International Journal of
Soft Computing And Software Engineering (JSCSE)
ISSN: 2251-7545
Vol.2,No.2, 2012
DOI: 10.7321/jscse.v2.n2.2

An Objects Detecting and Tracking method based on MSPF and SVM

[1]Wei Sun, [*2]Xu Zhang, [3]Yunyi Yan
[1, 2] *School of Mechano-electronic Engineering, Xidian University, 710071, Xi'an, China*
[3]*School of Electro-mechanical Engineering Xidian University, 710071, Xi'an, China*

[1]*sunweitom@tom.com,* [*2] *xuzhang@mail.xidian.edu.cn, yyyan@xidian.edu.cn*

Abstract. *Considering that the robust real-time tracking of non-rigid objects is difficult to realize, We present an objects detecting and tracking method based on mean-shift particle filter (MSPF) and support vector machine (SVM). The proposed algorithm uses the mean-shift vector of the tracking object to update the state transition matrix of particle filter algorithm, and we define the criterion of the particle degradation, to improve the conditions of degradation, the particles will be re-distributed as Gaussian distribution. Because of the real-time update of the particle motion parameters, the prediction accuracy of target motion parameters is improved. Under the condition of target conflicting and partially covering, the proposed algorithm is still tracking effectively. Apply SVM to relevance feedback of object detecting and tracking, the experiments results show that the method can overcome the shortness of the traditional methods, effectively improve the tracking speed and precision. The results of the experiment indicate that the average processing time per frame of the proposed algorithm is reduces by about 21% comparing with the classical ones, while the efficiency of particle increases by about 32%.*

Keywords: *Image process, Object tracking, Particle Filter, Mean shift, SVM*

1. Introduction

Object tracking has been an active research area in the vision community in recent years. It has many potential applications in the fields of intelligent robots, monitoring and surveillance, human computer interfaces, smart rooms, vehicle tracking, biomedical image analysis, video compression, etc. Numerous approaches have been proposed to track moving objects in image sequences. Among these different algorithms, the particle filter algorithm has been successfully applied to track moving objects in cluttered environments.

Monte Carlo particle filter method based on recursive Bayesian estimation, which can be effectively applied to nonlinear, non-Gaussian movement system. The state transition model characterizes the target motion characteristics between two frames in target tracking. It is very difficult to establish an accurate model of the target according to the random movement. Because of the random sample assumptions in PF, some tracking algorithms of PF dependent on the accuracy of the system state transition model, The error of particle state transition will lead to particle degradation and inaccurate tracking results, According to our research, it is difficult to obtain the real state of target which moved rapidly, but the researchers rarely discuss the relevant issues [1-5].

In order to obtain more accurate characteristics of the target attribute, MSPF used to obtain the initial frame in the best elliptical description of the target. Then, obtaining the offset vector by mean-shift algorithm; injecting it into the system state transition model, and fitting particles to Gaussian distribution. Finally, updating the kernel function parameters of the subsequent frames according to filter result, determining multi-objective trajectory by the data associated, and SVM can make object tracking to be correct.

This paper is organized as follows: in section 2, the classic particle filter algorithms and the problem will be solved are explained. The proposed algorithm is given in section 3. Section 5 contains the experimental results followed by a short discussion. Finally, the conclusions are highlighted in section 7.

2. Theory of Particle Filter

The basic idea of the particle filter is that the posterior density which is approximated by a set of discrete samples (called particles) with associated weights [6-8]. The working mechanism of particle filters is following: the state space is partitioned as many parts, in which the particles are filled according to some probability measure. The higher probability, the denser the particles are concentrated.

