Research Journal of Applied Sciences, Engineering and Technology 5(2): 427-432, 2013
DOI:10.19026/rjaset.5.4969
ISSN: 2040-7459; E-ISSN: 2040-7467

Submitted: May 04, 2012 Accepted: June 08, 2012 Published: January 11, 2013

**Research Article**

# Study on Multi-Target Tracking Based on Particle Filter Algorithm

[1,2]Junying Meng, [1]Jiaomin Liu, [1]Yongzheng Li and [1]Juan Wang
[1]College of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China
[2]Department of Computer Sciences, Shijiazhuang University, Shijiazhuang 050000, China

**Abstract:** Particle filter is a probability estimation method based on Bayesian framework and it has unique advantage to describe the target tracking non-linear and non-Gaussian. In this study, firstly, analyses the particle degeneracy and sample impoverishment in particle filter multi-target tracking algorithm and secondly, it applies Markov Chain Monte Carlo (MCMC) method to improve re-sampling process and enhance performance of particle filter algorithm.

**Keywords:** Important sampling, MCMC, multi-target tracking, particle filter, sequential

## INTRODUCTION

In recent years, many single-target video sequence tracking system are successfully developed one after another, but the multi-target tracking system is still a challenging project, especially the tracking of multiple targets of similar appearance and in complex motion. At present, the classic algorithms that widely applied are: Nearest Neighbor Filter (NNF), Joint Probability Data Association Filter (JPDAF) and Multiple Hypothesis Tracking (MHT), etc., NNF algorithm is proposed by Singer and it needs less computation and not relies on clutter distribution model, good for easy project implementation (Taek *et al*., 2005). However, it only uses the measurement in statistical sense that closest to the predicted position of tracked target as a candidate measurement and in reality, the measurement closest to the center of predicted location is not necessarily the correct objective measurement1. JPDA algorithm is proposed by Bar-Shalom, etc., which is regarded as one of most effective algorithms for solution of multi-target data association under intensive measurement and its tracking success rate is relatively high under all circumstances. With JPDAF algorithm, the search of associated solution is actually a seeking of combination issue and the computation amount of searching process is in exponential growth trend as quantity of targets and measurement grown, so it is difficult for such algorithm to be used widely in practical project (Mourad and Joris, 2002). Reid proposed MHT algorithm of multi-target tracking based on “All Neighbor” optimal filter proposed by Singer and concept of confirmation matrix proposed by Bar-Shalom. Such algorithm will attain a better result in tracking under high clutter density environment and be able to solve the problems of target appear and disappear in tracking period (Alexandre and Raphael, 2004). However, such algorithm will have computation amount rise rapidly as target number and observed quantity increased, so its application is limited in practical project. With particle filters putting into application, the problems existed in classical algorithms above said have been improved. Particle Filtering (PF) technique is a Optimal Regression Bayesian filtering algorithm based on MCMC simulation, which is not limited by linear error and high Gaussian noise hypothesis and is applicable for non-linear non-Gaussian model (Nummiaro *et al*., 2003). The basic idea is to take a series of weighted particles from current system state distribution to estimate and update the next system state. Arnaud and Nando (2001) have a research of the sequential monte carlo methods in practice. François *et al*. (2011) propose a direct, prediction- and smoothing-based kalman and particle filter algorithms.

This study introduces the particle filter such practical estimation problem-solving method into the field of vision tracking, constructing the tracking framework based on particle filter and combining with characteristics of targets at all levels, to manufacture trackers of good performance that have “multimodal” tracking features and be able to improve robustness. In specific implementation, re-sampling in MCMC method will be applied to solve particle degeneracy and sample impoverishment in particle filtering visual multi-target tracking algorithm. MCMC method can effetely enhance the performance of particle filter algorithm and reduce the computational complexity.

