SEQUENTIAL PARTICLE GENERATION FOR VISUAL TRACKING

Yuanwei Lao, Junda Zhu, and Yuan F. Zheng

Dept. of Electrical and Computer Engineering
The Ohio State University, Columbus, Ohio 43210
Email: {laoy, zhuj, zheng}@ece.osu.edu

ABSTRACT

A novel variant of particle filters is presented, where new particles are generated sequentially by adapting the proposal density dynamically according to the likelihood of the current particle which are just generated. The new algorithm is able to capture more nonlinear motion and produce a better localization of the moving target in an efficient way. Experiments on both synthetic and real-world data verify its effectiveness and demonstrate its superiority over the generic particle filter.

Index Terms**—** Particle filter, tracking, product enclosure

1. INTRODUCTION

With the increasing availability and popularity of cameras, visual tracking is becoming even more important in many applications. Human tracking is used for behavior analysis and event detection in video surveillance, while vehicle tracking plays a significant role in intelligent traffic systems. Meanwhile, visual tracking is found to be a challenging problem due to various reasons, such as rapid and/or nonlinear target motion, clutter background, occlusions, etc.

Among the state-of-the-art algorithms, Particle Filter (PF) [1], or Condensation algorithm [2], has achieved popularity due to its capability to handle nonlinear and non-Gaussian models. Compared to the classical mean shift tracking [3], PF shows advantages in robustness, occlusion handling, flexibility in multi-target tracking, model changes, etc., while the main disadvantage is the high computational complexity. Especially when the target is in nonlinear abrupt motions, which is common in low-frame-rate videos, the number of particles has to be increased significantly to cover a larger search space. How to improve its efficiency becomes a hot topic in recent years. One way is to introduce an adaptive PF. For example, Zhou et al. [4] proposed to tune both the number of particles and the Gaussian variance of the motion model dynamically according to the tracking status. Combining mean shift with PF is another way. Maggio and Cavallaro [5] presented a hybrid scheme, in which every particle is applied with mean shift until it reaches a stable position during each iteration. Cai et al. [6] introduced a similar boosted PF into multi-target tracking. By doing so, each particle is guided by a deterministic search to a local optimum and becomes more representative for the modes of the posterior probability.

A crucial reason why the generic PF is inefficient is the lack of the likelihood knowledge of the particle. Kreucher et al. [7] made the same observation and proposed a particle screening scheme, where a large number of particles are first generated, and only a portion is selected based on overall evaluations. Unfortunately the improvement of sampling efficiency is at the cost of even higher computational complexity.

Our contribution is to introduce a novel and adaptive variant of PF, named *Sequential Particle Filter* (SPF), in which particles are proposed sequentially, rather than all at once. Based on the likelihood of the current particle, the proposal density is dynamically updated for the next particle, so that particles are sampled to be either more concentrated in the high-likelihood area or scattered in the low-likelihood area to capture severe nonlinear motions. Without resorting to other methods, the intrinsic resource, i.e. the knowledge of likelihood, is fully exploited to improve the sampling efficiency. With the same number of particles as generic PF, SPF is able to achieve a significantly higher tracking accuracy and capture abrupt motions where generic PF usually fails. In other words, SPF requires much less particles to obtain a similar performance as generic PF.

Section 2 reviews the generic PF, and Section 3 introduces the proposed algorithm as well as a new motion prior initialization method. In Section 4, comprehensive experiments are performed to compare the new algorithm with the generic one. Section 5 concludes the whole work.

2. GENERIC PARTICLE FILTER

Let x_k and z_k denote the state vector and the measurement at time k, respectively, and we use $Z_k = \{z_1, .., z_k\}$ to represent the set of measurements till time k. Under Bayesian framework, the fundamental problem is to compute the posterior probability $p(x_k|Z_k)$ given the motion model $p(x_k|x_{k-1})$ and the measurement model $p(z_k|x_k)$. PF is one of the most successful ways to handle the nonlinear/non-Gaussian models by implementing a Bayesian filtering based on the Monte Carlo method. The main idea is to use an enough number of particles in the state space and their corresponding weights, $\{x_k^i, w_k^i\}_{i=1}^N$, to approximate the posterior distribution in a

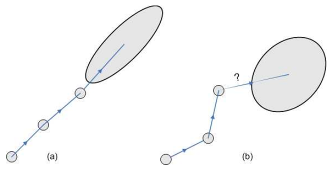


Fig. 1. Motion model initialization.