The particle filter consists of essentially two steps: prediction and update. The prediction stage uses the probabilistic system transition model $p(x_k \mid x_{k-1})$ to predict the posterior at time k as

$$p(x_k \mid Z_{k-1}) = \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid Z_{k-1}) dx_{k-1} \tag{1}$$

At time k, the observation Z_k is available, the state can be updated using Bayes' rule

$$p(x_k \mid Z_k) \propto p(z_k \mid x_k) p(x_k \mid Z_{k-1}) \tag{2}$$

In the particle filter, the posterior $p(x_k \mid Z_k)$ is approximated by a finite set of N samples with importance weights. The candidate samples are drawn from an importance distribution and the weight of the samples are

$$W_k^{(i)} \propto W_{k-1}^{(i)} \frac{p(z_k \mid x_k^{(i)}) p(x_k^{(i)} \mid x_{k-1}^{(i)})}{q_p(x_k \mid x_{k-1}^{(i)}, z_k)} \tag{3}$$

To ensure that $\sum_{i=1}^{N} w_k^{(i)} = 1$, then the Monte Carlo estimate of $f(x)$ is given by

$$\hat{f}(x) = \frac{\sum_{i=1}^{N_p} W_n(x_n^{(i)}) f(x_n^{(i)})}{\sum_{i=1}^{N_p} W_n(x_n^{(i)})} = \sum_{i=1}^{N_p} \hat{W}(x^{(i)}) f(x^{(i)}) \tag{4}$$

Monte Carlo importance sampling is to use a number of N_p independent samples drawn from $q(x)$ to obtain a weighted sum to approximate; we can see that the particle filter converges to the true posterior of global extreme point.

Since the pdf can be approximated by the point-mass histogram, by random sampling of the state space, we get a number of particles representing the evolving pdf. However, since the posterior density model is unknown or hard to sample, we would rather choose another distribution for the sake of efficient sampling. Hence, after some iterations, only few particles have non-zero importance weights, the phenomenon is often called weight degeneracy or sample impoverishment. In this paper, new algorithm will be proposed to solve the problem of sample impoverishment.

3. Proposed MSPF object tracking algorithm

Commonly we use simple transfer model such as first or second order autoregressive model, we can use first-order mode when the target just moved translational. For example, first-order autoregressive model, system state transition model of moving target is $S_t = As_{t-1} + w_{t-1}$ .in the video sequence, the transfer matrix indicate moving state of the target. Since movement of targets is complex, so the transfer matrix A is time varying. The particle filters uses in the papers to select a first-order autoregressive model $X_t = A'X_{t-1} + w'_{t-1}$. Generally, the transfer matrix is fixed, and the fixed value will lead to error of the particles transfer model and inaccurate results of tracking trajectory [9, 10].

In this paper, according to the color-based particle filter object tracking algorithm, we update the target model during slowly changing image observations to overcome the appearance changes. We use the mean-shift process to calculate the mean-shift vector of the target. And introduce transfer matrix A' into the particle filter algorithm, so the updated parameters is consistent to the state of the particle

International Journal of
Soft Computing And Software Engineering (JSCSE)
ISSN: 2251-7545
Vol.2,No.2, 2012
DOI: 10.7321/jscse.v2.n2.2

motion. Experiments shows that the dynamic transfer matrix A' can effectively improve performance of the PF tracking algorithm.

3.1 Color probability distributions

We want to apply a particle filter in a color-based context. Color distributions are used as target models as they achieve robustness against non-rigidity, rotation and partial occlusion. Suppose that the distributions are discredited into m-bins. The histograms are produced with the function, which assigns the color to the corresponding bin. In our experiments, the histograms are typically calculated in the RGB space using 8_8_8 bins. To make the algorithm less sensitive to lighting conditions, the HSV color space could be used instead with less sensitivity to V (e.g. 8_8_4 bins).

In a tracking approach, we need a similarity measure which is based on color distributions. A popular measure between two distributions $p(u)$ and $q(u)$ is the Bhattacharyya coefficient.