**Corresponding Author:** Junying Meng, College of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China

**Multi-target tracking technique and bayesian method:** To solve dynamic state system using Bayesian method, people hope to build state posterior probability density function based on all the information that could be obtained (including all measured value). Since it embodies all available information, this function can be considered is the complete solution of this estimation problem. In principle, the optimal estimation of the state can be got from probability density function, when get a measured value, can also get a estimate. Therefore, Bayesian recursive filter is a very good choice. When a new measured value comes, we deal with it in order and don't have to save all of the data. So, the filter consists of two steps, Prediction and fix:

- Prediction stage is to predict the probability density function at current time by system model.
- Fix stage is to update the probability density function of the state after getting the measured value at current moment.

Assume the state space model of dynamic system is:

$$x_k = f_k(x_{k-1}, v_k) \tag{1}$$

$$z_k = h_k(x_k, u_k) \tag{2}$$

In which, $x_k$ indicates the state of system at time k and $z_k$ indicates the measurement vector at time k, as shown in Fig. 1. $f_k : R^{n_x} \times R^{n_v} \to R^{n_x}$ and $h_k : R^{n_x} \times R^{n_u} \to R^{n_z}$, respectively indicates Status transfer function and measurement function, $v_k$, $u_k$ respectively indicates the process noise and measurement noise.

Bayesian filter is a recursive method base on probability density. From the analyses of the framework, the Move Status of object can be modeling use the Markov procedure. Get formula as bellow:

$$p(x_k | x_{1:k-1}) = p(x_k | x_{k-1}) \tag{3}$$

In which, $x_k | x_{1:k-1} = (x_1, x_2, \cdots x_{k-1})$. From formula 3, it can see, $p(x_k | x_{1:k-1})$ can change into $p(x_k | x_{k-1})$ by recursive method. That is, the value of current process state depends on the value of last state and all the other previous value. Then, the dynamic model of the state can be described by the formula below:

$$x_k = f_k(x_{k-1}, w_{k-1}) \tag{4}$$

In this formula, $x_k$ indicates the process state at time $t_k$, $f_k$ is the function Mapping of last state $x_{k-1}$ to current state $x_k$, $w_{k-1}$ is the process noise. According the structure in Fig. 1, We can achieve the measurement model by combine the system state and the measure value of tracking system:

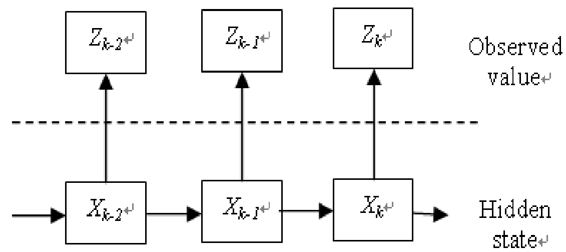


Fig. 1: Dynamic state space model

$$z_k = h_k(x_k, v_k) \tag{5}$$

$z_k$ indicates the process state at time $t_k$, $h_k$ is the function Mapping of system state $x_k$ to measurement state $z_k$, $v_k$ is the measurement noise.

In the process of filter, the main purpose is through the primitive state value $x_k$ at time $t_k$ and the measured value set $z_{1:k}$ to get $\hat{x}_k$, the estimate value of state. Can use the following formula to describe this process:

$$p(x_k | z_{1:k}) = c_k \, p(z_k | x_k) p(x_k | z_{1:k-1}) \tag{6}$$

$$p(x_k | z_{1:k-1}) = \int p(x_k | x_{k-1}) p(x_{k-1} | z_{k-1}) dx_{k-1} \tag{7}$$

In which, normalization can coefficient $c_k$ can be defined as:

$$1/c_k = p(z_k | z_{1:k-1}) = \int p(x_k | z_{1:k-1}) p(z_k | x_k) dx_k \tag{8}$$

$p(x_k | z_{1:k})$ indicates the posteriori probability, $p(x_k | z_{1:k-1})$ is prior probability, $p(x_k | z_k)$ is the likelihood degrees, $p(x_k | x_{k-1})$ is the transition probability. Likelihood degrees and transition probability can be deduced by formula 6 and 7. Then, the posteriori probability can be got through the recursive formula 7, 8 and 9 of Bayesian filter algorithm.

