2012 AASRI Conference on Modeling, Identification and Control

Improved method based on MPF for multi-target tracking

Qing Lin[1,a], Yaping Yang[*2,b], Ying Yin[3], Shitong Wang[4] and Dingan Liao[5]

[1,2,4,5]*School of Computer Science and Telecommunications Engineering, Jiangsu University, Zhenjiang, 212013, China*
[1]*School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing, 210094, China*
[3]*School of Information Technology, Southern Yangtze University, Wuxi 214122, China*

Abstract.

Particle filter algorithm is poor at consistently maintaining the multi-modality problem and remains particle degeneracy phenomenon. This shortcoming can be addressed through using Mixture Particle Filter (MPF) to model the target distribution as a non-parametric mixture model. According to computational complexity of the particle filter algorithm, Gaussian Particle Filter (GPF) is used. The particle set is obtained by sampling of Gaussian density function instead of re-sampling to improve the computing speed of particle filter algorithm. In this paper, combined with the advantages of GPF and MPF, GM-MPF algorithm is proposed to improve the real-time of MPF. The experimental results show that GM-MPF algorithm can effectively solve the problem of multi-modality, particle degradation and achieve real-time multi-target tracking.

keywords: Particle filter, Mixture Particle filter, Gaussian Particle Filtering, Particle degradation, GM-MPF.

1. Introduction

Object detection and tracking is an important task in the field of computer vision. Object detection and tracking technology is one of the critical technologies in computer vision. Object detection and tracking has been applied to computer vision applications domain such as video surveillance, traffic control. Object tracking method based on filtering theory is that object tracking transforms to probability density function estimation in image sequences, for example, kalman filter [1] or particle filter [2-4] to track the object. The linear Gaussian system can only be solved by kalman filtering. However, this linear Gaussian assumption does not often establish due to abrupt object motion, camera motion, changing appearance patterns of both object and scene, image noise influence, illumination changes, the complex scene. Particle filtering method is sample-based conditional density propagation by learning the characteristics of dynamic model and visual observation model. It shows good performance to solve non-linear and non-Gaussian problem and deals effectively with object occlusion. Particle filtering has become to a powerful tracking algorithm in the field of object tracking.

Monte Carlo characteristic of particle filter requires a set of particles to accurately express the probability distribution. The calculation can be expensive. The expression of the HSV color information is employed to reduce the computational cost of

Corresponding author. Tel.:+86-18796083529
E-mail address: xgkx06s003@126.com   ,jcs533@qq.com

each particle [5] in the paper. Monte Carlo-based particle filter algorithm is poor at consistently maintaining the multi-modality problem and remains serious particle degeneracy phenomenon. In recent years contraposing its degeneration phenomenon, the sequence importance sampling (SIS) and the sequence importance resampling (SIR) technique has been proposed. Gaussian particle filter (GPF) [6] algorithm that Gaussian distribution approximates posterior distribution is an improved particle filtering algorithm to reduce the particle computational cost and enhance the real-time. The method is also applies to solve non-Gaussian systems and does not require re-sampling to avoid particle depletion. The jaco vermaak et al proposed Mixture particle filter [7] to solve the multi-modal and non-Gaussian systems. Combined the ability of Mixture particle filter to maintain multi-modality effectively with real-time requirements of GPF, GM-MPF algorithm is proposed in this paper.

2. Mixture Tracking

2.1. Observation model

In this paper HSV color space model is employed to model moving object tracking and observation model to reduce the computational cost. Assume that HSV histogram contains N gray-level ($N = N_h N_s + N_y$).D pixel gray-level index at time t can be represented by $b_t(d) \in \{1, \cdots, N\}$ corresponding to the color vector $y_t(k)$. Color probability nuclear density function estimation of the candidate region, that is, the probability of d pixel gray-level index at time t is formulated in the paper.

$$k(n; x_t) = \eta \sum_{d \in R(x_t)} \delta[b_t(d) - n].$$ (1)

Where δ is the Dirac delta, η is normalization constant. With $\sum_{n=1}^{N} k(n; x_t) = 1$ .Bhattacharyya coefficient is used in this paper to measure the similarity (2) between the candidate region $K(x_t)$ and the Object region $K*$.

$$\xi[K*, K(x_t)] = [1 - \sum_{n=1}^{N} \sqrt{k*(n; x_0)k(n; x_t)}]^{\frac{1}{2}}$$ (2)

Where $k(n; x_t)$ obtained by formula (1) represents the candidate region Color model. $K* = \{k*(n, x_0)\}_{n=1,\cdots N}$ represents reference color model. Finally similarity distribution $L(y_t | x_t)$ is computed via formula (3).

