Improving Particle Filter with Support Vector Regression for Efficient Visual Tracking

Guangyu Zhu[1], Dawei Liang[1], Yang Liu[1], Qingming Huang[2], and Wen Gao[1,2]
[1]Department of Computer Science, Harbin Institute of Technology, Harbin, P. R. China
[2]Graduate School of Chinese Academy of Sciences, Beijing, P. R. China
{gyzhu, dwliang, yliu, qmhuang, wgao}@jdl.ac.cn

Abstract—**Particle filter is a powerful visual tracking tool based on sequential Monte Carlo framework, and it needs large numbers of samples to properly approximate the posterior density of the state evolution. However, its efficiency will degenerate if too many samples are applied. In this paper, an improved particle filter is proposed by integrating support vector regression into sequential Monte Carlo framework to enhance the performance of particle filter with small sample set. The proposed particle filter utilizes an SVR based re-weighting scheme to re-approximate the posterior density and avoid sample impoverishment. Firstly, a regression function is obtained by support vector regression method over the weighted sample set. Then, each sample is re-weighted via the regression function. Finally, ameliorative posterior density of the state is re-approximated to maintain the effectiveness and diversity of samples. Experimental results demonstrate that the proposed particle filter improves the efficiency of tracking system effectively and outperforms classical particle filter.**

Keywords-visual tracking; particle filter; support vector regression

I. INTRODUCTION

In recent years, particle filter [1, 2] has been successfully applied in computer vision community for visual tracking [3, 4] due to its ability to carry multiple hypotheses and relaxation of linearity/Gaussian assumption. Particle filter is based on sequential Monte Carlo approach, where the probability density is represented by a set of weighted samples (called particles). Large numbers of samples are needed in practice for two reasons: 1) to properly approximate posterior density of the state evolution over time steps and 2) to be able to recover from object loss. However, the size of sample set is directly related to the computational cost and should be kept as small as possible in order to improve the efficiency of particle filter.

When particle filter is running with a small sample set, the major problem is how to properly approximate the posterior density of state evolution so that the effectiveness and diversity of samples can be maintained to avoid the emergence of sample impoverishment. In classical particle filter, the posterior density cannot be approximated with small sample set properly due to inaccuracy of the sample weights. Moreover, sample degeneracy is an unavoidable phenomenon in particle filter. Resampling [1] can be implemented to solve this problem, but it will introduce sample impoverishment consequently that leads to the loss of effectiveness and diversity among the samples. When a small sample set is employed, impoverishment will be aggravated due to the improper posterior density. This will degrade the efficiency and diversity of samples more rapidly and seriously. Existing schemes use forward filter backward smoothing technology [5] or Markov Chain Monte Carlo method [6] to counteract this problem.

In this paper, we propose an improved particle filter by integrating support vector regression (SVR) into sequential Monte Carlo framework to achieve satisfied performance of particle filter with small sample set. Unlike [5, 6], the proposed particle filter adopts a simple but effective sample re-weighting scheme based on SVR. At each iteration stage, the regression function over the weighted sample set is obtained by SVR after update step. Then, each sample is re-weighted. Finally, the posterior density of the state evolution is re-approximated. Since sample weights are reevaluated, the posterior density re-approximated is more proper than the one in classical particle filter. Thus the effectiveness and diversity of the sample set is maintained and the problem of sample impoverishment is avoided.

The rest of the paper is organized as follows. In Section 2, we analyze the mechanism of particle filter in detail with an example of sampling importance resampling (SIR) filter and highlight the key issues of particle filter running with small sample set. In Section 3, SVR solution is introduced and the improved particle filter is proposed. In Section 4, experimental results on a real world visual tracking are presented. Conclusions are summarized in Section 5.

