

Visual Tracking Using Particle Filters with Gaussian Process Regression

Hongwei Li, Yi Wu, and Hanqing Lu

National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China
{hwli, ywu, luhq}@nlpr.ia.ac.cn

Abstract. Particle degeneracy is one of the main problems when particle filters are applied to visual tracking. The effective solution methods on the degeneracy phenomenon include good choice of proposal distribution and use of resampling. In this paper, we propose a novel visual-tracking algorithm using particle filters with Gaussian process regression and resampling techniques, which effectively abate the influence of particle degeneracy and improve the robustness of visual tracking. The main characteristic of the proposed algorithm is that we incorporate particle filters with Gaussian process regression which can learn highly effective proposal distributions for particle filters to track the visual objects. Experimental results in challenging sequences demonstrate the effectiveness and robustness of the proposed method.

Keywords: Gaussian Processes, Particle Filter, Particle Degeneracy, Visual Tracking.

1 Introduction

Visual tracking is currently one of the most actively researched areas of computer vision and pattern recognition. During many practical applications such as surveillance and human-computer interfaces, visual tracking is a challenging task owing to the difficulties arising from illumination changes, occlusions, diverse appearance of objects, low-image resolution and noises etc.

Tracking algorithms generally involve two basic procedures: object observation (the object representation and similarity measurement) and object tracking control. Based on the nature of object tracking control, tracking algorithms can be roughly classified into two categories: deterministic methods and stochastic methods. In deterministic methods, the object is tracked by implementing an iterative search for the local maximum of a similarity cost function between the target image and the candidate images. These algorithms, such as the Mean Shift algorithm [1], are fast and efficient, but can not recover from the temporary tracking failures because of being sensitive to occlusion and clutter. In stochastic methods, the object is tracked by estimating the probability distribution and maintain multiple hypotheses in the state space. These algorithms, such as Particle Filter algorithm [2,3,4,5], can achieve more robustness and have the capability to recover from the temporary tracking failures.

The Particle Filter is a mainstream algorithm for online nonlinear/non-Gaussian Bayesian tracking, which is convenient to handle multivariate and nonlinear processes in contrast to traditional time-series techniques. The Particle Filter has been widely applied in many fields, such as visual tracking, video surveillance, mobile robot localization and failure detection etc. But Particle degeneracy, where all but one particle will have negligible weight after a few iterations [2], blocks the further development of algorithms, which means a large computational effort is devoted to updating particles whose contribution is almost zero. The degeneracy problem can be effectively mitigated by choosing better proposal distribution and using resampling techniques.

The main idea of resampling techniques is to reduce the influence of particles with small weights and to focus on particles with large weights. In this paper, we use resampling techniques whenever a significant degeneracy is observed. Better proposal distributions can minimize the variance of the true weights. For example, Freitas [6] introduced Extended Kalman Filtering into the particle filter algorithm. Eric A. Wan [7] presented unscented Particle Filter by incorporating the unscented Kalman Filtering. In this paper, we can learn effective proposal distributions for particle filters by incorporating Gaussian process regression, which can provide full predictive distributions modeling predictive uncertainties and choose the particles with the highest likelihood. In the following sections, the key steps of proposed algorithm are described in detail.

The organization of the paper is as follows: Section 2 briefly reviews the algorithm of Particle Filters. Gaussian process regression and the proposed algorithm with Gaussian process proposals are presented in Section 3. Experimental Results and Analysis are discussed in Section 4. The conclusions and possible extension are given in Section 5.

2 Particle Filters

In this section, we briefly review some basic notions of particle filter algorithm. Particle filters, known as sequential Monte Carlo method, are based on the point mass representations of probability densities. Through using the Bayesian inference framework, the particle filter algorithm is developed for visual tracking problem viewed as a state-space estimation problem. These state-space methods focus attention on the state vector(feature vector) of dynamic system, which is convenient to handle multivariate data and contains the correlative information (features) describing this dynamic system. For the sake of making a better state estimation of tracked objects, the importance sampling technique is employed on particle filters to obtain the samples of random variable generated from other distribution, called proposal distribution, which is easy to be sampled from.