$$L(y_t | x_t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\xi^2}{2\sigma^2}}.$$ (3)

2.2. State transition model.

In this system, particle filter theory is applied to track object of video. Each object is indicated with rectangle to determine the location and size of the object. Secondly, the system equation is described by a state equation and an observation equation. The speed and direction of object motion associates with that of few frames before, so there are not abrupt changes. In this paper second-order autoregressive process is used to represent the state equation mutation and observation equation is mainly based on the HSV color histogram model.

$$\text{Observation equation: } X_t - \bar{X} = A1(X_{t-1} - \bar{X}) + A2(X_{t-2} - \bar{X}) + B_0 \omega_t.$$ (4)

Where $\bar{X}$ represents the mean of all particle state estimates, X_t represents the particle state at the moment t. ω_t the random noise A_1 、 A_2 、 B_0 the state transition matrix.

3. Gaussian particle filter (GPF)

PF whose essential idea is Monte Carlo simulation is proposed to solve the nonlinear and non-Gaussian problem. A set of

weighted particles $\left\{x_{t-1}^{(i)}, \omega_{t-1}^{i}\right\}_{i=1:N}$ is used to approximate the posterior probability density $p\left(x_t \middle| y^t\right)$. Ideally, the weight of every particle should be equal, so the posterior density will be only characterized by the number of the particle in the state space. GPF is to approximate the posterior distribution to avoid particle depletion.

GPF is very similar to the SIS (sequence of importance sampling) via importance sampling distribution to sample particles set $\left\{x_t^i\right\}_{i=1,\ldots,N}$. Unlike SIS, however, filter distribution and predict distribution of the GPF is approximated by the Gaussian density distribution, which reduces the complexity of particle calculation. The sequence updated filtering distribution can be expressed as

$$D\left(x_t^{(i)} \middle| x_{t-1}^{(i)}\right) \approx \mathrm{N}\left(x_t = x_t^{(i)}; \bar{\mu}_t, \bar{\Sigma}_t\right).$$ *(5)*

GPF algorithm reduces the complexity but remains particle degeneracy phenomenon. In previous experiments, we learned that PF can not solve multi-modality problem. The corresponding experiment result is shown in the experimental section. During multi-target tracking process, particles will transfer to one object, so PF algorithm can not be achieved multi-target tracking and MPF is proposed.

4. Gaussian model - Mixture particle filter (GM-MPF)

As mentioned above, PF can not handle multi-modality. Jaco vermaak, et al proposed MPF to solve the multi-modality and non-Gaussian system. In mixture tracking process, we let $P_t = \left\{N, M, \Pi_t, \chi_t, \omega_t, c_t\right\}$ denote particle representation of the mixture particle filter distribution. With N the number of particles, M the total number of mixture particle filter, $\Pi_t = \left\{\pi_{m,t}\right\}_{m=1}^{M}$ mixture component weights and $\sum_{m=1}^{M}\pi_{m,t} = 1$. $\chi_t = \left\{x_t^{(i)}\right\}_{i=1}^{N}$ denotes the particle state, $\omega_t = \left\{\omega_t^{(i)}\right\}_{i=1}^{N}$ represents the particle weight and $\sum_{i=1}^{N}\omega_t^{(i)} = 1$. $c_t = \left\{c_t^{(i)}\right\}_{i=1}^{N}$ the component index, $c_t^{(i)} \in \{1..M\}$, with $c_t^{(i)} = m$ if i belongs to mixture m-component.

Calculate m mixture particle filters distribution: $p\left(x_t \middle| y^t\right) = \sum_{m=1}^{M}\pi_{m,t}p_m\left(x_t \middle| y^t\right)$ with $\sum_{m=1}^{M}\pi_{m,t} = 1$

The mixture component distribution can be computed using the two step recursion:

predict: $p\left(x_t \middle| y^{t-1}\right) = \sum_{m=1}^{M}\pi_{m,t-1}p_m\left(x_t \middle| y^{t-1}\right).$ *(6)*

update: $p\left(x_t \middle| y^t\right) = \sum_{m=1}^{M}\pi_{m,t}p_m\left(x_t \middle| y^t\right).$ *(7)*

Filtering distribution of MPF is regarded as m independent particle filter and the posterior distribution is approximately estimated by Gaussian distribution. The choice of Gaussian filter importance function depends on the specific circumstances, but under normal circumstances, $q(\bullet) = N(x_t; \bar{\mu}_t, \bar{\Sigma}_t)$. So let $p_m\left(x_t \middle| y^t\right) \approx \sum_{m=1}^{M}g^{(m)}N\left(x_t; \bar{\mu}, \bar{\Sigma}\right)$ denote the new filtering distribution for the m-component. With M a Gaussian component, $g^{(m)}$ the Gaussian component weight. $N(x_t; \bar{\mu}_t, \bar{\Sigma}_t)$ denotes variable Gaussian distribution.