II. PARTICLE FILTER WITH SAMLL SAMPLE SET

Particle filter utilizes sequential Monte Carlo method for online inference within Bayesian framework. Monte Carlo simulation is used to approximate the probability density via a set of weighted samples. Prediction and Update are executed alternately at each time step. Resampling method is used to reduce the influence of the degeneracy problem. Exampling with SIR particle filter [7], Markovian transition kernel $p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$ is used to generate the sample set at current time step k and then updates the sample weights using likelihood function $p(\mathbf{z}_k \mid \mathbf{x}_k)$, where $\mathbf{x}_k$ is the state vector and $\mathbf{z}_k$ is the observation vector respectively. The algorithm of SIR filter is summarized in Fig. 1.

The posterior density of state $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$ at time step k is given by $\{(\mathbf{x}_k^i, w_k^i)\}_{i=1}^N$ in a discrete form after update step:

1. Initialization:
   Generate sample set $\{\mathbf{x}_0^i\}_{i=1}^N$ from the initial distribution $p(\mathbf{x}_0)$, set $k=1$.
2. Prediction:
   Draw predicted sample $\mathbf{x}_k^i \sim p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^i)$, $i=1,\ldots,N$
3. Update:
   Once the observation data $\mathbf{z}_k$ is measured, evaluate weight of sample $\tilde{w}_k^i = w_{k-1}^i p(\mathbf{z}_k \mid \mathbf{x}_k^i)$, and normalize $w_k^i = \tilde{w}_k^i \big/ \sum_{q=1}^N \tilde{w}_k^q$, $i=1,\ldots,N$.
4. Resampling:
   Generate new sample set $\{\mathbf{x}_k^j\}_{j=1}^N$ by resampling (with replacement) N times from $\{\mathbf{x}_k^i\}_{i=1}^N$, where $\Pr(\mathbf{x}_k^j = \mathbf{x}_k^i) = w_k^i$, and set weight $w_k^j = 1/N$.
   - Set $k = k+1$ and go back to step 2.

Fig. 1 SIR particle filter algorithm

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) \approx \sum_{i=1}^N w_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \tag{1}$$

where $\delta(\bullet)$ is the Dirac delta function, $\mathbf{z}_{1:k} = \{\mathbf{z}_1,\ldots,\mathbf{z}_k\}$ is the history of observations up to time step k. To approximate the posterior density properly using (1), two approaches can be applied. One is to increase the number of samples, and the other is to improve the accuracy of sample weights. Too many samples will degrade the efficiency of particle filter. Nevertheless posterior density is hard to be approximated properly with a small sample set, and then affects the performance of resampling, which may result in severe sample impoverishment. Thus, if we want to reduce the size of sample set, more accurate weight is obligatory.

According to above analysis, for particle filter running with a small sample set, weighting sample more accurately is the key to construct the proper approximation of the posterior density. In classical particle filter, it is separate that the weights are evaluated by likelihood function $p(\mathbf{z}_k \mid \mathbf{x}_k)$. However, the robustness of likelihood function in practice is influenced severely by the noise of circumstance such as background clutter, deformation of non-rigid object in visual tracking, etc. Therefore, a few weights will be corrupted, which mislead the corresponding false samples close to the true state and result in a noisy posterior distribution. The small set with false samples cannot approximate the posterior density properly and sample impoverishment will arise more rapidly and seriously than that with a large sample set. In next section, a solution based on support vector regression is proposed, which can re-weight samples accurately by smoothing the density via a regression function to eliminate the influence of noise in posterior.

III. SUPPORT VECTOR REGRESSION BASED PARTICLE FILTER

Support vector regression is briefly described as follows. Given a set of training set:

$$\{(\mathbf{x}^i, y^i) \mid \mathbf{x}^i \in \Re^d, y^i \in \Re, i=1,\ldots,N\} \tag{2}$$

where $\mathbf{x}^i$ is a sample point drawn from the space of input pattern, and y^i possibly corrupted by noise is generated by some function:

$$f : \Re^d \to \Re \tag{3}$$

Following the Vapnik's method [8], we can estimate function $f(\mathbf{x})$ using given training set and the result is:

$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i K(\mathbf{x}, \mathbf{x}^i) + b \tag{4}$$

where $K(\bullet)$ is the kernel function adopted by support vector regression.