2.1 A Brief Review of Particle Filters

Let $X = \{x_t, t \in \mathbb{N}\}$ denote the state sequence and $Y = \{y_t, t \in \mathbb{N}\}$ the corresponding observation sequence respectively. The objective of tracking is to recursively estimate the state $\{x_t\}$ of tracked objects from the observation $\{y_t\}$:

$$y_t = f_t(x_t, n_t), t \in \mathbb{N} \tag{1}$$

where n_t denotes an i.i.d observation noise at time t .

Based on the Bayesian rule and conditional independence properties, the posterior distribution of the state variable $\{x_t\}$ can be formulated as follows:

$$p(x_t|y_{1:t}) \propto p(y_t|x_t) \int p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \tag{2}$$

The function f in Equation (1) is generally nonlinear/non-Gaussian that makes the posterior distribution $p(x_t|y_{1:t})$ analytically intractable, so generating samples from this posterior distribution is usually not possible and the analytic approximations of integrals are needed.

In view of visual tracking problems, the true posterior distribution $p(x_t|y_{1:t})$ can be approximated by a weighted particle set $\{x_t^i, w_t^i\}_{i=1}^{N_s}$, where N_s is the number of particles and the weights are normalized by $\sum_{i=1}^{N_s} w_t^i = 1$. All the particles are directly generated from an approximation distribution $q(x_t|x_{t-1}, y_t)$, called proposal distribution, which is easy to be sampled from. Then the Particle filter algorithm proceeds with two key steps:

(1) The updated weights. Drawing N_s particles from the proposal density, each particle is assigned by appropriate importance weights:

$$w_t^i \propto w_{t-1}^i \frac{p(y_t|x_t^i)p(x_t^i|x_{t-1}^i)}{q(x_t^i|x_{t-1}^i, y_t)} \tag{3}$$

(2) The predictive distribution. By introduced by the modified weights w_t^i , the predictive posterior distribution can be approximated as:

$$p(x_t|y_{1:t}) \propto \sum_{i=1}^{N_s} w_t^i \delta(x_t - x_t^i) \tag{4}$$

where the function $\delta(\cdot)$ is the Dirac delta measure.

Note that the proposal distribution $q(x_t|x_{t-1}, y_t)$ is significant for any Monte Carlo method and should be properly approximated. In this paper, we propose Gaussian Processes as the proposal distribution, which incorporates arbitrary features into the covariance matrix modeling the uncertainties and provides a transition probability needed for the weight updating. Furthermore, Gaussian process proposal distribution, described in the following section, can choose the particles with highest likelihood by the optimal algorithm.

3 Gaussian Processes

Gaussian processes (GP) [8] [9] can facilitate modeling the uncertainty of complex data sets, and provides a completely theoretical framework for model selection and probability prediction simultaneously. Gaussian processes is a generalization of a multivariate Gaussian distribution and has the marginalization property. GP controls the properties of random data x by a random

process $f(x)$ and synchronously describes this random process by a probability distribution. GP describes a distribution over function and is fully specified by the mean function $m(x) = E[f(x)]$ and the covariance function (kernel function) $K(x, x') = E[(f(x) - m(x))(f(x') - m(x'))]$ of this random process $f(x)$:

$$f(x) \sim GP(m(x), K(x, x')) \quad (5)$$

where the kernel $K(x, x')$ is usually chose as the form of Mercer kernel. For example, the Radial Basis Function (RBF) kernel function has the following form:

$$K(x, x') = \theta_1 \exp\left(-\frac{(x - x')^T(x - x')}{2\theta_2^2}\right) \quad (6)$$

where θ_1 and θ_2 are the hyperparameters of the RBF kernel, which are generally selected by maximizing the marginal likelihood (evidence).

