, \omega_k^{(i)}\}_{i=1}^{N_s}$. MAP estimation of object's position component is $E(S_k) = S_{kMAP}$.

(5) *Gain computation and state update in KF:* Kalman gain can be computed from the equation of $K_k = \hat{P}_k H_k'^T (H_k' \hat{P}_k H_k'^T + [O_2 \quad I_2] R_k [O_2 \quad I_2]^T)^{-1}$. Find the least uncertainty measurement $Z_k^{(m)}$ of S_{kMAP} from foreground measurement vector set $\{Z_k^{(i)}\}_{i=1}^{M}$ as the actual observation in KF, and then update the state in KF with $B_k = \hat{B}_k + K_k (Z_k^m - \hat{Y}_k)$. State covariance can be updated by $P_k = \hat{P}_k - K_k H \hat{P}_k$.

(6) *Output of HPF:* Posterior estimation of target object's state is $X_k = [S_{kMAP} \quad B_k]^T$ at time k.

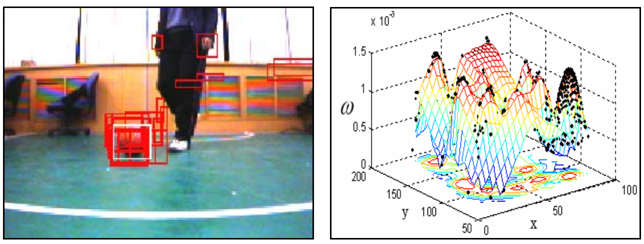

Fig 1. MAP estimation of target object (Left), distribution of particles and corresponding weights allocation (Right)

A practical application case of HPF algorithm is shown in Figure 1. To test the validity of this algorithm in non-Gaussian measurement environment, a moving red ball is tracked with a stationary camera. The MAP estimation of current state is illustrated with a white square in the left image, while the distribution of particles and corresponding weights allocation based on (16) is illustrated in the right image.

4. IMPLEMENTATION OF HYBRID PARTICLE FILTER ON A PERSON FOLLOWING ROBOT

The algorithm presented above is based on the assumption that the camera is stationary. But on a person following robot, the location of the camera must be changed while the robot is moving. In this situation, it is not enough to predict the state vector of PF only with the system state propagation equation, so HPF can not be used directly. Image-forming principle of CCD camera is applied in a method which can reduce the influence caused by the movement of camera and predict the position of target in different image planes.

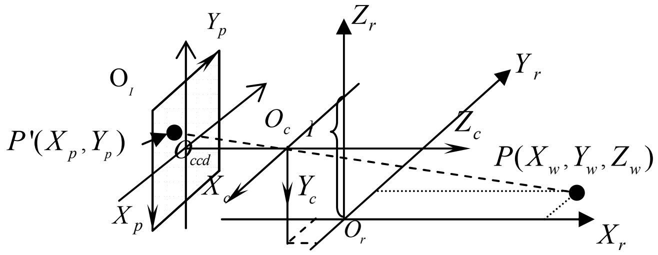

Fig.2. The pinhole model of CCD camera

The camera is modelled as a pinhole (Figure 2). The subscript r and c mean the camera coordinates and the robot coordinates. If the coordinates of a 3D point is known, it is easy to calculate the corresponding point in the image plane. The procedure of image-forming is introduced in many literatures, so it will not be repeated in this paper. According to the image-forming procedure, we can deduce a point on the ground in robot coordinates by using its pixel coordinates (x_p, y_p) and camera's intrinsic parameters f_{cx}, f_{cy}, c_x, c_y:

$$x_r = \frac{f_{cx} * H * \cos(\alpha) + H * \sin(\alpha) * (y_p - c_y)}{(y_p - c_y) * \cos(\alpha) - f_{cy} * \sin(\alpha)} \tag{15}$$

$$y_r = \frac{(x_p - c_x) * (H * \sin(\alpha) - x_r * \cos(\alpha))}{f_{cx}} \tag{16}$$

where H is the height of camera installed on the mobile robot and α is camera's elevation angle. After we get the point's position in robot coordinates, it is easy to transform it to the world coordinates $p(x_w, y_w, 0)$.

