VISUAL TARGET TRACKING USING IMPROVED AND COMPUTATIONALLY EFFICIENT PARTICLE FILTERING

Y. Zhai†, *M. Yeary*†, *J.-C. Noyer* ††, *J. Havlicek*†, *S. Nemati*†, *and P. Lanvin* ††

†School of Electrical and Computer Engineering
University of Oklahoma, Norman, OK USA
††Laboratoire d' Analyse des Systèmes du Littoral
Université du Littoral Côte d'Opale, Calais Cedex, France

E-mail: yan.zhai@ieee.org, yeary@ieee.org

ABSTRACT

In this paper, we present a new particle filtering (PF) algorithm for visual target tracking where Galerkin's projection method is used to generate the proposal distribution. Galerkin's method is a numerical approach to approximate the solution of a partial differential equation (PDE). By leveraging this method in concert with L^2 theory and the FFT, we obtain a new proposal which directly approximates the true state posterior distribution and is fundamentally different from various local linearizations or Kalman filter-based proposals. We apply this improved PF algorithm to track a human head in a video sequence. As predicted by theory and demonstrated by our experimental results, this new algorithm is highly effective for tracking targets which exhibit complex kinematics. The new proposal distribution given here captures the high probability area in the state space, thereby gleaning increased support from the true posterior distribution.

***Index Terms*—** TV surveillance systems, nonlinear filters, computer vision

1. INTRODUCTION

The objective of visual target tracking is to estimate the position and velocity of an object given observations acquired from a video sequence. This task is important in many applications including, *e.g.*, surveillance, visual servoing, gestural human-machine interfaces, motion capture, robotics, and many others [1, 2]. Recently, a new tracking algorithm called *particle filtering* has captured international attention because of its exceptional performance in nonlinear and non-Gaussian tracking problems [3–6]. This is in part due to the increasing availability of low cost, high performance computing platforms capable of accomodating the significant complexity of practical particle filtering algorithms. In essence, a particle filter uses a set of weighted samples (called particles) to approximate the distribution of the target motion states based on system measurements. At each iteration, the particles are drawn from the *proposal distribution*, which ideally should derive its support from the true state distribution. Traditionally, the state transition prior is used as the proposal distribution in standard PF algorithms such as, *e.g.*, the *condensation* algorithm [4]. But due to the fact that this proposal does not account for the currently observed data, it is prone to loss of track lock when a target exhibits unpredictable kinematics. Other improvements include the localization based method [2], the unscented particle filter [5], and the state partition based method [7]. However, these methods are based on assumptions of Gaussianity, depend on a linearization process, or both, which can result in nontrivial modelling errors.

In this paper, we propose a new particle filter algorithm in which Galerkin's projection method is used to generate the proposal distribution. The rationale behind Galerkin's method is to assume that the state posterior distribution is in $L^2(\mathbb{R}^n)$. Then, this distribution can be approximated by its projection onto a finite set of orthogonal basis vectors. In addition, by choosing a special set of exponential basis vectors, the projections can be approximated by the computationally efficient FFT. This approach does not require any local linearization of the nonlinear systems and also does not require imposing any Gaussianity assumption on the system state distribution. Thus, it is fundamentally different from the various Kalman filter based PF algorithms such as those in [2, 5, 7].

2. PROBLEM FORMULATION

Target tracking is often formulated as a state estimation problem. Particle filtering is a nonlinear state estimation technique which is based on sequential importance sampling and Bayesian inference. Consider the state space model: $\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \mathbf{w}_{t-1}$ and $\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{v}_t$, where $\mathbf{x}_t$ and $\mathbf{y}_t$ denote the hidden states and the measurements, respectively. Both

This work was supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under grant W911NF-04-1-0221.

$f(\cdot)$ and $h(\cdot)$ could be nonlinear functions, $\mathbf{w}_t$ and $\mathbf{v}_t$ denote the process and measurements noises. Our estimation goal is to evaluate the posterior state distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ which is governed by the Chapman-Kolmogorov equation and the Bayes' formula given as follows [3]:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1} \ , \quad (1)$$

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{\int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})dx_t} \ . \quad (2)$$

In the framework of particle filtering, the state posterior distribution can be approximated by a set of weighted samples (also called *particles*) denoted by $\mathbf{x}^{(i)}$, see [2] for details:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx \sum_{i=1}^{N_s} \tilde{\omega}_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)}) \ , \quad (3)$$

where $\tilde{\omega}_t^{(i)} = \omega_t^{(i)} / \sum_{j=1}^{N_s} \omega_t^{(j)}$ is the normalized importance weight, and $\omega_t^{(i)}$ is given as:

$$\omega_t^{(i)} = \omega_{t-1}^{(i)} \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{q(\mathbf{x}_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})} \ . \quad (4)$$