3.1 Gaussian Process Regression

We assume we are given a dataset (input vectors) $X = \{x_i\}_{i=1}^N$ and the corresponding target values $Y = \{y_i\}_{i=1}^N$. The graphical representation of Gaussian process regression [10] [11] is shown in Figure 1: The nodes are shaded to represent different treatments. White shaded nodes are unobserved variables (the latent function $f_i = f(x_i)$), grey shaded nodes are observed variables (input vectors X and target values Y) and black shaded nodes are optimized (hyperparameters θ of kernel function K).

The main idea of Gaussian process regression is to assume that there is an unobservable latent function $f(x)$ which is imposed on a Gaussian process prior $f(X, \theta) \sim GP(0, K)$, and the latent function preserves the mapping relationships between input vectors X and target values Y : $Y = f(X) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ is an i.i.d Gaussian noise. Based on the Bayesian theorem, the posterior probability of the latent function f given the input vectors X can be written as:

$$p(f|y, X, \theta) = \frac{p(y|f)p(f|X, \theta)}{p(y|X, \theta)} \quad (7)$$

where $p(y|f)$ is a Gaussian likelihood function, $p(f|X, \theta)$ is a Gaussian process prior and $p(y|X, \theta)$ is the normalization factor known as the evidence for the hyperparameters.

Based on the marginalization (consistency) property of Gaussian processes, the mean μ and the variance Σ of the predictive distribution is obtained as following:

$$\mu = k_*^T (K + \sigma^2 I)^{-1} y \quad (8)$$

$$\Sigma = k(x_*, x_*) - k_*^T (K + \sigma^2 I)^{-1} k_* \quad (9)$$

where $k_* = k(x, x_*)$ is the covariance matrix between the known target value x and the new target x_* .

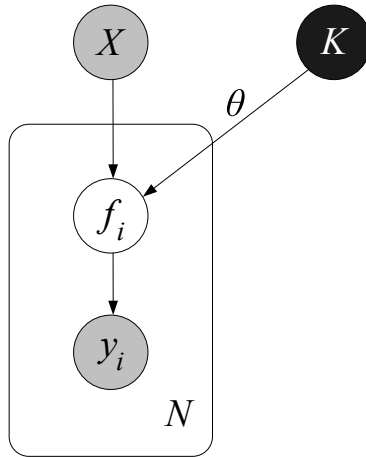


Fig. 1. The graphical representation of Gaussian Process Regression

3.2 Particle Filters with Gaussian Process Proposal

Based on the ideas of the previous sections, we propose the algorithm of particle filters with Gaussian process proposal as following:

- 1. At each time $t - 1$, we obtain N_s particles (i.e., target candidates) and extract the feature vectors $X_{t-1} = \{x_{t-1}^i\}_{i=1}^{N_s}$ for each particle;
- 2. Then we calculate the similarity (distance) between the target and target candidate: $Y_{t-1} = \{y_{t-1}^i\}_{i=1}^{N_s} = \{w_{t-1}^i\}_{i=1}^{N_s}$;
- 3. Given the feature vectors X_{t-1} and the similarity measurement Y_{t-1} , we learn the Gaussian process regression model (hyperparameters θ of the covariance function K) by maximizing the marginal likelihood as following:

$$p(y|X, \theta) = \int p(y|f)p(f|X, \theta)df \tag{10}$$

and the covariance function K has a squared-exponential form in this paper:

$$K(x, x') = \theta_1 \exp\left(-\frac{(x - x')^T(x - x')}{2\theta_2^2}\right) + \sigma^2\delta_{xx'} \tag{11}$$

where the optimal hyperparameters $\theta^* = (\theta_1^*, \theta_2^*, \sigma^*)$ is needed to be learned in this step;

- 4. At each time t , we sample n times and draw N_s particles each time from Gaussian process proposal as the following formula:

$$\{x_t^i\}_{i=1}^{N_s} = m + chol(K^*)u \tag{12}$$

where $chol(K^*)$ is the Cholesky decomposition of covariance matrix K^* with the optimal hyperparameters θ^* , m is the mean and $u \sim \mathcal{N}(0, I)$. Then we

choose one N_s particles set from n sets, which can maximize the marginal likelihood in Equation (10), namely, this selected set particles with highest likelihood can maximize the item $p(y_t|x_t^i)$ in Equation (3);

- 5. Assign the weight w_t^i to each particle according to Equation (3);
- 6. Resampling technique is applied to decrease the number of low weighted particles and to increase the ones with high weighted particles.