$$\rho(p,q)=\int\sqrt{p_u q_u} \tag{5}$$

Considering discrete densities such as our color histograms

$$\rho(p,q)=\sum_{u=1}^{k}\sqrt{p_u q_u} \tag{6}$$

The larger $\rho(p,q)$ is, the more similar the distributions are. For two identical normalized histograms we obtain $\rho(p,q)=1$, indicating a perfect match. As distance between two distributions we denote the measure

$$d=\sqrt{1-\rho(p,q)} \tag{7}$$

This called Bhattacharyya distance. The small Bhattacharyya distances correspond to large weights:

$$w^{(n)}=\Lambda e^{-d}, \Lambda$$

Particle filter tracking algorithm is usually used as a Epanechnikov kernel function. When the target zone is near the round or square shape, the tracking result can be more accurately. However, when the shape of target is rectangular area, it will take part of the background region as a target, resulting in inaccurate tracking results, and even failure. In this paper, we choose Elliptical particles. Location of the color distribution of target：

$$p_y=\{p_y^{(u)}\}_{u=1,2,\ldots,m}$$

$$p_y^{(u)}=f\sum_{i=1}^{I}k\left(\frac{\|y-x_i\|}{a}\right)\delta\left[h(x_i)-u\right], \quad \sum_{u=1}^{m}p_y^{(u)}=1, \tag{8}$$

Parameters $a=\sqrt{H_x^2+H_y^2}$ is used to determine the shape of particles, f is Normalization factor. The size of kernel function plays a very important role in Particle filter tracking algorithm, because it not only reflects the tracking window size, but also determines the number of samples of Monte Carlo iterations, window size of the kernel function is usually the initial tracking window and it no longer changed in the tracking process. However, there are significant changes in the target scale; the fixed window width can often lead to tracking failure. So we update the ellipse parameters based on the filter results of following frame.

3.2 Mean Shift vector of the target

Mean shift is a non-parametric feature-space analysis technique, a so-called mode seeking algorithm, a nonparametric estimator of density gradient. Given n data points, mean shift vector always points toward the direction of the maximum increase in the density. The mean shift procedure guarantees to converge to a point where the gradient of density function is zero. Iteration of mean shift gives rise to

International Journal of
Soft Computing And Software Engineering (JSCSE)
ISSN: 2251-7545
Vol.2,No.2, 2012 DOI: **10.7321/jscse.v2.n2.2**

natural clustering algorithms. Application domains include clustering in computer vision and image processing[11-15].

We start with an initial estimate x. Let a kernel function $K\left(x_i - x\right)$ be given. This function determines the weight of nearby points for re-estimation of the mean. Typically we use the Gaussian kernel on the distance to the current estimate, $K\left(x_i - x\right) = e^{c\|x_i - x\|^2}$ .The weighted mean of the density in the window determined by K is

$$M_{h,G}(x) = \frac{\sum_{i=1}^{n} x_i K(\left\|\frac{x - x_i}{h}\right\|^2)}{\sum_{i=1}^{n} K(\left\|\frac{x - x_i}{h}\right\|^2)} - x \tag{9}$$

where $N(x)$ is the neighborhood of x , a set of points for which $K(x \neq 0)$ .The mean-shift algorithm now sets $0 \leftarrow M_{h,G}(x)$, and repeats the estimation until $M_{h,G}(x)$ converges.

3.3 Update of the particles movement model

When tracking a color object, particle filter operates on color probability distribution of the image, which derived from color histograms. The particle filter calculates the centroid of the 2D color probability distribution within the 2D window, re-centers the window, and then calculates the position for the next window. Thus, it is no necessary to calculate the color probability distribution over the whole image, and calculation of the probability distribution will be restricted to a smaller image region surrounding the current particle area. Using his feedback of calculation region size, it tends to result in large computational savings when flesh color does not dominate the image.

The proposed MSPF algorithm employs the Bhattacharyya distance to update the a priori distribution calculated by the particle filter. Each sample of the distribution represents an ellipse and is given as $X = \{x, \dot{x}, y, \dot{y}, H_x, H_y, \theta\}$, where x, y specify the location of the ellipse, $\hat{x}, \hat{y}$ specify the motion, H_x, H_y specify the length of the half axes and θ the corresponding scale change.