Equation (3) is actually the discretized posterior distribution. In addition, the distributions $p(\mathbf{y}_t|\mathbf{x}_t^{(i)})$ and $p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})$ represent the system's likelihood and the state transition prior, respectively. At each iteration, the particles (or samples) are drawn from a proposal distribution $\mathbf{x}_t^{(i)} \sim q(\mathbf{x}_t|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{0:t})$, which is effective provided that it has sufficient support from the true posterior distribution. Choosing a good proposal is a crucial and challenging step in designing a particle filter.

3. NEW TRACKING ALGORITHM

As indicated in Section 1, the traditional choice of the proposal cannot provide enough support from the posterior when tracking a target with complex kinematics. This is because the state transition prior does not include the most recent measurement information. In this section, we introduce an improved technique, based on Garlerkin's method, for designing the proposal distribution.

3.1. Galerkin's Method and Computational Efficiency

Galerkin's method is ubiquitous in the solution of PDEs, and in fact forms the basis for the finite element, finite difference, and boundary element methods. Galerkin's method is a discretization procedure that represents the solution in terms of an orthogonal function expansion where each trial function satisfies certain boundary conditions [8]. Let $\mathcal{P}(x,t) = 0$ denotes a PDE, which is a function of the temporal variable t and spatial variable x. Assume $p(x,t) \in L^2(\mathbb{R}^n)$. is the solution of the PDE, such that it can be decomposed according to $p(x,t) = \sum_{l=0}^{\infty} \epsilon_l(t)\phi_l(x)$, where $\{\phi_l(x)\}_{l=0}^{\infty}$ is a complete orthogonal basis for $L^2(\mathbb{R}^n)$ and where $\epsilon_l(t)$ is the projection of $p(x,t)$ onto $\phi_l(x)$ at time t defined by

$$\langle p(x,t), \phi_l(x) \rangle = \int p(x,t)\phi_l(x)^* dx \ . \quad (5)$$

Our objective is to find an approximation $\hat{p}(x,t)$ of $p(x,t)$ such that $\hat{p}(x,t) = \sum_{l=0}^{N-1} c_l(t)\phi_l(x)$.

The approximation error in $\hat{p}$ arises from the use of only $N-1$ as opposed to an infinite number of basis elements ϕ_l. The projections $c_l(t), l = 0, \cdots, N-1$, are the values to be determined. With this setup, we project $\mathcal{P}(x,t)$ onto the subspace $\text{span}\{\phi_l(x)\}_{l=0}^{N-1}$ as: $\langle \mathcal{P}(x,t), \ \phi_l(x) \rangle = 0, \ \ l = 0, \cdots, N-1$. Here, instead of solving the original problem $\mathcal{P}(x,t) = 0$, we solve it's projection, which is a collection of N ordinary differential equations (ODE). Next we apply this method to the nonlinear state estimation problem defined in Section 2.

First we assume $p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) \approx \sum_{l=0}^{N-1} \tilde{c}_l(t)\phi_l$, where $\tilde{c}_l(t)$ will be determined later. for notational simplicity, we drop the variable x. We apply Galerkin's method to equation (2) by projecting it onto the $\text{span}\{\phi_l(x)\}_{l=0}^{N-1}$ as:

$$\begin{aligned}\langle p(\mathbf{x}_t|\mathbf{y}_{1:t}), \phi_k \rangle &= \sum_{l=0}^{N-1} c_l(t) \langle \phi_l, \phi_k \rangle \\ &= \frac{\sum_{l=0}^{N-1} \tilde{c}_l(t) \langle p(\mathbf{y}_t|\mathbf{x}_t)\phi_l, \phi_k \rangle}{\sum_{l=0}^{N-1} \tilde{c}_l(t) \langle p(\mathbf{y}_t|\mathbf{x}_t), \phi_l^* \rangle}\end{aligned} \quad (6)$$

where $k = 0, \cdots, N-1$. Eq. (6) is the "projection version" of the Bayes' formula given in (2). For simplification, (6) can be written in a matrix form as

$$\mathbf{C}(t) = \frac{\mathbf{\Upsilon}_t \tilde{\mathbf{C}}(t)}{\upsilon_t^T \tilde{\mathbf{C}}(t)}, \quad (7)$$