With the context of particle filter, the posterior density of the state evolution is usually nonlinear/non-Gaussian, where we cannot specify its parameterized representation in advance. SVR is a nonparametric regression technique. Therefore it is more suitable to tackle our problem than other existing parametric regression methods. Moreover, SVR is robust to noisy distribution since $\varepsilon-$insensitive loss function is utilized and can work effectively even with small sample set [9].

Considering the weighted sample set $\{(\mathbf{x}_k^i, w_k^i)\}_{i=1}^N$ and (1), $\mathbf{x}_k^i$ can be treated as the sample point drawn from the current posterior distribution and w_k^i as the corresponding density value. Intuitively, we can construct regression function l_k over $\{(\mathbf{x}_k^i, w_k^i)\}_{i=1}^N$ at time step k:

$$l_k : \Re^{n_x} \to \Re \tag{5}$$

where n_x is the dimension of the state vector. SVR is adopted to obtain l_k, which has similar form as (4):

$$l_k(\mathbf{x}_k) = \sum_{i=1}^N \alpha_i K(\mathbf{x}_k, \mathbf{x}_k^i) + b_k \tag{6}$$

Afterwards, each sample is re-weighted using (6) and the new weighted sample set $\{(\mathbf{x}_k^i, \hat{w}_k^i)\}_{i=1}^N$ is generated, where each sample weight is refined.

Essentially, equation (6) transforms samples from a discrete space into a continuous space. Kernel $K(\bullet)$ is used for smoothing so that each sample is re-weighted using the information of its neighboring samples rather than separately in classical particle filter. Thus, the posterior density of state at time step k is re-approximated as follows:

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) \approx \sum_{i=1}^N \hat{w}_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \tag{7}$$

Fig. 2 illustrates the effectiveness of SVR based re-weighting scheme. The posterior density is assumed to be 1-D bimodal distribution approximated by 50 samples. It is obvious that the density after re-weighting, which is represented by (7), is more proper than the original one.

We name the improved particle filter as SVR particle filter and describe it in Fig. 3. Since re-weighting is implemented at every time step, the efficiency and diversity of sample set is maintained and the sample impoverishment is avoided consequently.

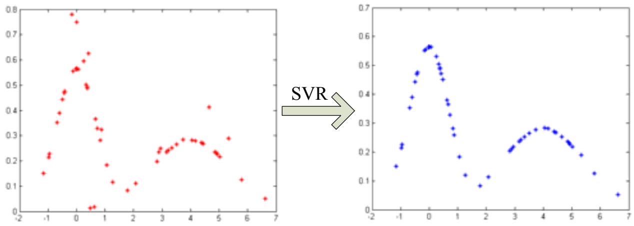


Fig. 2 Performance of SVR re-weighting

IV. Experiment

In this section, the improved particle filter is tested on a visual tracking problem to compare performance with classical particle filter. We implement SVR particle filter for tracking football player in a video sequence. SIR filter illustrated in Fig. 1 is utilized as comparison since it is the most common method in particle filter and does have the advantage of easy implementation.

The object of interest (football player) is represented by its bounding box. For both trackers, the state is described by a four dimensional vector representing the bounding box:

$$\mathbf{x}_k = (x, y, v_x, v_y)^T \tag{8}$$

where x and y are the coordinates of top-left point of bounding box, while v_x and v_y are the velocity of bounding box along x and y axis, respectively. The state dynamic model is defined as:

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{v}_k \tag{9}$$

where $\mathbf{v}_k$ is a zero mean Gaussian random variable. Matrix $\mathbf{A}$ is defined manually by assuming the constant velocity on player's motion. The likelihood function is computed based on the Bhattacharyya similarity of HSV histogram between the target and the candidates [4]. Suppose that the histogram of target is denoted by $q^*(i)$, where $i = 1,\ldots,N$ and N is the dimension of histogram. The distance of two histograms D is derived from the Bhattacharyya similarity:

$$D[q^*, q(\mathbf{x}_k)] = \left(1 - \sum_{i=1}^{N} \sqrt{q^*(i) q(\mathbf{x}_k, i)}\right)^{1/2} \tag{10}$$

Then the likelihood function at time step k is given by

$$p(\mathbf{z}_k \mid \mathbf{x}_k) \propto \exp\left(-\lambda D^2[q^*, q(\mathbf{x}_k)]\right) \tag{11}$$

where $\lambda = 10$ is set empirically. 50 samples are utilized in both trackers and the kernel function adopted by SVR is radial basis function (RBF).