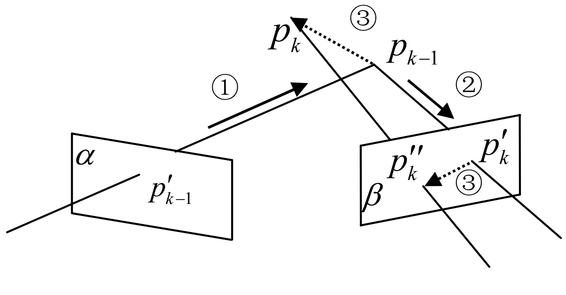

Fig.3. The process of prediction for the moving object in different image planes

The prediction for moving object in different image planes consists of three steps which are briefly shown in Fig 3:

(1) At time k-1, the object's position in the world coordinates is p_{k-1} and its projection in image plane α is p'_{k-1}. The world coordinates of p_{k-1} can be deduced from the pixel coordinates of p'_{k-1} by using (15) and (16).

(2) At time k, mobile robot moves to a new position and the new image plane is β. Suppose that the object is immovable and project p_{k-1} to image plane β, then we could get the projection result p'_k.

(3) Considering the object's movement in measurement environment, object moves from p_{k-1} at time k-1 to the new position p_k at time k. In order to provide enough measurement information for later target tracking, p_k is projected to β again to predict the new pixel position in the image plane.

After the above three steps, we complete the pixel position prediction in different image planes. Now HPF is available, and the moving person can be tracked by a moving camera on the robot.

To follow a person, the robot must turn to the direction of the person first. If (x_p, y_p) represents the position of the person in the image plane and (x_0, y_0) is the center of image, the angle that the robot has to turn is $k_p \times (x_p - x_0)$, where k_p is a proportion factor. By this way, the person's image can be always kept in the center of image planes. With the output of HPF, it is easy to obtain the person's position in the robot coordinate system using (15) and (16), and then the distance information between the person and the robot is known. If the distance is greater than the maximum distance we set, the robot moves forward. On the contrary, if the distance is less than the safe distance, the robot falls back.

5. EXPERIMENTAL RESULTS

The vision system in our experiment includes a Canon VC-C50i CCD colour camera mounted on Pioneer 3 robot and a frame grabber with BT848 chip. Two tracking applications are implemented which demonstrate the efficiency of the HPF algorithm in a person following robot, including fixating a person in laboratory and following a person in the corridor.

5.1 Fixating Experiment

In this experiment, the initial position of the person is in the right part of the image. When the person's image is moving from right to the left and deviates from the image center, the robot rotates and changes its direction to keep the person in the image center. The initial position is given by clicking on the image by a mouse, and a rectangle area is also selected by the mouse to decide the thresholds of the H and S component. 150 particles are exploited in our filter algorithm and the result is shown in Figure 4.

Fig.4. Image sequence from digital video shows the results in fixating experiment.

The image size is 192×144. In the first frame, the person is far from the center, and then the robot turns to fixate him. In the following frames, both the true value and the estimated value are close to the expected value, which is the center of the y-axis in the image plane. If the position of person ranges from 76 to 116 on the y-axis, the robot keeps stationary. Fig 5 is the plotting of the estimated value and true value in fixating experiment.

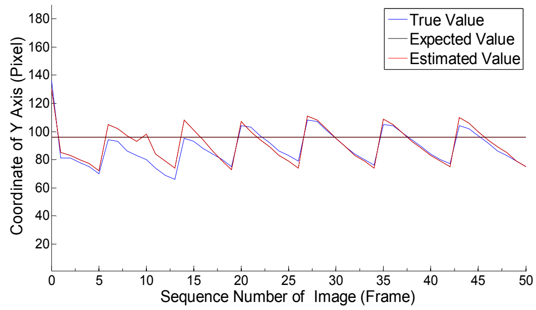

Fig.5. Plotting of the estimated value and true value in fixating experiment.