For multi-target tracking system, assume the target number is M, $i$ is one of them. Assume all targets movement is obey the first order Marko chain and the movement of these targets is independent. Then realizing multi-target tracking by estimate on vector $X_t = (X_t^1, \cdots, X_t^M)$. At time t, the formula of system state can be give as bellow:

$$X_t^i = F_t^i\left(X_{t-1}^i, V_t^i\right) \forall i = 1, \cdots, M \quad (9)$$

**Particle filter multi-target tracking technique:**

**Particle filter theory:** Particle filter is a probability estimation method based on Bayesian framework and it is very suitable to describe the target tracking uncertainty. Particle filter approach provides a flexible framework and many traditional vision tracking methods can make a great robustness enhancement through slight model modification and embedment into the particle filter frame work alone. Moreover, particle filter approach has unique advantage in handling non-linear non- Gaussian multi-modal cases (Xinguang *et al*., 2010). The essence is to realize Bayesian filter in non-parametric Monte Carlo simulation method.

Particle filter approach itself is able to express a number of assumptions in particle sets, so it can be used to solve multi-target tracking problem. Due to data association is only considered in a given period of time, the complex of data association is thus reduced. Using hybrid bootstrap filter to solve the data association problem, in which each particle involves single target state information and expresses one target state hypothesis; using Gaussian mixer model to express posterior distribution of all targets under the given observation conditions and each model of posterior distribution corresponds to a target (Milstein *et al*., 2002).

The core idea of particle filter algorithm is to use weighting of a series of random samples and posterior probability density required by expression, to get the estimated state value. When the sample number is very large, such probability estimation will be equal to posterior probability density. Assume Ns indicate the particle number, then $\{X_k^i, i = 1,..., N_s\}$ means a support point set and its corresponding weight is $\{w_k^i, i = 1,..., N_s\}$ and normalized weight is $\sum_{i=1}^{N_s} w_k^i = 1$, then $\{X_k^i, w_k^i\}_{i=1}^{N_s}$ indicates the random particle set describing posterior density. Thereupon, posterior probability density at the time k can use discrete weight sum that is approximate to:

$$p\left(X_k \middle| Z_{1:k}\right) \approx \sum_{i=1}^{N_s} w_k^i \delta\left(X_k - X_k^i\right) \quad (10)$$

In which, the weight $w_k^i$ can be sampled and selected from important density function $q\left(X_k \middle| Z_{1:k}\right)$ in sequential important sampling method.

If the sample $X_k^i$ can be obtained from important density $q\left(X_k \middle| Z_{1:k}\right)$, then the weight of the i'th particle can be defined as:

$$w_k^i \propto \frac{p\left(X_k^i \middle| Z_{1:k}\right)}{q\left(X_k^i \middle| Z_{1:k}\right)} \quad (11)$$

If the important density function can be decomposed as follows:

$$q\left(X_k \middle| Z_{1:k}\right) = q\left(X_k \middle| X_{k-1}, Z_{1:k}\right) q\left(X_{k-1} \middle| Z_{1:k-1}\right)$$

Then the posterior probability density can be expressed as:

$$\begin{aligned} p\left(X_k \middle| Z_{1:k}\right) &= \frac{p\left(Z_k \middle| X_k\right) p\left(X_k \middle| Z_{1:k-1}\right)}{p\left(Z_k \middle| Z_{1:k-1}\right)} \\ &= \frac{p\left(Z_k \middle| X_k\right) p\left(X_k \middle| X_{k-1}, Z_{1:k-1}\right) p\left(X_{k-1} \middle| Z_{1:k-1}\right)}{p\left(Z_k \middle| Z_{1:k-1}\right)} \\ &= \frac{p\left(Z_k \middle| X_k\right) p\left(X_k \middle| X_{k-1}\right)}{p\left(Z_k \middle| Z_{1:k-1}\right)} p\left(X_{k-1} \middle| Z_{1:k-1}\right) \\ &\propto p\left(Z_k \middle| X_k\right) p\left(X_k \middle| X_{k-1}\right) p\left(X_{k-1} \middle| Z_{1:k-1}\right) \end{aligned} \quad (12)$$