discrete way by $p(x_k|Z_k) \approx \sum_{i=1}^{N} w_k^i \delta(x_k - x_k^i)$. A carefully designed *proposal density* $q(x_k|x_{k-1}, z_k)$ (PD, also called *importance density*) is used to generate all particles x_k^i, while the associated weights w_k^i are calculated iteratively in Eq. (1) by factorizing the posterior distribution and PD:

$$w_k^i \propto w_{k-1}^i \cdot p(z_k|x_k^i) p(x_k^i|x_{k-1}^i) / q(x_k^i|x_{k-1}^i, z_k). \quad (1)$$

where $p(z_k|x_k^i)$ is usually named likelihood. The optimal PD $q(x_k|x_{k-1}, z_k)$ is proven to be $p(x_k|x_{k-1}, z_k)$, which is not computationally feasible in most cases, while the most popular choice is the motion model, $q(\cdot) = p(x_k|x_{k-1})$. With this substitution, Eq. (1) is reduced to $w_k^i \propto w_{k-1}^i p(z_k|x_k^i)$. To prevent PF from degenerating, a re-sampling technique is usually introduced at the end of each iteration. Hence the weight calculation is further simplified to $w_k^i = p(z_k|x_k^i)$. This is normally referred as the generic PF (GPF).

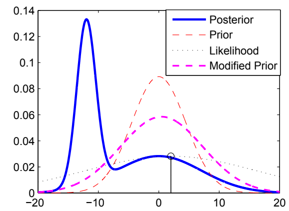
3. THE PROPOSED ALGORITHM

3.1. Motion Prior Initialization

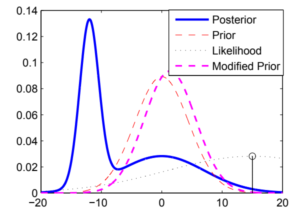
Previously, the most frequently used motion models are the ones with isotropic Gaussian distributions, which are not efficient in many cases. As shown in Fig. 1(a), when the target motion is relatively linear, a polarized Gaussian will greatly enhance the sampling efficiency by constraining the particles to spread along the moving direction, in comparison with an isotropic one for nonlinear motions in Fig. 1(b).

Therefore we propose an adaptive anisotropic Gaussian based on the target motion pattern, which is very much like an inverse procedure of the Principal Component Analysis (PCA) [8]. First, we calculate the major eigen-vector by $V_1 = x_k - x_{k-1}$. Let the normalized version be $V_1 = [\beta_1 \ \beta_2]^T$, and the second eigen-vector should satisfy $V_1^T V_2 = 0$, by which we obtain $V_2 = [\beta_2 \ -\beta_1]^T$. Based on the average displacement of the previous few frames, we have an estimate of the target speed square, $\rho = (|x_k - x_{k-1}|^2 + |x_{k-1} - x_{k-2}|^2)/2$. Then along the major axis, we have the first eigen-value $\lambda_1 = \rho$, while on the minor axis we assign $\lambda_2 = \alpha\rho$, where $\alpha \in [0, 1]$ adjusts the tradeoff between efficiency and nonlinearity. When $\alpha \rightarrow 1$, the method is the least efficient and accommodates most nonlinearity. When α is close to zero, it is more efficient while considering less nonlinearity. Once we have $D = diag(\lambda_1, \lambda_2)$ and $V = [V_1 \ V_2]$, the covariance matrix is given by $\Sigma_0 = VDV^{-1}$. Thus with the estimated speed $v_k = x_k - x_{k-1}$, we have the 2^{nd}-order motion prior:

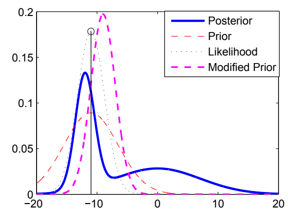
$$p(x_k|x_{k-1}) = N((x_{k-1} + v_{k-1}), \Sigma_0). \quad (2)$$



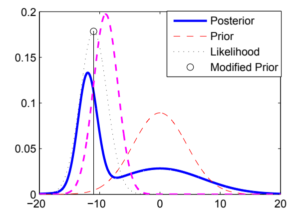
(a) Low likelihood & close to mean



(b) Low likelihood & far from mean



(c) High likelihood & close to mean



(d) High likelihood & far from mean

Fig. 2. Four typical scenarios for proposal density updating: (a) the variance increases; (b) the previous prior is kept; (c)&(d) the next prior is dragged towards the current particle.