The sample set is propagated through the application of a dynamic model $S_t = A's_{t-1} + w_{t-1}$, where A defines the deterministic component of the model and w_{t-1} is a multivariate Gaussian random variable, expanding this model to second order is straightforward. According to the movement model of the object and the result of mean shift vector, we will change the position of the particles. In our application, we currently use a first order model to describe a region moving with constant velocity $\hat{x}, \hat{y}$ and scale change θ. We define that $m_x = \left(M_{h,G}\right)_x$, $m_y = \left(M_{h,G}\right)_y$, so

$$A' = \begin{bmatrix} 1 & m_x & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & m_y & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

The dynamic model of the particles is:

$$X_k = A_k' \cdot X_{k-1} + w_{k-1} \tag{11}$$

So, unlike the mean shift algorithm, which is designed for static distributions, MSPF is designed for dynamically changing distributions. These occur when objects in video sequences are being tracked and the object moves so that the size and location of the probability distribution changes in time. The MSPF algorithm adjusts the search window size in the course of its operation. Initialized window size can be set at any reasonable value. Instead of a set or externally adapted window size, MSPF relies on the moment model information, extracted as part of the internal workings of the algorithm, to continuously adapt its window size and position within or over each video frame. Thus, window radius, or height and width, is set to a function of the zero moment found during search. Considering the whole sample sets, the tracker handles multiple hypotheses simultaneously.

3.4 Proposed MSPF tracking Algorithm

Provided that we draw a sequence of random samples $\left\{x_{n-1}^{(i)}, w_{n-1}^{(i)}\right\}_{i=1}^{N}$ from probability distribution $p(x)$, number of active objects is defined as $t = 1,2,...,\tau$. The samples are resample to generate an unweighted particle set according to their importance weights to avoid degeneracy. Thus the importance weights $W_n^{(i)}$ can be updated recursively, for which the weights are given by

$$W_n^{(i)} = \max_{t=1}^{\tau} W_{t,n-1}^{(i)} \frac{p(y_{t,n} \mid x_n^{(i)}) p(x_n^{(i)} \mid x_{n-1}^{(i)})}{q(x_n^{(i)} \mid x_{0:n-1}^{(i)}, y_{t,n})} \tag{12}$$

Now we redefine the mean shift algorithm based on our generalizations summarized in the introduction. The complete adaptive implementation for the proposed object tracking algorithm is presented below.

1) Initialization of the particle filter, find the initial starting values x, y and H_x, H_y.

2) Choose the initial location of the 2D mean shift search window. Calculate the color probability distribution in the 2D region centering at the search window which locates in an area slightly larger than the mean shift window size.

3) According to the mean-shift vector $M_{h,G}(x) = \hat{y}_{k-1} - \hat{y}_k$, update the matrix A' using $m_x = \left(M_{h,G}\right)_x$, $m_y = \left(M_{h,G}\right)_y$.

4) Calculate the particle of t using dynamical model $X_k = A'_k \cdot X_{k-1} + w_{k-1}$.

5) Estimate the mean state of the particle set, then jump to step 2 after updating particles weights and the multinomial sampling-importance resampling.

4. Object conformation based on SVM

The support vector machine (SVM) is a widely used tool in classification problems. The SVM trains a classifier by solving an optimization problem to decide which instances of the training data set are support vectors, which are the necessarily informative instances to form the SVM classifier. Suppose that there are m instances of training data. Each instance consists of a (x_i, y_i) pair where $x_i \in IR^{IN}$ is a vector containing attributes of the ith instance, and $y_i \in \{+1, -1\}$ is the class label for the instance. To classify two classes of data, we need solve the quadratic programming optimization problem of equation (11), which is subject to equation (14):

$$\arg\min_{w,b,\xi} \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{m} \xi_i \tag{13}$$

$$\begin{gathered} yi(f(x_i)) \geq 1 - \xi_i, \xi_i \geq 0, for\ i = 1,...,m \\ f(x_i) = w \cdot x_i + b \end{gathered} \tag{14}$$

International Journal of
Soft Computing And Software Engineering (JSCSE)
ISSN: 2251-7545
Vol.2,No.2, 2012
DOI: **10.7321/jscse.v2.n2.2**

In the functions above, minimizing $\| w \|^2 / 2$ corresponds to maximizing the margin between $w \cdot x + b = 1$ and $w \cdot x + b = -1$. We suppose the true target to 1, and false target to -1. Using the optimal hyper plane of equation (16), we can get optimization results.