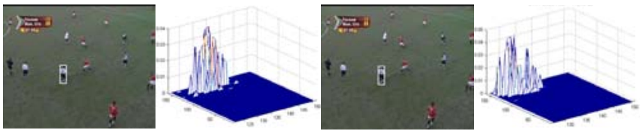
The experimental results are shown in Fig. 4. The left two columns are the tracking results and corresponding sample distributions of SIR particle filter, and the right two columns are the ones of SVR particle filter.

1. Initialization:
   Generate sample set $\{\mathbf{x}_0^i\}_{i=1}^N$ from the initial distribution $p(\mathbf{x}_0)$, set $k = 1$.
2. Prediction:
   Draw predicted sample $\mathbf{x}_k^i \sim p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^i)$, $i = 1,\ldots,N$
3. Update:
   Once the observation data $\mathbf{z}_k$ is measured, evaluate weight of sample $\tilde{w}_k^i = w_{k-1}^i p(\mathbf{z}_k \mid \mathbf{x}_k^i)$, and normalize $w_k^i = \tilde{w}_k^i \Big/ \sum_{q=1}^{N} \tilde{w}_k^q$, $i = 1,\ldots,N$.
4. SVR based Re-weighting:
   Obtain regression function l_k in (6) using SVR based on sample set $\{(\mathbf{x}_k^i, w_k^i)\}_{i=1}^N$. Re-weighting each sample's weight using (6) and generate refined set $\{(\mathbf{x}_k^i, \hat{w}_k^i)\}_{i=1}^N$.
5. Resampling:
   Generate new sample set $\{\mathbf{x}_k^j\}_{j=1}^N$ by resampling with replacement N times from $\{\mathbf{x}_k^i\}_{i=1}^N$ where $\Pr(\mathbf{x}_k^j = \mathbf{x}_k^i) = \hat{w}_k^i$, and set weight $w_k^j = 1/N$.
   - Set $k = k + 1$ and go back to step 2.

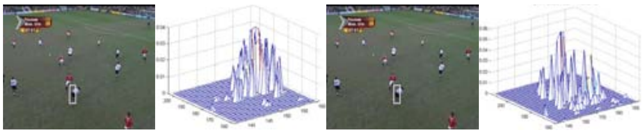
Fig. 3 SVR particle filter algorithm

At frame 25, two trackers have the similar sample distribution with the same estimation position. However, deformation of player occurs at frame 45, which results in more background noise being involved in the bounding box, and leads to sample weights being evaluated improperly in SIR particle filter. At frame 67 when player deformation finishes, the above problem is aggravated. The approximated posterior density is severely improper so that sample impoverishment occurs and the diversity of samples is lost. With a small sample set, SIR particle filter fails to recover the player because no samples fall into the neighborhood of the new position. In contrast, SVR particle filter maintains the diversity of samples as the figure of sample distribution illustrates and can recover the object at the subsequent time step. This situation can be demonstrated further at frame 100. SVR particle filter is able to not only solve the deformation problem but also counteract occlusion effectively. As illustrated in Frame 180 and 190 where the tracked player is occluded by the referee, SVR particle filter can still achieve satisfied result. The sample distribution underlying SVR particle filter is maintained much better than that of SIR particle filter through the whole tracking process. The video clip of this experiment can be found at [10].