3.2. Sequential Particle Generation

The complexity of PF is directly determined by the sampling efficiency. In most previous PFs, particles are generated i.i.d. by the same PD, where it is likely that many particles emerge in a low-likelihood area due to target abrupt motions. The lack of correlation between particles is one major reason for the inefficiency. We propose to generate particles sequentially, by which the likelihood knowledge of the current particle can be fused to the proposal for the next particle.

As shown in Fig. 2 for the 1-D case, the solid curve represents the underlying posterior distribution, while the thin-dashed Gaussian denotes the current proposal. If the current particle yields a low likelihood (the black circle), the case that it is close to the proposal mean should be differentiated from the case that it is far away. As implied in the former case, Fig. 2(a), the probability that the posterior mode exists around the proposal mean is reduced, such that the search space should be enlarged to increase the chance of capturing the mode. We propose that the variance of the next proposal (thick-dashed Gaussian) should be amplified, in which the closer the particle is to the mean, the larger the variance should be. The latter case provides no extra information, and the best guess is still the previous proposal density or its approximate, as shown in Fig. 2(b). However, when the particle yields a higher likelihood, either close to or far away from the proposal mean, as in Figs. 2(c) and 2(d), it indicates that the posterior mode is likely to be around, and the succeeding particles should be more likely to appear in that area. Therefore the next proposal is dragged towards the current particle in both cases.

Based on these intuitive ideas, we propose an adaptive scheme to update the PD by utilizing both the particle distance

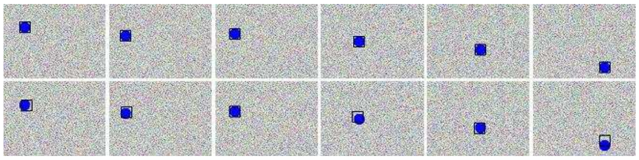


Fig. 3. SPF (top row) vs GPF (bottom row) with 60 particles on synthetic data frame 10, 18, 24, 32, 41 and 48 (total 50 frames).

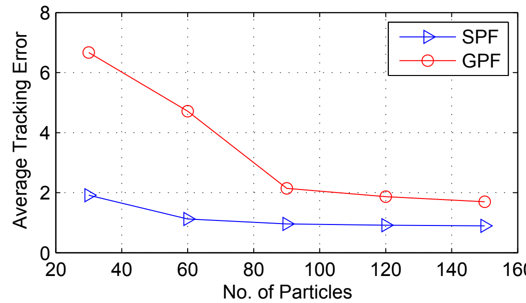


Fig. 4. Quantitative comparison on synthetic data.

from the proposal mean and its likelihood. Let $q_i(x_k|x_{k-1}) = N(\mu_i, \Sigma_i)$ be the i^{th} PD at time k. The likelihood of particle x_k^i is obtained by the measurement model $L_i = p(z_k|x_k^i)$, and we impose a multivariate Gaussian distribution onto this particle with the peak value as the likelihood, named the likelihood distribution, $p_d(z_k|x_k^i) = N(x_k^i, \Sigma_L)$. In the 2-D case, we solve $L_i = 1/(2\pi \cdot |\Sigma_L|^{1/2})$ for the covariance matrix Σ_L and obtain $|\Sigma_L| = 1/(2\pi L_i)^2$. We therefore select $\Sigma_L = diag([1/(2\pi L_i)\ 1/(2\pi L_i)])$.

For each new frame, we initialize the PD with the motion prior $q_1(x_k|x_{k-1}) = p(x_k|x_{k-1})$ by Eq. (2), and then obtain the first particle $x_k^1 \sim q_1(x_k|x_{k-1})$. The second PD and the successive PDs $q_i(x_k|x_{k-1})$ are iteratively updated in the same fashion by,

$$q_i(x_k|x_{k-1}) \propto q_{i-1}(x_k|x_{k-1})^{\lambda_i} \cdot p_d(z_k|x_k^{i-1}), \qquad (3)$$

where λ_i is the parameter determined by the distance:

$$\lambda_i = 1 - exp(-\alpha||x_k^i - \mu_i||^2). \qquad (4)$$

where μ_i is the current proposal mean and α is the parameter adjusting the converging speed of λ_i. The key idea is when x_k^i is close to μ_i, $\lambda_i \to 0$; otherwise $\lambda_i \to 1$. The major advantage of this scheme is to utilize the *product enclosure* property of multivariate Gaussians. By multiplying the current PD and the likelihood Gaussian in Eq. (3), we obtain the updated PD $q_{i+1} = N(\mu_{i+1}, \Sigma_{i+1})$ for the next particle with

$$\begin{cases} \Sigma_{i+1} = (\lambda_i \Sigma_i^{-1} + \Sigma_L^{-1})^{-1} \\ \mu_{i+1} = \Sigma_{i+1}(\lambda_i \Sigma_i^{-1} \mu_i + \Sigma_L^{-1} x_k^i) \end{cases}. \qquad (5)$$

In summary, we have the particles $x_k^i \sim q_i(x_k|x_{k-1})$ and corresponding weights $w_k^i = p(z_k^i|x_k^i)$. Once we collect $\{x_k^i, w_k^i\}_{i=1}^N$, we could perform the same estimation as the GPF. Compared to the previous adaptive schemes, SPF is simple yet effective. Its uniqueness lies in that PD is updated for each particle to fully exploit the likelihood knowledge.

4. NUMERICAL RESULTS

In this section, we carry out experiments to test the proposed algorithm on both synthetic and real-world image sequences. We specially select some difficult sequences with frequently nonlinear and abrupt motions to compare the performance of SPF with GPF. Among the appearance-based models, such as color, edge, texture, contour, etc., we choose the most popular and the simplest color histogram as our likelihood model, which has been applied in many previous works, such as [3] [6] [9]. In specific, a joint RGB histogram with 10 bins per channel is utilized.

4.1. Synthetic Image Sequences

We first generate various synthetic sequences where a blue target moves from the top-left towards the bottom-right in a noisy background. By imposing an additive Gaussian noise onto the constant speed, the target imitates a nonlinear movement. For both SPF and GPF, we initialize the proposal covariance to be $\Sigma_0 = diag([25\,25])$ and repeat the experiments over 300 times. In Fig. 3, a typical sequence of tracking results are shown for comparison. With only 60 particles, SPF is able to locate the target exactly in almost every frame, while GPF encounters obvious tracking errors around Frame 32 and 48 due to abrupt movements. When we select $\alpha = 0.2$, the average tracking error of SPF is only 1.1267 pixels/frame. We also varies the number of particles in both schemes, and the average tracking errors (300 repetitions) are plotted in Fig. 4. With only 30 particles, SPF achieves a similar performance as GPF with over 120 particles. The significantly lower errors demonstrate the superiority of SPF over GPF.

4.2. Real-World Applications

Then we test SPF and GPF on several real-world image sequences with $\alpha = 1$ and 60 particles. The first one is the 'Stennis' sequence, in which the white ball is moving fast up and down and changing directions frequently. As shown in Fig. 5, GPF ($\Sigma_0 = diag([100\,100])$) is easy to lose track due to rapid movement though it could re-capture the target from time to time. SPF, however, successfully traces the target in every frame by capturing the motion information and adapting the sampling. On a workstation with an Intel Xeon CPU and 3G RAM, the average computing times of GPF and SPF in MATLAB without optimization are approximately 38.6 and 43.3 milliseconds per frame, respectively. The limited difference denotes the overhead for updating PD, which implies that for GPF to achieve a similar performance, the computational cost will be significantly higher than SPF due to much more particles needed. In Fig. 6, we show a sequence of a pool table tracked with SPF. The tracker locates