where $\mathbf{\Upsilon}_t$ is an $N \times N$ matrix with the elements $[\mathbf{\Upsilon}_t]_{k,l} = \langle p(\mathbf{y}_t|\mathbf{x}_t)\phi_l, \phi_k \rangle$. The variables $\mathbf{C}(t)$, $\tilde{\mathbf{C}}(t)$ and υ_t are $N \times 1$ vectors, with $[\upsilon_t]_l = \langle p(\mathbf{y}_t|\mathbf{x}_t), \phi_l^* \rangle$. We choose the exponential basis as $\phi_l(x) = \frac{1}{\sqrt{b-a}} \exp\left(j2\pi l \frac{x-a}{b-a}\right)$, where a and b are the integral limits. It was shown in [8] that with this basis the inner product can be approximated with an FFT as

$$\begin{bmatrix} \langle p(x), \phi_0 \rangle \\ \vdots \\ \langle p(x), \phi_{N-1} \rangle \end{bmatrix} \approx \frac{\sqrt{b-a}}{N} \text{FFT}[p(x)]$$

$$\begin{bmatrix} \langle p(x), \phi_0^* \rangle \\ \vdots \\ \langle p(x), \phi_{N-1}^* \rangle \end{bmatrix} \approx \sqrt{b-a} \ \text{IFFT}[p(x)] \ .$$

Then, by using the FFT, (7) can be approximated as

$$[\mathbf{\Upsilon}_t]_l \approx (\sqrt{b-a}/N) \ \text{FFT} \ [p(\mathbf{y}_t|\mathbf{x}_t) \ \phi_l] \quad (8)$$

$$\upsilon_t \approx \sqrt{b-a} \ \text{IFFT} \ [p(\mathbf{y}_t|\mathbf{x}_t)] \ . \quad (9)$$

By applying Galerkin's method to equation (1) in a similar way, $\tilde{c}_l(t)$ can be obtained according to

$$\tilde{c}_l(t) \approx (\sqrt{b-a}/N)\mathrm{IFFT}_l\left[c_l(t-1)\mathrm{FFT}_l[p(\mathbf{x}_t|\mathbf{x}_{t-1})]\right] , \quad (10)$$

where the $\mathrm{FFT}_l[\cdot]$ represents the l^{th} bin of the FFT of the argument. In addition, $\tilde{c}_l(t)$ can also be calculated by

$$\tilde{\mathbf{C}}(t) \approx \mathrm{FFT}\left[\sqrt{b-a}\cdot p(\mathbf{x}_t|\mathbf{x}_{t-1})\mathrm{IFFT}[\mathbf{C}(t-1)]\right] . \quad (11)$$

Moreover, the prediction distribution and posterior distribution can be calculated according to

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) \approx (N/\sqrt{b-a})\ \mathrm{IFFT}[\tilde{\mathbf{C}}(t)] \quad (12)$$

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx (N/\sqrt{b-a})\ \mathrm{IFFT}[\mathbf{C}(t)] . \quad (13)$$

As a summary, in order to approximate the posterior distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, we only need to update the vector $\mathbf{C}(t)$ at each iteration.

3.2. The Improved Particle Filtering

In this section, we incorporate Galerkin's method within the particle filter framework. Firstly, at each iteration, we use $\tilde{\mathbf{C}}(t)$ and $\mathbf{C}(t)$ to approximate the posterior distribution by leveraging the IFFT defined in (12) and (13). Secondly, we draw particles from this approximated distribution and evaluate the particle weights. The final step is the resampling and update stage. Since the proposal is generated by projecting the true posterior distribution onto a subspace of $L^2(\mathbb{R}^n)$, the accuracy of the proposal is guaranteed by appropriately choosing N in (6). Also in this algorithm, evaluating the projection is achieved by implementing FFT and IFFT which is computationally efficient. The detailed algorithm is summarized as follows:

Incorporating Galerkin's method in the PF algorithm :

- Sequential Importance Sampling (SIS) Step:
  - Calculate the parameters $\tilde{\mathbf{C}}(t)$ and $\mathbf{C}(t)$ using (8) to (10) or (11);
  - Construct the proposal distribution using (13);
  - Draw particles for the proposal as:
    $$x_t^{(i)} \sim q(x_t^{(i)}|x_{0:t-1}^{(i)}, y_{1:t}) \approx p(\mathbf{x}_t|\mathbf{y}_{1:t}) \approx (N/\sqrt{b-a})\ \mathrm{IFFT}[\mathbf{C}(t)];$$
  - Evaluate and normalize the importance weights according to (4);
- Resampling Step:Generate a new set of particles $x_t^{i\star}$ from $x_t^{(i)}$ by sampling N_s times the approximate distribution of so that $Pr\left(x_t^{i\star} = x_t^{(j)}\right) = \tilde{\omega}_t^{(j)}$;
- Output and Update Step:Approximate x_t by $\hat{x}_t \approx \frac{1}{N_s}\sum_{i=1}^{N_s} x^{i\star}(t)$ and update the proposal.