And updated formula for weights is obtained therefore:

$$\begin{aligned} w_k^i &\propto \frac{p\left(Z_k \middle| X_k^i\right) p\left(X_k^i \middle| X_{k-1}^i\right) p\left(X_{k-1}^i \middle| Z_{1:k-1}\right)}{q\left(X_k^i \middle| X_{k-1}^i, Z_{1:k}\right) q\left(X_{k-1}^i \middle| Z_{1:k-1}\right)} \\ &= w_{k-1}^i \frac{p\left(Z_k \middle| X_k^i\right) p\left(X_k^i \middle| X_{k-1}^i\right)}{q\left(X_k^i \middle| X_{k-1}^i, Z_{1:k}\right)} \end{aligned} \quad (13)$$

Weights can be normalized as:

$$\tilde{w}_k^i = \frac{w_k^i}{\sum_{i=1}^{N_s} w_k^i} \quad (14)$$

If $q\left(X_k \middle| X_{k-1}, Z_{1:k}\right) = q\left(X_k \middle| X_{k-1}, Z_k\right)$ is achieved, namely the important density function only depends on $X_{k-1}$ and $Z_k$, then only storage sample $X_k^i$ but not $X_{k-1}^i$ and the past observation $Z_{1:k-1}$ is needed, therefore computation storage can be greatly reduced. At this time the weight is revised as:

$$w_k^i \propto w_{k-1}^i \frac{p\left(Z_k \middle| X_k^i\right) p\left(X_k^i \middle| X_{k-1}^i\right)}{q\left(X_k^i \middle| X_{k-1}^i, Z_k\right)} \quad (15)$$

Thus, the posterior probability density at the time K can use discrete weight sum that approximate to:

$$p\left(X_k \middle| Z_{1:k}\right) \approx \sum_{i=1}^{N_s} \tilde{w}_k^i \delta\left(X_k - X_k^i\right) \tag{16}$$

For multi-target tracking system, N (quantity) particles are involved in initial particle set:

$$S_0 = \left(s_0^n, \frac{1}{N}\right)_{n=1,\dots,N} \tag{17}$$

In which each element $s_0^{n,i}$ from $i = 1,\dots,M$ is obtained from independent $p\left(X_0^i\right)$ sampling. The particle set at the time t-1 is assumed as $S_{t-1} = \left(s_{t-1}^n, p_{t-1}^n\right)_{n=1,\dots,N}$, in which $\sum_{n=1}^{N} p_{t-1}^n = 1$. Each particle is a vector of dimension $\sum_{i=1}^{M} n_x^i$ and $s_{t-i}^{n,i}$ represents the i'th element in $s_{t-1}^n$ and $n_x^i$ represents the state vector dimension of the i'th target.

Each iteration in particle filter algorithm is divided into two steps: prediction and weight updating. Prediction means sampling from proposed density function $F_t^i$ and proposed density function is consistent with the target motion model; weight updating is to make the weight at the time t-1 multiplied by the observation likelihood:

$$\tilde{s}_t^n = \begin{pmatrix} F_t^i\left(s_{t-1}^{n,1}, v_t^{n,M}\right) \\ \vdots \\ F_t^M\left(s_{t-1}^{n,M}, v_t^{n,M}\right) \end{pmatrix} \tag{18}$$

For the likelihood calculation of the nth particle, the observed value $\tilde{s}_t^n, n = 1,\dots,N$ can be expressed as:

$$p\left(Z_t = \left(z_t^1,\dots,z_t^{m_t}\right) \middle| X_t = \tilde{s}_t^n\right) = \prod_{j=1}^{m_t} p\left(z_t^j \middle| \tilde{s}_t^n\right) \propto \prod_{j=1}^{m_t}\left[\frac{q_t^0}{V} + \sum_{i=1}^{M} l_t^i\left(z_t^j; \tilde{s}_t^{n,i}\right) q_t^i\right] \tag{19}$$

In which, $l_t^i\left(z_t^j; \tilde{s}_t^{n,i}\right) = p\left(z_t^j \middle| K_t^j = i, X_t^i = \tilde{s}_t^{n,i}\right)$, $q_t^i, i = 1,\dots,M^t$ means the probability of the j'th observed value from the i'th target and $M^t$ means the quantity of target at the time t.