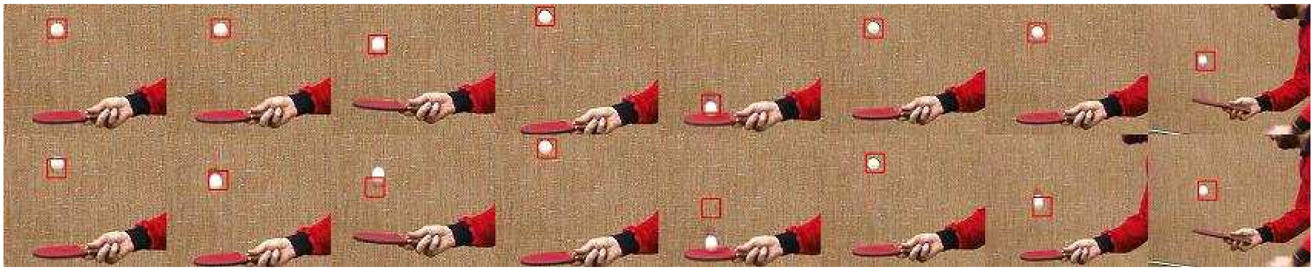


Fig. 5. SPF (top row) vs GPF (bottom row) with 60 particles on Stennis sequence frame 2, 8, 14, 20, 26, 32, 38, and 44.

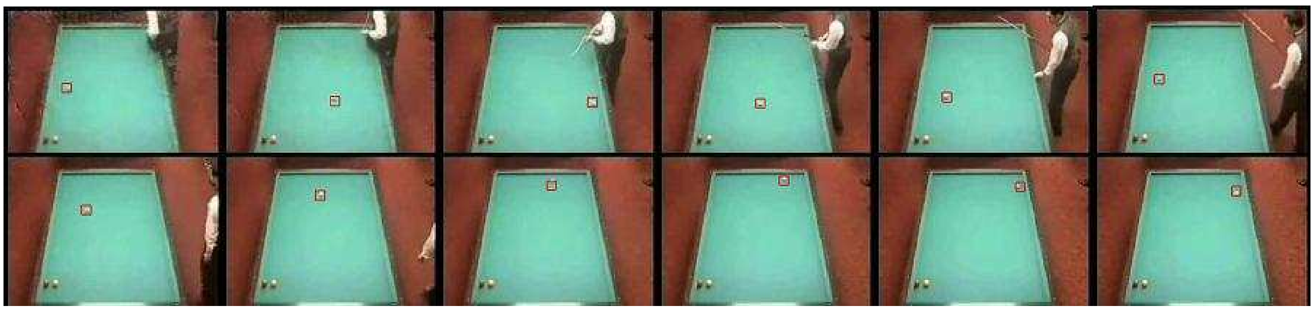


Fig. 6. SPF tracking in Pool Table sequence with 60 particles on frame 2431∼2540 (every 10 frames).

the ball successfully when it moves fast and changes directions abruptly.

In summary, with a smaller initial covariance, GPF is unable to capture abrupt motions, while the tracking errors will inevitably increase with a larger covariance. This prior information is not available in most scenarios. Fortunately, the adaptive SPF succeeds in capturing this nonlinearity by adapting its PD to the current measurement.

5. CONCLUSIONS

This work introduces a new simple and efficient variant of PF, named Sequential Particle Filter, in which particles are proposed sequentially through dynamic adjustment of the proposal density. That is achieved by fully exploiting the likelihood information of the current particle. In specific, the scheme is able to automatically gather particles for a linearly-moving target or to disperse particles to increase search space in a nonlinear case. Comprehensive experiments have demonstrated its superiority over the GPF, especially in terms of efficiency and swift adaptability to nonlinear and abrupt motions.

6. REFERENCES

[1] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters For Tracking Applications*, Artech House, 2004.

[2] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Proc. European Conf. Computer Vision*, Cambridge, UK, 1996, pp. 343–356.

[3] D. Comaniciu and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, pp. 564–577, May 2003.

[4] S. K. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Trans. Image Processing*, vol. 13, no. 11, pp. 1491–1506, 2004.

[5] E. Maggio and A. Cavallaro, "Hybrid particle filter and mean shift tracker with adaptive transition model," in *Proc. of Int. Conf. Acoustics, Speech, and Signal Proc.*, PA,USA, March 2005, pp. 221–224.

[6] Y. Cai, N. De Freitas, and J. J. Little, "Robust visual tracking for multiple targets," in *Proc. European Conf. Computer Vision*, 2006, pp. 107–118.

[7] C. Kreucher, K. Kastella, and A.O. Hero, III, "Multitarget tracking using the joint multitarget probability density," *IEEE Aerosp. Electron. Syst. Mag*, vol. 41, no. 4, pp. 1396–1414, Oct. 2005.

[8] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[9] P. Perez, C. Hue, and J. Vermaak, "Color-based probabilistic tracking," in *Proc. European Conf. Computer Vision*, 2002, pp. 661–675.