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i^* y_i K(x, x_i) + b^*\right) \tag{15}$$

Using the SVM, we can effectively confirm that the tracked object is correct, and then eliminate the wrong target to improve the precision of detection.

5. Experimental Results

The proposed tracking method adds an adaptive transfer model based on color distributions to particle filtering. The color-based tracker can efficiently and successfully handle non-rigid and fast moving objects under different appearance changes. Moreover, as multiple hypotheses are processed, objects can be tracked well in cases of occlusion or clutter. The proposed algorithm runs comfortably in real time with 10 to 30 frames per second without any special optimization on a normal 2.4GHz PC. In Fig.1, The result has shown that tracking of ONECUP video sequence at frame 4, 35, 60 using MSPF is obviously accuracy and better than only using mean shift.

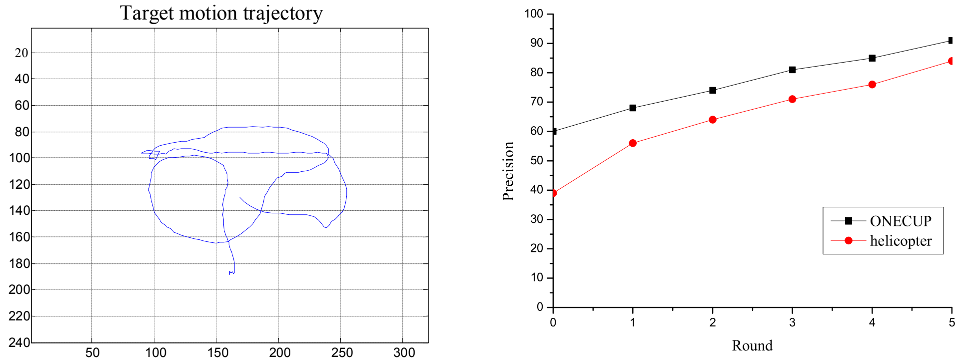
(a) Tracking result of ONECUP video sequence using MSPF

（(b) Tracking result of ONECUP video sequence using particle filter

Fig. 1 Comparison of MSPF and the particle filter

Fig.2 shows the analysis parameters of tracking results of ONECUP sequence based on SVM, in which (a) is the depiction of the target motion trajectory, and (b) is the capability comparison curves. We can see from (b) that the precision of detecting and tracking is improved after 5 rounds feedback.

International Journal of
Soft Computing And Software Engineering (JSCSE)
ISSN: 2251-7545
Vol.2,No.2, 2012
DOI: **10.7321/jscse.v2.n2.2**



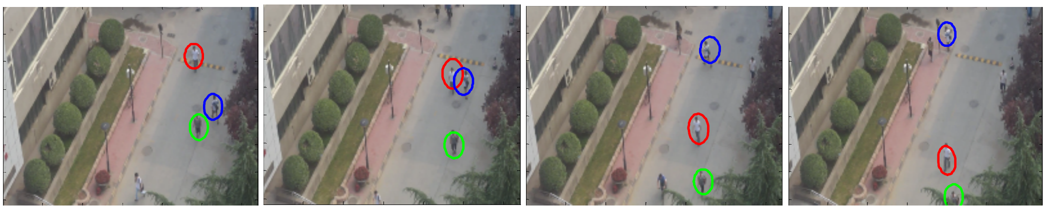
(a) Trajectory of MSPF tracking algorithm (b) Capability comparison curves

Fig. 2 The analysis parameters of different tracking results of ONECUP sequence

In order to exam the validity of proposed method, more experiments are implemented. Fig.3 shows the tracking result of sea surveillance sequence at different frame using the proposed MSPF algorithm. Fig.4 shows the tracking result of outdoor surveillance sequence at different frame using MSPF. We can see that when the object overlap or fusion in the background scenery, MSPF can still achieve effective detecting and tracking.