5.2 Following Experiment

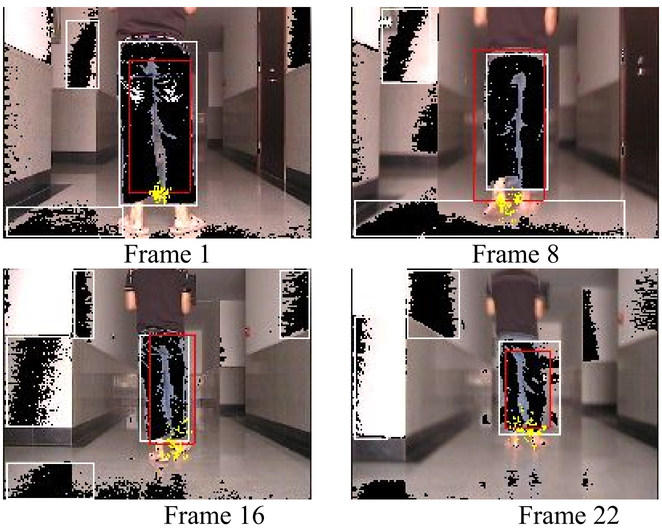

Fig. 6. Tracking results in person following experiment.

This experiment shows the person following performance of the robot. When a person goes forward in the corridor, the robot follows behind him. The robot controls its velocity to keep the distance between them in a suitable range. The initial position of target and the colour threshold are also obtained by clicking on the image by a mouse. Figure 6 shows that the tracking algorithm used in this experiment is effective and it is practical for a person following robot. As shown in figure 6, the tracking target is the person's legs and pixels detected as target are marked by black. The white rectangles are candidate observations. The red rectangle is the output of HPF. The yellow points are particles.

6. CONCLUSIONS

A practical method for target tracking in noisy scene has been proposed and applied to a person following robot. Hybrid particle filter fuses the data of PF and KF, and tracks a target in non-Gaussian measurement environment with a low computational burden successfully. The pinhole model of the camera is used to predict the target's position in different image planes, which makes an important contribution to the implementation of HPF on a mobile robot. The experimental results illustrate that the proposed method can effectively track the target, even when the camera is not stationary. Further work is planned to perform the proposed method to realize multi-target tracking in the dynamic environments.

7. ACKNOWLEDGEMENT

This work was supported by National Natural Science Foundation of China (Grant No. 60605023, 60775048) and the National High Technology Research and Development Program of China (Grant No. 2007AA04Z257).

REFERENCES

Bruce, J., Balch, T., *et al.* (2000). Fast and inexpensive color image segmentation for interactive robots. *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2061-2066.

Coraluppi, S. and Grimmett, D. (2001). Intra-ping timing issues in multistatic sonar tracking. *The 7th International Conference on Information Fusion s*, pp. 510-517.

Gordon, N., Salmond, D., *et al.* (1993). Novel approach to nonlinear and non-Gaussian Bayesian state estimation. *Proc. Inst. Elect. Eng*, pp. 140: 107–113.

Isard, M. and Blake, A. (1998). Condensation-Conditional density propagation for visual tracking. *Int. J. Computer Vision*, pp. 29(1) , pp. 5-28.

Katja, N., Esther, K. M., *et al.* (2003). Color features for tracking non-rigid objects. *Acta Automatica Sinica*, 29 (3), pp. 345-355.

Montemerlo, M., Thrun, S., *et al.* (2002). Conditional particle filters for simultaneous mobile robot localization and people tracking, *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 695- 701.

Schulz, D, Burgard, W., *et al.* (2003). People tracking with a mobile robot using sample-based joint probabilistic data association filters. *Int. Journal of Robotics Research*, vol.22, no.2, pp.99-116.