To sum up the ideas described above, a pseudo code description of this proposed algorithm is given as following:

Algorithm 1. Particle Filters with Gaussian Process Proposal (PFGP)

$[\{x_t^i, w_t^i\}_{i=1}^{N_s}] = PFGP[\{x_{t-1}^i, w_{t-1}^i\}_{i=1}^{N_s}, y_t]$

- Extract the feature vectors: $X_{t-1} = \{x_{t-1}^i\}_{i=1}^{N_s}$
- Calculate the similarity: $Y_{t-1} = \{y_{t-1}^i\}_{i=1}^{N_s} = \{w_{t-1}^i\}_{i=1}^{N_s}$
- Learn the optimal hyperparameters θ^* of the covariance function K

for $j = 1 : n$ **do**

- Draw N_s^j particles from GP proposal according to Equation (12)
- Choose one N_s particles set with highest likelihood in Equation (10)

end

for $i = 1 : N_s$ **do**

- Assign the weight w_t^i to each particle according to Equation (3)

end

- Calculate total weight: $sum = \sum_{i=1}^{N_s} w_t^i$

for $i = 1 : N_s$ **do**

- Normalize: $w_t^i = sum^{-1} w_t^i$

end

- Evaluate the Particle Degeneracy by a certain rule: N_e

if N_e below threshold N_T **then**

- Resampling
- the weights are reset to $w_t^i = 1/N_s$

end

4 Experiments

To order to evaluate the effectiveness and robustness of the particle filters with Gaussian process proposal algorithm, we have experimented with jogging sequences (200 frames) and person-car sequence (120 frames) respectively under the different experimental conditions. The number of particles is set to 100 in the each experiment. We have also tried to compare the results of our proposed algorithm to the results of standard Particle Filters algorithm.

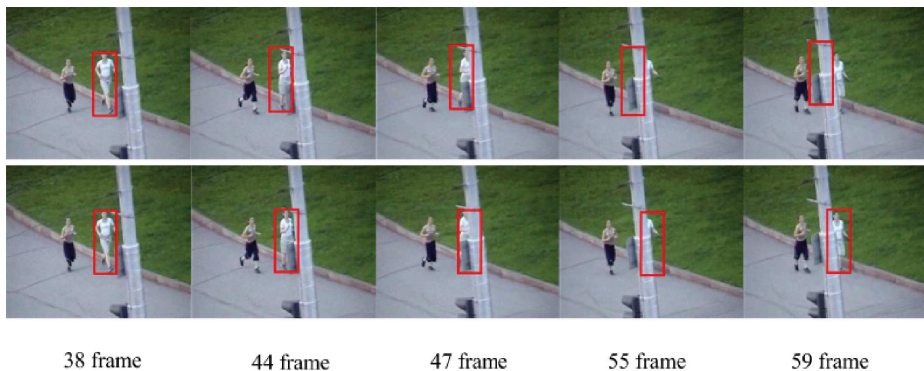


Fig. 2. Tracking results of jogging sequence employing color histogram as features. Standard Particle Filter tracker(the first row). Our proposed tracker (the second row).