**Re-sampling:** The basic problem to be solved in sequential importance sampling algorithm is particle degeneracy, after a few or multiple recursion, the weights of most particles become very small and only a few particles have a relatively large weights.

Re-sampling technique is used herein to solve particle degeneracy, namely removing the particles of small weight and reproducing those of large weights. Detailed process is as follows.

After systematic observation, the first step is to recalculate and confirm the weight ranges of the particles. The realistic particles will be granted relatively large weights and those deviating from reality will be given relatively small ones. The second step is re-sampling process, in which the particles of large weights will derive much more "offspring" particles and those of small weights will correspondingly derive less ones, moreover, the weights of "offspring" particles will be re-set. The third step is system state transition process, in which the state of each particle at the time t will be predicted through adding a random amount of particles. The forth step is system observation process at time t, similar to the first step, the final representation of target state will be obtained through weighting of a numbers of particles.

These new particles propagated into the calculate of next frame, Then the dynamic model change the position of particles and the observation model change the weight of particles, determine the target position. Re-sampling cycling constantly, this process is shown in Fig. 2.

Particle filter re-sampling inhibits the weight degeneracy, but also introduces other problems. At first, the particles are no longer independent, reducing the opportunity of parallel computing because of continuous re-definition of new particle set; second, the particles of relatively large weights will be chosen for many time, weakening the particle diversity and the sample particles contain many duplicate points, when the system noise is small, they said will be obvious and after several iteration, all particles will converge to a point and this is known as particle depletion.

Particle depletion resulted from re-sampling process makes the number of particles expressing PDF state too small and therefore inadequate, while unlimited increase of particle number is not realistic.

**MCMC method:** Markov Chain Monte Carlo (MCMC) method is introduced to generate samples from target distribution through constructing Markov chain, which has a good convergence effect (Taek *et al*., 2001).

In each process of iteration of sequential important sampling, the particles can move to different places by combining with MCMC, so that particle depletion is avoided and furthermore, Markov chain can push the particles to the places closer to PDF state and make the sample distribution more reasonable. There are many MCMC methods put into application and Metropolis Hasting method is adopted herein.

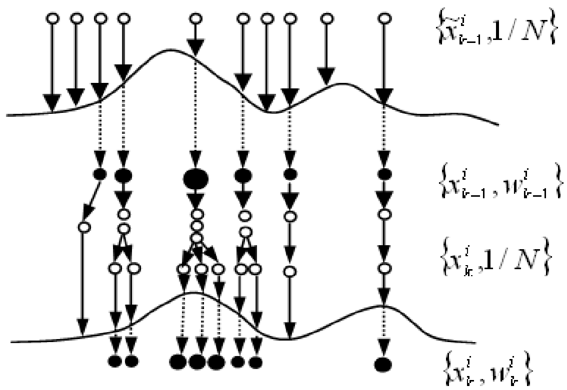


Fig. 2: The main process of re-sampling

Specific re-sampling process is as follows:

- According to the samples uniformly distributed in the range (0, 1), thresholds u-U (0, 1) are obtained.
- Sampling $x_t^{*(i)}$ as per distribution probability $p(x_t \mid x_{t-1}^{(i)})$, i.e., $x_t^{*(i)} \sim p(x_t \mid x_{t-1}^{(i)})$.
- Accept $x_t^{*(i)}$, if $u < \min[1, \frac{p(y_t \mid x_t^{*(i)})}{p(y_t \mid \tilde{x}_t^{(i)})}]$; otherwise, drop $x_t^{*(i)}$, make $x_t^{(i)} = \tilde{x}_t^{(i)}$.

