POWER-AWARE PARTICLE FILTERING FOR VIDEO TRACKING

Pan Pan and Dan Schonfeld

ECE department, University of Illinois at Chicago
851 S. Morgan St., Chicago, IL, 60607
{ppan3, dans}@uic.edu

ABSTRACT

This paper presents a novel approach to particle filtering which minimizes the total tracking distortion by considering dynamic variance of proposal density and adaptive number of particles for each frame. Traditionally, particle filters use fixed variance of proposal density and fixed number of particles per frame. We first propose the tracking distortion measurement and then obtain the optimal particle number and memory size allocation equations under two different constraints. After that, the optimal particle number allocation equation is demonstrated in one-dimensional and two-dimensional object tracking. Experimental results show the improved performance of our power-aware particle filters in comparison to traditional particle filters. At last, we give the complete algorithm for real application and show the better performance. To the best of our knowledge, this paper is the first to consider the variant numbers of particles for each frame.

1. INTRODUCTION

Over the past few years, particle filters have gained popularity in object tracking. The number of particles used is an essential index of the CPU time and tracking power, which are critical resources. By utilizing power efficiently, we can extend the lifetime of battery, decrease the interference and make the whole system more stable.

When we use particle filters for tracking, each sample is assigned a weight in order to get the posterior density functions. It can be shown that if the number of samples is sufficiently large, the sample approximation of the posterior density can be made arbitrarily accurate[1]. However, because of the limited power, only a finite number of particles could be used in practice.

In traditional particle filters, the variance of proposal density and the number of particles per frame for are fixed during the whole tracking process. These parameters have to be set by experience before tracking. However, this does not consider the different characteristics of each frame. In some frames the objects moves fast, while in others moves slowly. In the meanwhile, when the power is limited, the number of particles should be allocated wisely to get the best tracking quality. Our goal is to minimize the total tracking distortion while maintaining the same number of particles over a video sequence as traditional particle filters.

2. TRACKING DISTORTION MEASUREMENT

2.1. Particle Filter Theory

For Bayesian tracking, we assume the states form a first-order Markov chain. Since the state X_k is hidden and can only be estimated by the observations Z_k, the propagation rule is given by [2]

$$p(X_k|Z_{1:k}) \propto p(Z_k|X_k)p(X_k|Z_{1:k-1}), \tag{1}$$

where

$$p(X_k|Z_{1:k-1}) = \int_{X_{k-1}} p(X_k|X_{k-1})p(X_{k-1}|Z_{1:k-1})dX_{k-1}. \tag{2}$$

Let X_k^i denote the i^{th} sample at time k and $\{X^i, i \in \mathbf{N}\}$ be a sample set generated from a proposal density $q(X)$. The normalized weights $\pi(X^i)$ are then given by

$$\pi(X^i) = \frac{p(X^i)}{q(X^i)}, \tag{3}$$

$$\text{where } \pi(X^i) = \frac{w^i}{\sum_{j=1}^{n} w^j}. \tag{4}$$

w^i is the weight of i^{th} sample measured by weight function $w(x)$. The estimate state is then given by the sample mean

$$\widehat{X}_k = \sum_{i=1}^{n} \pi(X_k^i)X_k^i. \tag{5}$$

2.2. Tracking Distortion Measurement

In real tracking, the tracking error of i^{th} particle $\widetilde{X}^i$ is defined as the difference between the real state vector R and sample state vector X^i, i.e. $\widetilde{X}^i = R - X^i$. Since the tracking error is random, we use the variance of the weighted average error brought by n particles as a measurement of total tracking distortion.

Assume tracking vector has N dimensions and each component of error vector is independent from others, it can be shown that tracking distortion of k^{th} frame D_k is as follows

$$D_k = \frac{\sigma_k^2 \varepsilon_k}{n_k^N} \quad \text{for large } n_k/\widetilde{X}_{max}, \tag{6}$$

where n_k is the number of particles of k^{th} frame.