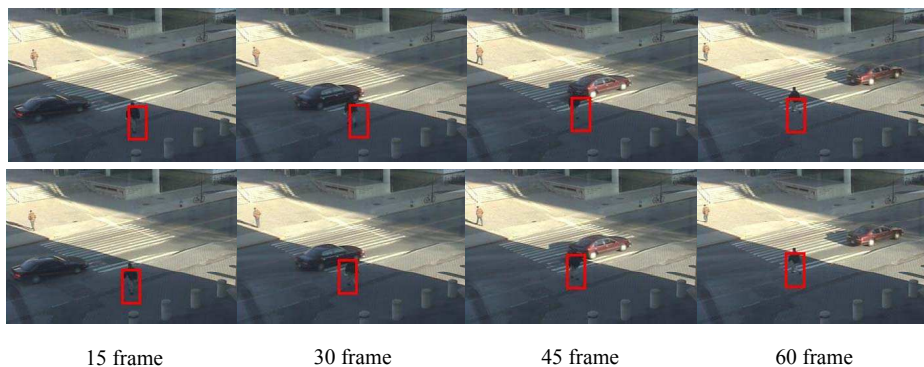


Fig. 3. Tracking results of person-car sequence employing color histogram as features. Standard Particle Filter tracker(the first row). Our proposed tracker (the second row).

4.1 Experiment 1

In experiment 1, for the purpose of demonstrating the effectiveness of proposed algorithm, we employ the simplest features, the normalized 8-bin color histogram, describing the target and candidate regions for each particle. For the similarity measure, we apply the Bhattacharyya distance [12], as the following equation, to measure the distance between the normalized color histograms of two regions.

$$B(h_t, h_c) = \sum_{i=1}^m \sqrt{h_t^i h_c^i} \quad (13)$$

where h_t and h_c respectively denote the m -bin normalized color histograms of target and candidate.

In Figure 2 and Figure 3, we demonstrate the tracking results of jogging sequence and person-car sequence respectively. The first row of each figure

corresponds to the standard particle filter tracker, and the second rows to the results of our method. We can notice that the jogger is occluded from 44 frame to 59 frame and the color of clutters is similar to the target in the jogging sequence. Also in the person-car sequence, the color of the shadow and the car is similar to the walker. As a result, the standard Particle Filters method fails to track the target in the occlusion, while our proposed method better deals with the occlusion case with weak features exactly demonstrating the effectiveness of our algorithm.

4.2 Experiment 2

In experiment 2, we employ the stronger features, 7-dimensional feature[13], describing the target and candidate regions for each particle:

$$F(x, y) = (x, y, \Delta_x(x, y), \Delta_y(x, y), R(x, y), G(x, y), B(x, y)) \quad (14)$$

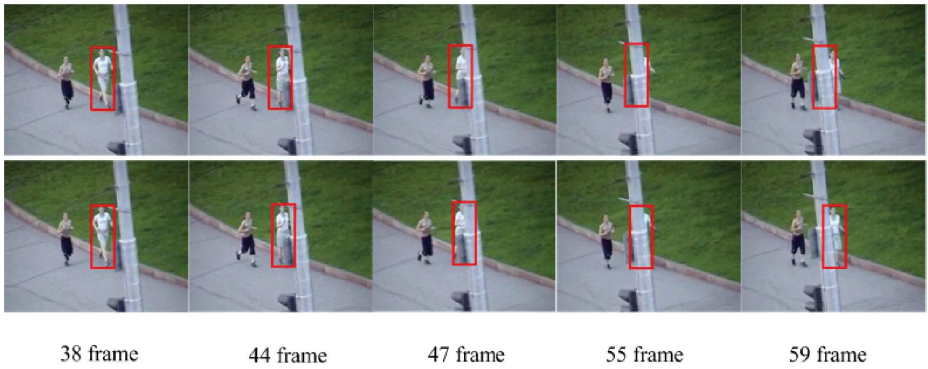


Fig. 4. Tracking results of jogging sequence employing 7-dimensional features. Standard Particle Filter tracker(the first row). Our proposed tracker (the second row).