We also utilize the classical particle filter with sufficient samples to perform the same experiment in order to compare the efficiency with SVR particle filter. 200 samples are used by SIR filter to achieve the similar tracking result as SVR filter. The running speed of SVR filter is about 1.8 times as fast as that of SIR filter. This demonstrates that SVR particle filter effectively reduces the size of sample set, meanwhile improves the efficiency of tracking system remarkably.

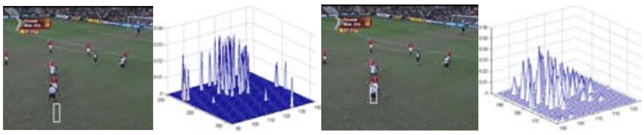


(a) Frame 25. SIR PF: (132, 141), SVR PF: (132, 141)

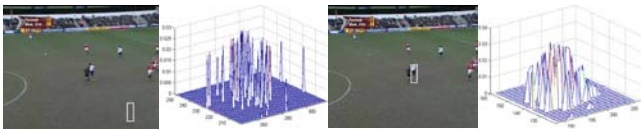


(b) Frame 45. SIR PF: (154, 185), SVR PF: (156, 185)

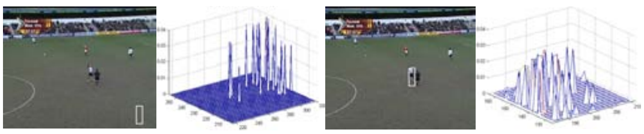
(c) Frame 67. SIR PF: (197, 213), SVR PF: (188, 217)



(d) Frame 100. SIR PF: (119, 227), SVR PF: (110, 183)



(e) Frame 180. SIR PF: (281, 227), SVR PF: (195, 138)



(f) Frame 190. SIR PF: (306, 235), SVR PF: (196, 143)

Fig. 4 Experimental results with two different trackers

V. CONCLUSION

In this paper, the performance of particle filter with small sample set is improved by integrating support vector regression into sequential Monte Carlo framework. SVR based re-weighting scheme is employed to approximate the posterior density properly. The effectiveness and diversity of samples are maintained meanwhile impoverishment is avoided. The efficiency of the proposed particle filter is demonstrated in a visual tracking problem by comparison with classical particle filter.

In our future work, we will investigate the sample propagation scheme and explore to apply the proposed particle filter to multiple object tracking.

ACKNOWLEDGMENT

This work is partly supported by NEC Research China on “Context-based Multimedia Analysis and Retrieval Program” and “Science 100 Plan” of Chinese Academy of Sciences. This work is a part of “SPISES: SPorts vIdeo Summarization and Enrichment System” at web address:

http:// www.jdl.ac/en/project/spises/SPISES.html

REFERENCES

[1] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking,” IEEE Trans. on Signal Processing, 50(2), pp. 174-188, Feb 2002.

[2] A. Doucet, N. de Freitas, and N. Gordon, Sequential Monte Carlo Methods in Practice, Springer-Verlag, New York, 2001.

[3] M. Isard, A. Blake, “Condensation—Conditional Density Propagation for visual Tracking,” Intl. J. Computer Vision, 29(1), pp. 5-28, 1998.

[4] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, “Color-based Probabilistic Tracking,” In Proc. Europ. Conf. Computer Vision, pp. 661-675, 2002.

[5] S. Godsill, A. Doucet, and M. West, “Methodology for Monte Carlo Smoothing with Application to Time-Varying Autoregressions,” In Proc. Int. Symp. Frontiers Time Series Modeling, 2000.

[6] B. P. Carlin, N. G. Polson, and D. S. Stoffer, “A Monte Carlo approach to Nonnormal and Nonlinear State-Space Modeling,” Journal of the American Statistical Association, 87(418), pp. 493-500, 1992.

[7] N. Gordon, D. Salmond, and A.F.M. Smith, “Novel Approach to Nonlinear and Non-Gaussian Bayesian State Estimation,” Proc. Inst. Elect. Eng., F, vol. 140, no. 2, pp. 107-113, 1993.

[8] V. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1995.

[9] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press, Cambridge, 2002.

[10] http://www.jdl.ac.cn/en/project/spises/demo.htm.