Multi-target tracking is realized via combining the Gaussian distribution with MPF in this paper. MPF algorithm is applied in tracking m objects with N particles, so Each object ,m independent filter, is allocated M = N/m particles, Particles are divided into m intervals in this paper. The posterior distribution of each interval is approximately estimated by Gaussian distribution to realize multi-target tracking. The concrete realization is as follows:

1. Initialization. Weighted particles for m intervals $\left\{x_t^{(i)}, \omega_t^{(i)}\right\}_{i=1:N}^{c}$ $c \in (1,\ldots,m)$ are obtained from sampling the prior

density $p\left(x_0^{(i)}\right)$.

2. Measurement updated algorithm

(1) Draw Samples from the importance function $q\left(x_t^{(i)} \middle| x_{t-1}^{(i)}, y_t\right)$ and denote them as the particle set $\left\{x_t^i\right\}_{i=1,\ldots,N}$

(2) Compute weights and normalize the weights of m-component particle as

$$\overset{\sim}{\omega_t}{}^{(i)} = \frac{L\left(y_t \middle| x_t^{(i)}\right) \sum_{m=1}^{M} \pi_{m,t} \sum_{m=1}^{M} g^{(m)} N\left(x_t; \overline{\mu}, \overline{\overline{\Sigma}}\right)}{q\left(x_t^{(i)} \middle| x_{t-1}^{(i)}, y_t\right)} . \tag{8}$$

$$\omega_t^{(i,\ c)} = \frac{\overset{\sim}{\omega_t}{}^{(i)}}{\sum_{j=1}^{M} \overset{\sim}{\omega_t}{}^{(j)}}, i = 1,\ldots,M; c \in (1,\ldots,m) . \tag{9}$$

(3) Estimate mean and covariance at t time for m-component as GPF

$$\mu_t^{(c)} = \sum_{i=1}^{M} \omega_t^{(i,\ c)} x_t^{(i)}, c \in (1,\ldots,m) . \tag{10}$$

$$\Sigma_t^{(c)} = \sum_{i=1}^{M} \omega_t^{(i,\ c)} \left(\mu_t^{(c)}\ x_t^{(i)}\right)\left(\mu_t^{(c)}\ x_t^{(i)}\right)^{H}, c \in (1,\ldots,m) . \tag{11}$$

3. Time updated algorithm

(1) Draw Samples from the posterior probability density $N\left(x_t; \overline{\mu}_t^{(c)}, \overline{\Sigma}_t^{(c)}\right)$ in the measurement updated algorithm at time t and denote as particle set $\left\{x_t^i\right\}_{i=1,\ldots,N}$.

(2) Draw Samples from the state transition distribution denote as particle set $\left\{x_{t+1}^i\right\}_{i=1,\ldots,N}$.

(3) Estimate mean and covariance for m-component as GPF

$$\overline{\mu}_{t+1}^{(c)} = \frac{1}{M} \sum_{i=1}^{M} x_{t+1}^{(i)}, c \in (1,\ldots,m) . \tag{12}$$

$$\overline{\Sigma}_{t+1}^{(c)} = \frac{1}{M} \sum_{i=1}^{M} \left(\overline{\mu}_{t+1}^{(c)}\ x_{t+1}^{(i)}\right)\left(\overline{\mu}_{t+1}^{(c)}\ x_{t+1}^{(i)}\right)^{H}, c \in (1,\ldots,m) . \tag{13}$$

4. t = t +1, Continue to track and go to (2); otherwise the algorithm ends.

5. Results and analysis

The experiment is based on the OPENCV computer vision library and is implemented in the visual studio 2008 software platform. The video comes from the PETS 2001 / DATA SET [8] and there are multiple targets in 350 frame, The experiment is to remove the previous 350 frames and starts to process from 351 frame at first. Gaussian mixture modeling background method is used to detect target and the region of interest [9] is marked in experiment. In the process of object tracking, particle filter is applied to solve the nonlinear and non-Gaussian problem. In the paper 100 particles are employed to achieve two objects tracking. Each target can be obtained 50 particles. More and more objects tracking can be achieved by changing the coefficient of the target number and the number of particles during the experiment. This paper compares the improved particle filter with improved mixture particle filter to achieve two objects tracking. The result is shown as Figure 1 and 2.