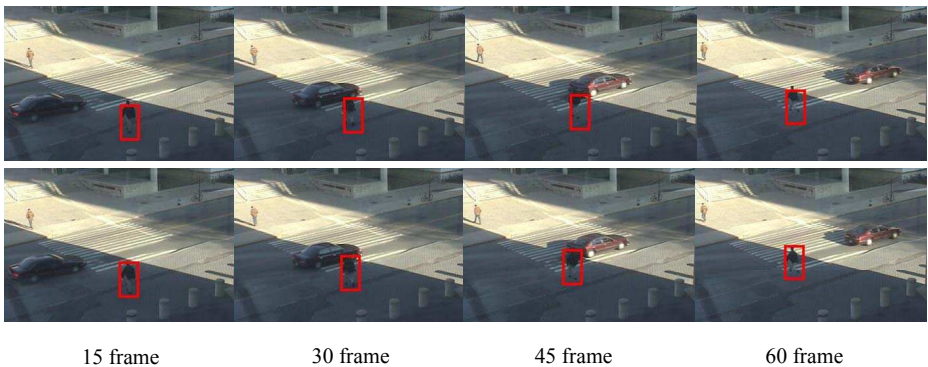


Fig. 5. Tracking results of person-car sequence employing 7-dimensional features. Standard Particle Filter tracker(the first row). Our proposed tracker (the second row).

where x and y are pixel location, Δ_x and Δ_y denote the intensity derivatives, R , G and B are RGB color values.

As the results shown in Figure 4 and Figure 5, we note that the standard Particle Filter tracker and our proposed tracker also improve the performance over the occlusion with the stronger features, simultaneously our proposed method obtains the optimal tracking results that further demonstrate the effectiveness and robustness of our proposed algorithm.

5 Conclusion

In this paper we have presented an algorithm that applies Gaussian process regression as the proposal distribution of Particle filters, which can choose the particles with the highest likelihood for particle filters, and improve the robustness and ability of the standard particle filter tracker, even obtain better performance in the occlusion case. As future work, we plan to extend the approach in order to be able to track multiple objects with occlusions and improve the multi-object tracking performance.

Acknowledgments. The work is supported by National Natural Science Foundation of China (Grant No.60605004), Key Project of National Natural Science Foundation of China (Grant No.60833006) and National High-Tech Research and Development Plan of China (863) (Grant No.2006AA01Z117).

References

1. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based Object Tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence* 25(5), 564–577 (2003)
2. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions On Signal Processing* 50(2) (2002)
3. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-Based Probabilistic Tracking. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*. LNCS, vol. 2350, pp. 661–675. Springer, Heidelberg (2002)
4. Okuma, K., Taleghani, A., de Freitas, N., Little, J.J., Lowe, D.G.: A boosted particle filter: Multitarget detection and tracking. In: Pajdla, T., Matas, J.(G.) (eds.) *ECCV 2004*. LNCS, vol. 3021, pp. 28–39. Springer, Heidelberg (2004)
5. Jin, Y., Mokhtarian, F.: Variational Particle Filter for Multi-Object Tracking. In: *International Conference on Computer Vision* (2007)
6. De Freitas: Sequential Monte Carlo methods to train neural network models. *Neural Computation* 12(4), 955–993 (2000)
7. Wan, E.A., van der Merwe, R.: The Unscented Kalman Filter for Nonlinear Estimation. *Adaptive Systems for Signal Processing, Communications, and Control Symposium* (2000)
8. Rasmussen, C.E., Williams, C.K.I.: *Gaussian Processes for Machine Learning*. MIT Press, Cambridge (2006)

9. Rasmussen, C.E.: Advances in Gaussian Processes. Advances in Neural Information Processing Systems (2006)
10. Williams, C.K.I.: Prediction with Gaussian processes:from the linear regression to linear prediction and beyond. Learning and Inference in Graphical Models (1998)
11. Csato, L., Opper, M.: Sparse online Gaussian processes. Neural Computation (2002)
12. Nummiaroa, K., Koller-Meierb, E., Gool, L.V.: An Adaptive Color-Based Particle Filter. Image and Vision Computing 21, 99–110 (2003)
13. Wu, Y., Wu, B., Liu, J., Lu, H.: Probabilistic Tracking on Riemannian Manifolds. In: The 19th International Conference on Pattern Recognition (2008)