**Template updating:** Selection of target template is an important part of visual tracking algorithm and a good target template shall be distinctive and unique to ensure the tracking accuracy and effectiveness. In motion process, the targets will be changed due to effects of its motion, light and perspective and only appropriately and reasonably updating of target template can overcome to some extent the impaction of such changes on tracking effect. Reasonable update strategy shall be able to adapt to slow changes of target characteristics, but also rapid changes.

Template is generally divided into stationary template and dynamic template. Stationary template is often applied because of sample and reliable. However, characteristics of moving target will be changed over time, when the change of moving target state leads to corresponding change of its characteristic, it requires the algorithm to take appropriate strategy to response and obviously the stationary template cannot satisfy such requirement.

Dynamic template is a resolution responding to the requirements above said. The simplest update rule for dynamic template is update frame by frame, which abandons all previous template information and adopts the best matched sub-region image of previous time as current target template. However, due to image shading, light change, deformation or accumulation of matching error, the dynamic template will easily lead to target tracking drift and even lost.

Dynamic template can be expressed as a forgetting process as follows:

$$M_{updated} = \alpha \cdot M_{fixed} + (1-\alpha) M_{new} \tag{20}$$

In which, $\alpha$ indicates retention of stationary template that can takes empirical value. Coefficient Bhattacharyya indicating target similarity is adopted herein as a parameter, compared with empirical value, it is more in line with update requirement. $M_{fixed}$ indicates dynamic template and generally it is target weighted color histogram in initial position. $M_{updated}$ indicates new template and generally it is target weighted color histogram in estimated position. By using the dynamic template update rule above said, the target weighted color histogram model contains the target color information of initial time and current time, but also makes real-time adjustment of update rate according to target similarity in estimated position, which can effectively inhibit tracking errors from accumulating and tracking target from drifting.

## CONCLUSION

Target tracking is usually non-linear and non-Gaussian and the targets usually do voluntary movement, so their movement cannot be accurately described in mathematical equations. The study introduces the particle filter such practical estimation problem-solving method into the field of vision tracking, constructing the tracking framework based on particle filter and combining with characteristics of targets at all levels, to manufacture trackers of good performance that have “multimodal” tracking features and be able to improve robustness.

In specific implementation, re-sampling in MCMC method will be applied to solve particle degeneracy and sample impoverishment in particle filtering visual multi-target tracking algorithm. MCMC method can effetely enhance the performance of particle filter algorithm and reduce the computational complexity.


## ACKNOWLEDGMENT

This research was supported by the Natural Science Foundation of Hebei Province under Grant No. F2012208004. The authors would like to thank the anonymous reviewers for their valuable remarks and comments.

**REFERENCES**

Alexandre, H. and L. Raphael, 2004. Malignant Hypertension (MHT): A systematic therapeutic approach through blockade of renin angiotensin system. Am. J. Hypertens., 17(5): 155-162.

Arnaud, D. and D.F. Nando, 2001. Sequential Monte Carlo Methods in Practice. The Springer Press, New York, pp: 82-86.

François, D., P. Yohan and B. Ait-El-Fquih, 2011. Direct, prediction-and smoothing-based Kalman and particle filter algorithms. Signal Process., 91(8): 2064-2077.

Milstein, A., J.N. Sanchez and E.T. Williamson, 2002. Robust global localization using clustered particle filtering. Proceedings of the National Conference on Artificial Intelligence, Edmonton, pp: 581-586.

Mourad, O. and D.S. Joris, 2002. Hybrid fuzzy probabilistic data association filter and joint probabilistic data association filter. Inform. Sci., 142: 195-226.

Nummiaro, K., E. Koller-Meier and L. Van Gool, 2003. An adaptive color-based particle filier. Image Vision Comput., 21(1): 99-110.

Taek, L.S., G.L. Dong and R. Jonha, 2005. A probabilistic nearest neighbor filter algorithm for tracking in a clutter environment. Signal Process., 8(10): 2044-2053.

Xinguang, S., H. Biao and M.L. Jong, 2010. Constrained Bayesian state estimation-A comparative study and a new particle filter based approach. J. Process. Cont., 20(2): 143-157.