It is shown that the vast majority of particles after the 27 frame have been transferred to another target in Figure 1 and all have been transferred in the 29. Obviously improved particle filtering algorithm is poor in maintenance of multi-modal distribution of the target. However, mixture particle filter algorithm models the filtering distribution as a mixture model to better cope with the multi-modality. In Figure 2 it is observed that this mixture particle filter is able to maintain the

multi-modality inherent in tracking problem and multi-target tracking is realized stably where the standard particle filter fails.

Now we define a metric measure N_{eff} of the degradation: $N_{eff} = \dfrac{N}{1 + \text{var}(\omega_k^i)}$ .Where $\text{var}(\omega_k^i)$ is the variance of ω_k^i .

Generally N_{eff} can not be calculated exactly, so it is approximated as

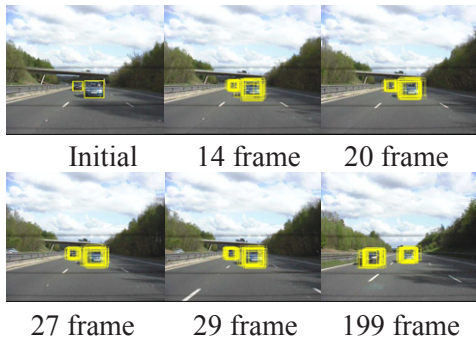$$N_{eff} = \frac{1}{\sum\limits_{i=1}^{N_t} \left(\omega_k^i\right)^2} . \tag{14}$$

Fig 1 Multi-target tracking based on improved PF
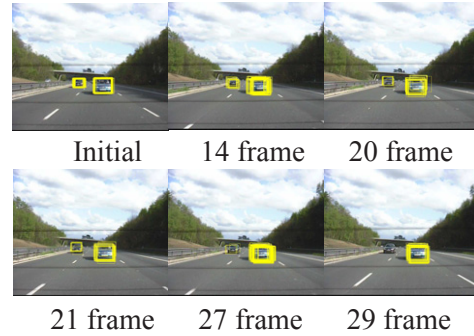
Fig 2 Multi-target tracking based on improved MPF

The experimental result is shown in Figure 3 and 4. The experimental results show that mixture particle filter method can effectively solve the problem of multi-modality, simplify particle degradation and achieve steady multi-target tracking.
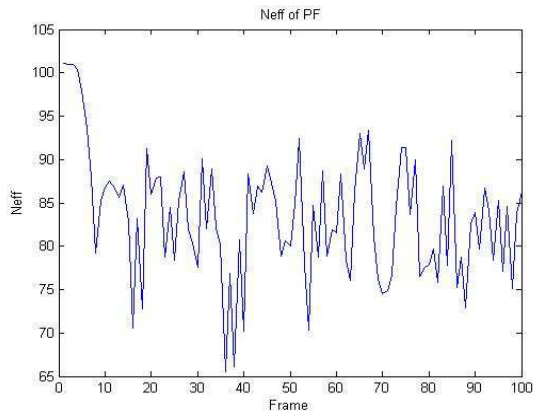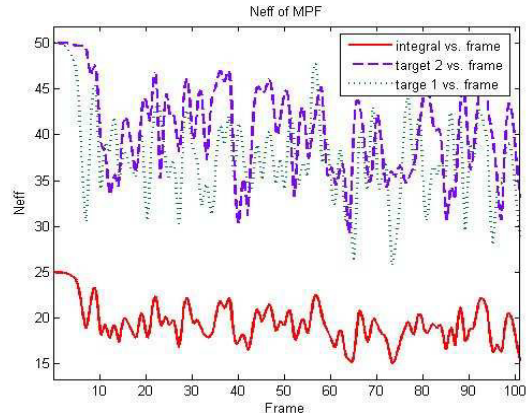

Fig 3 Neff of PF to track

Fig 4 Neff of MPF to track

Root mean square error (RMSE) is defined as $RMSExf = \dfrac{1}{N} \sum\limits_{n=1}^{N} \|x_t - \overline{x}_t\|$, $\overline{x}_t = E(x_t | y_{0:t})$. RMSE curves with the different

number of particles (100, 200, 300) are shown in Figure 5 and Figure 6.