4. LABORATORY EXPERIMENTS

As an illustrative example, we apply the improved PF algorithm to track a human in a video sequence by using the following state space model:

$$\begin{bmatrix} r_{t+1} \\ s_{t+1} \\ \dot{r}_{t+1} \\ \dot{s}_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_t \\ s_t \\ \dot{r}_t \\ \dot{s}_t \end{bmatrix} + \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{w}_t ,$$

where $x_t = [r_t\ s_t\ \dot{r}_t\ \dot{s}_t]^T$ is the state vector, and $\mathbf{w}_t$ is a 2×1 the process noise vector. To construct the observation model of this tracking problem, we use a method similar to that proposed in [5], in which the human head is modelled as a moving ellipse. In addition, K equally spaced rays are drawn from the center of the ellipse, which serves as the origin of a local coordinate system (u, v). The intersections of these rays with the ellipse boundary are taken as observations. In local coordinates, the intersections along the k^{th} ray are obtained according to

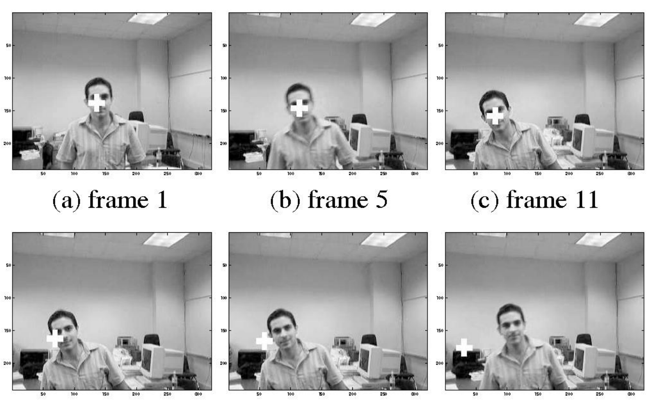
$$u_k = \sqrt{\alpha^2\beta^2/(\beta^2 + \alpha^2\tan{\theta_k}^2)} \quad \text{and}$$
$$v_k = \tan\theta_k \cdot \sqrt{\alpha^2\beta^2/(\beta^2 + \alpha^2\tan{\theta_k}^2)} , \quad (14)$$

by solving the ellipse equation $\frac{(u_k-m)^2}{\alpha^2} + \frac{(v_k-n)^2}{\beta^2} = 1$ and the ray equation $v_k = u_k \tan\phi_k$, where α and β denote the major and minor axes of the ellipse, respectively. Let y represent the observation and convert the local coordinates (u_k, v_k) back to image coordinates. The observation equation is then given by $y_t = [(u_k + r_t, v_k + s_t)] + v_t$, where v_t is the measurement noise. To reject clutter appearing in the video sequence, we apply edge detection along each ray. The edges are labelled as $j = 1 \cdots J_k$, where J_k is the total number of edges detected along the k^{th} ray. The likelihood of the *k-th* ray is $p_k(y_t|x_t) = N_m \sum_{j=1}^{J} \mathcal{N}\left((u_k, v_k), \sigma_{kj}^2\right)$, where N_m is a normalizing factor. The overall likelihood computed across all rays is given by $p\left(y_t|x_t\right) = \prod_{k=1}^{K} p_k\left(y_t|x_t\right)$, which is a simplified version of the likelihood given in [5].

To test the performance of our improved PF algorithm, we choose a video in which the target has relatively complex motions. The digital video was recorded in a standard laboratory environment with a frame size of 240×320 pixels. Both the *condensation* method and the improved PF algorithm were implemented for a comparison. Tracking results for the two filters are shown in Fig. 1 and Fig. 2, where the estimated target centroid is indicated by a white cross. In the first 10 frames the target's motion was nearly rectilinear with a constant velocity and both filters performed well. However from frame 11 to 15, the target begin to have obvious a negative acceleration followed by a positive acceleration. The condensation method (with 100 particles) begin to lose its target around frame 11, and it keeps a constant velocity. Finally, the estimated centroid moves out of the image domain. On the other hand, our improved PF (with 100 particles) keeps tracking the target throughout the whole video. Moreover, the target makes another maneuver around frame 35 to frame 40, and the proposed PF algorithm still keeps a close track. Figure 1 and 2 illustrate one typical realization of our simulations. It is demonstrated that because of the improved proposal distribution, the new proposed PF algorithm is able to yield accurate and robust estimations when tracking a target with complex motions.