(a)3# (b)282# (c)636# (d)720#

Fig.3 Sea surveillance video tracking results



(a)3# (b)42# (c)96# (d)160#

Fig.4 Outdoor surveillance video tracking results

6. Conclusion

In summary, method of mean-shift particle filter and SVM are effectively combined in this paper. The proposed algorithm has the ability to improve the detecting and tracking capabilities by user participation. Experiments show the proposed method has some advantages: 1. Color and shape features of the target are take into account to weight histogram, which can overcome the resulting appearance changes. 2. The number of bins in the histogram is optimized with respect to the noise of the camera, and one different similarity measurement takes account into neighboring bins. 3. The adaptive system movement model is currently used to propagate the sample set. By utilizing some prior knowledge of the expected object silhouette, the quality of detecting and tracking is improved. 4. Interaction of user and computer improve the detecting and tracking precision.

However, object detecting and tracking is a difficult problem. Further research will focus on: (1) improving the detecting precision of target by using more effective algorithms, (2) improve the speed of detecting and tracking while guaranteeing precision.

7. Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grants No. 61003196; the Fundamental Research Funds for the Central Universities under Grants K50511040008.

9. References

[1] Alper Yilmaz, Omar Javed, Mubarak Shah, "Object tracking: a survey," ACM Computing Surveys, 2006, 38(4):1-45.
[2] Yong Rui, Yunqiang Chen, "Better proposal distributions: object tracking using unscented particle filter," Proc. of IEEE CVPR, San Diego:IEEE Computer Society Press,2001:786-793.
[3] Sankaranarayanan A C, Srivastava A, Chellappa R. Algorithmic and architectural optimizations for computationally efficient particle filtering. IEEE Transactions on Image Processing 2008; 17(5): 737-748.
[4] W. Qu, D. Schonfeld, M. Mohamed, "Real-time distributed multi-object tracking using multiple interactive trackers and a magnetic-inertia potential model," IEEE Transactions on Multimedia, 2007, 9(3):511-519.
[5] M. Jaward, L. Mihaylova, N. Canagarajah, "Multiple object tracking using particle filters," Proceedings of the IEEE Aerospace Conference.Big Sky, Montana: IEEE CS Press, 2006:8-16.
[6] Blom H A P, Bloem E A. Exact Bayesian and particle filtering of stochastic hybrid systems. IEEE Transactions on Aerospace and Electronic Systems 2007;43(1): 55-70.
[7] Tao Yang, Quan Pan, "Real-time multiple objects tracking with occlusion handling in dynamic scenes," IEEE Computer Vision and Pattern Recognition Conference, San Diego: IEEE Computer Society Press, 2005:970-975.
[8] M. Han, A. Sethiy, W. Hua, Y. Gong, "A detection-based multiple object tracking method," International Conference of Image Processing, USA: IEEE Computer Society Press, 2004:3065-3068.
[9] K. Hariharakrishnan, D. Schonfeld, "Fast object tracking using adaptive block matching," IEEE Transactions on Multimedia. 2005, 7(5): 853-859.
[10] Doucet, N. de Freitas, N. Gordon, editors, "Sequential monte carlo methods in practice," New York, Springer-Verlag, 2001.
[11] Y. Cheng. Mean shift, "Mode seeking, and clustering," IEEE Trans, PAMI, 1995, 17(8):790-799.
[12] Y. Bar-Shalom, A.G.Jaffer, "Tracking and data association," San Diego, CA, Academic Press, 1988.
[13] Rajashekar U, Bovik CA, Cormack KL. GAFFE: A gaze-attentive fixation finding engine[J].IEEE Transactions on Image Processing, 2008, 17(4): 564-573.
[14] Jing Li, Chen Zhaoqian, "No-rigid object real-time tracking based on particle filter algorithm," Journal of Nanjing University of Aeronautics & Astronautics, 2006,38(6):776-779.
[15] Sun Wei, Guo Baolong, "Multiple objects detecting and tracking with the pseudo particle filter," Journal of Xidian University, 2008, 35(2):248-253.