It is evident from the graph that root mean square error RMSE of (N = 200 and 300) is smaller than RMSE of (N = 100). The smaller number of particles is, the larger tracking error is, but the higher computational complexity is. With the increase of the number of particles, algorithm tracking accuracy is improved greatly and the time of the algorithm computation also increases. The average processing time of 50 frames with 1000 particles is compared among the four kinds of algorithms in Table 1. Gaussian particle filter (GPF) algorithm that Gaussian distribution approximates posterior distribution is an improved particle filtering algorithm to reduce the particle computational cost. GM-MPF can enhance the real-time of MPF.
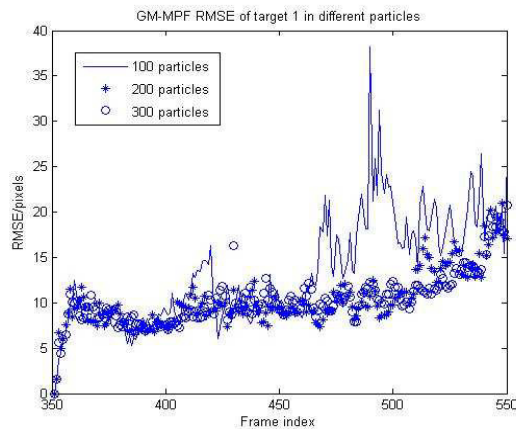
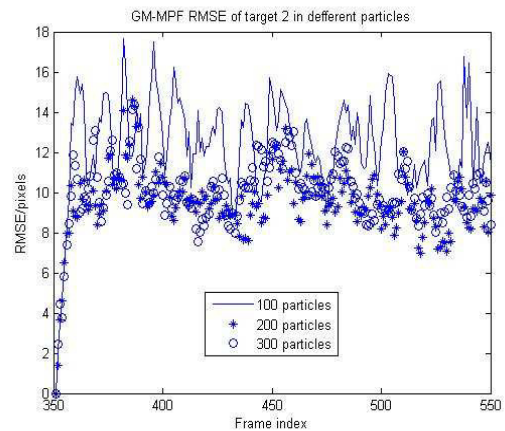Fig 5 RMSE of target 1 in different particles   Fig 6 RMSE of target 2 in different particles

Table 1．Processing time of 4 algorithms for 1000 particles

Algorithm	Average processing time every frame [ms]
PF	1.68 ± 0.046
GPF	1.52 ± 0.028
MPF	1.82 ± 0.034
GM-MPF	1.75 ± 0.019

6.Conclusions

Particle filter algorithm is poor at consistently maintaining the multi-modality problem and remains particle degeneracy phenomenon. This shortcoming can be addressed through using the component of Mixture Particle Filter (MPF) to express independent particle filter. Neff of improved particle filter and the mixture particle filter is used to describe the particle degradation. Gaussian Particle Filter (GPF) is used to reduce computational complexity of particle filter algorithm. The particle set is obtained by sampling of gaussian density function instead of re-sampling to improve the computing speed of particle filter algorithm and to obtain more accurate estimation performance. Combined with the advantages of GPF and MPF, GM-MPF algorithm is proposed. Meanwhile, comparing to the processing time of PF, GPF, MPF, GM-MPF, the experimental results are shown that the improved algorithm enhances the real-time and system estimate accuracy of the MPF during multi-targets tracking.

Acknowledgements

Reference

[1]   R.E. Kalman, A new approach to linear filtering and prediction problems, Transactions of the ASME—Journal of Basic Engineering, J. 1960, 82: 35–45.
[2]   Doucet, A, de Freitas, J. F. G., N. J. Gordon, editors: Sequential Monte Carlo Methods in Practice , M. Springer-Verlag, New York, 2001.
[3]   Doucet A, Godsill S, Andrieu C, On sequential monte carlo sampling methods for bayesian filtering, J. Statistics and Computing, 2000, 10(3): 197-208.
[4]   Comaniciu D, Ramesh V, Meer P, Kernel-based object tracking, J. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(5): 564-575.
[5]   Li Peihua, Moving object tracking method in image sequences, M. Science Press.
[6]   Kotecha J H, Djuric P M, Gaussian particle filtering, J. IEEE Transactions on Signal Processing, 2003, 51(10):

2592-2601.

[7]   Vermaak J, Doucet A, Perez P, Maintaining multi-modality mixture tracking, C. Proceedings of the Ninth IEEE International Conference on Computer Vision, Nice, 2003: 1110-1116.

[8]   Information on http://ftp.pets.rdg.ac.uk/PETS2001/.

[9]   Lin Qing, Xu Zhu, Wang Shi-Tong, Zhan Yong-Zhao, Moving Object Detection of Adaptive Gaussian Mixture on HSV, J. Computer Science, 2010, 37(10): 254-256.