5. CONCLUSIONS AND FUTURE WORKS

In this paper, we present an improved particle filter algorithm whose proposal distribution is calculated based on Galerkin's method. The tracking result of the proposed method is promising. Future improvements in several aspects are planned. To cope with complex target kinematics, multiple-model methods will be used to describe the so-called *maneuvering* target. Multiple-model target tracking is often referred to as a jump Markov process, in which the target is assumed to operate according to one model from a finite set of hypothetical models. Multiple-model particle filter (MMPF) has been successfully implemented in radar tracking applications [9]. We will apply a MMPF with a Galerkin-based proposal to visual target tracking applications in the future. In addition, multiple measurement cues will also be used to improve the tracking algorithm.



(a) frame 1 (b) frame 5 (c) frame 11

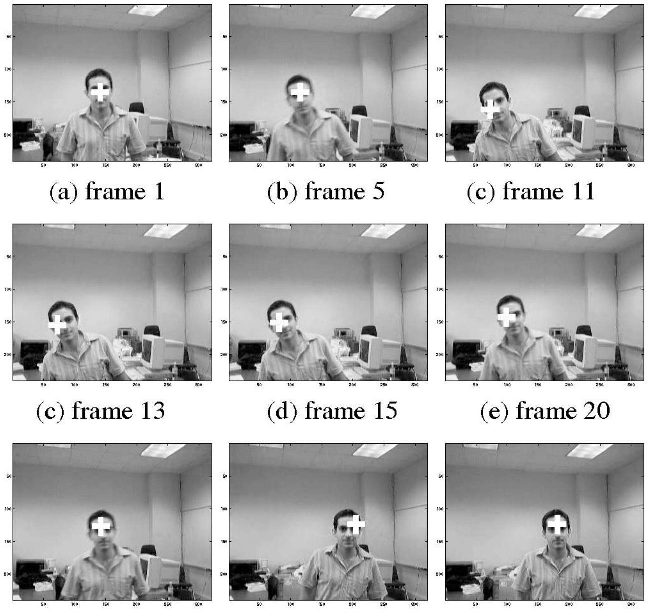
(c) frame 13 (d) frame 15 (e) frame 20

Fig. 1. *Visual tracking by using the condensation method. These frames illustrates a dramatic motion.*

6. REFERENCES

[1] P. Pérez, J. Vermaak and A. Blake, "Data fusion for visual tracking with particles," *Proceedings of IEEE*, vol. 29, no.3, pp. 495-513, 2004.

[2] A. Doucet, and *et. al.*, *Sequential Monte Carlo Methods in Practice*, Springer, New York, 2001.



(a) frame 1 (b) frame 5 (c) frame 11

(c) frame 13 (d) frame 15 (e) frame 20

(f) frame 25 (g) frame 35 (h) frame 45

Fig. 2. *Visual tracking using the improved PF. Even under several maneuvers, tracking is maintained.*

[3] M. Arulampalam, and *et. al.*, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. on Sig. Proc.* pp. 174-188, vol. 50, No. 2, Feb., 2002.

[4] M. Isard and A. Blake, "Visual tracking by stochastic propagation of conditional density," *Proc. 4th Euro. Conf. Computer Vision*, pp., 343-356, Apr., 1996.

[5] Y. Rui and Y. Chen, "Better proposal distributions: object tracking using unscented particle filter," *Proc. of IEEE CVPR*, pp. 786-793, Dec., 2001.

[6] C. Chang and R. Ansari, "Kernel particle filter for visual tracking," *IEEE Signal Processing Letters*, vol. 12, no. 3, pp. 242-245, Mar., 2005.

[7] Y. Zhai, M. Yeary and *et. al.*, "Visual Tracking Using Sequential Importance Sampling with a State Partition Technique," *Proc. of IEEE ICIP*, pp., 876-879, Sep., 2005.

[8] J. Gunther, R. Beard and *et. al.*, "Fast nonlinear filtering via Galerkin's method," *American Control Conference (ACC)*, pp. 2815-2819, 1997.

[9] S. McGinnity and G. Irwin, "Multiple model bootstrap filter for maneuvering target tracking," *IEEE Trans. Aerospace and Electronic Systems*, vol. 36, no. 3, pp. 1006-1012, 2000.