$$\text{and} \quad \sigma_k^2 = \prod_{j=1}^{N} \sigma^2_{\widetilde{X}^{j,k}}, \;\; \varepsilon_k = \prod_{j=1}^{N} \varepsilon_{\widetilde{X}^{j,k}}, \tag{7}$$

where $\sigma^2_{\widetilde{X}^{j,k}}$ is the variance of the tracking error of j^{th} component at k^{th} frame and

$$\varepsilon_{\widetilde{X}^j} = \frac{2\widetilde{X}^j_{max} \int_{-\widetilde{X}^j_{max}}^{\widetilde{X}^j_{max}} w^2(\widetilde{X}^j)\beta(\widetilde{X}^j)d\widetilde{X}^j}{(\int_{-\widetilde{X}^j_{max}}^{\widetilde{X}^j_{max}} w(\widetilde{X}^j)\beta(\widetilde{X}^j)d\widetilde{X}^j)^2}, \tag{8}$$

where $\widetilde{X}_{max}$ is the bound of tracking error and β describes the interval density of the errors of n particles.

Given the fixed σ_k^2 and ε_k, the more particles we use, the small tracking distortion D_k. When n reaches infinity, D_k converges to zero and the tracking is quite accurate. This consists with the property of Bayesian importance sampling[1], which verifies our tracking distortion measurement is effective.

3. OPTIMAL PARTICLE NUMBER / MEMORY SIZE ALLOCATION EQUATIONS

We derive the optimal allocation equations for video tracking under two different constraints: (1) Constraint on the average number of particles n and (2) Constraint on the average memory size of the particle indexing tables R. For example, if we have 8 particles, i.e. n=8, then we need 3 indexing bits, i.e. R=3, where (000) represents particle 1, (001) represents particle 2, ..., (111) represents particle 8. The relationship between n and R is $R = \log_2 n$.

3.1. Constraint on the average number of particles

Given the desired average number n over M frames, the question now is how to allocate the total nM particles among M frames so that the total distortion is minimized.

Hence, we want to minimize

$$D_T = \frac{1}{M}\sum_{k\in M} \frac{\sigma_k^2 \varepsilon_k}{n_k^N} \quad \text{such that} \quad \sum_{k=1}^{M} n_k = nM. \tag{9}$$

we obtain

$$n_k = n\frac{\sqrt[N+1]{N\sigma_k^2\varepsilon_k}M}{\sum_{k=1}^{M} \sqrt[N+1]{N\sigma_k^2\varepsilon_k}}, \tag{10}$$

$$R_k = R + log_2(\frac{\sqrt[N+1]{N\sigma_k^2\varepsilon_k}M}{\sum_{k=1}^{M} \sqrt[N+1]{N\sigma_k^2\varepsilon_k}}). \tag{11}$$

The above formulas (10) and (11) imply that when the total amount of particles is fixed, a frame with larger variance of error should be given more particles for tracking, while a frame with a smaller variance of error should be given fewer particles. This complies with the common sense.

With this particle allocation, the distortion of each frame is

$$D_k = \frac{\sqrt[N+1]{\sigma_k^2\varepsilon_k}}{n^N M^N N^{\frac{N}{N+1}}}(\sum_{k=1}^{M} \sqrt[N+1]{N\sigma_k^2\varepsilon_k})^N. \tag{12}$$

3.2. Constraint on the average memory size

Under the constraint on the average memory size of the particle indexing tables, we want to minimize

$$D_T = \frac{1}{M}\sum_{k=1}^{M} \sigma_k^2\varepsilon_k 2^{-NR_k} \quad \text{such that} \quad \sum_{k=1}^{M} R_k = RM. \tag{13}$$

We obtain

$$R_k = R + \frac{1}{N}log_2\frac{\sigma_k^2\varepsilon_k}{(\prod_{k=1}^{M}\sigma_k^2\varepsilon_k)^{1/M}}, \tag{14}$$

$$n_k = n(\frac{\sigma_k^2\varepsilon_k}{(\prod_{k=1}^{M}\sigma^2\varepsilon_k)^{1/M}})^{1/N}. \tag{15}$$

The above formulas (14) and (15) imply that a frame with larger variance of error should be given more memory sizes of the particle indexing tables, while a frame with a smaller variance of error should be given fewer memory sizes.

The optimal allocation under constraint R is such that all the frames have the same tracking distortion, which is,

$$D_k = (\prod_{k=1}^{M}\sigma_k^2\varepsilon_k)^{1/M}2^{-NR} = (\prod_{k=1}^{M}\sigma_k^2\varepsilon_k)^{1/M}n^{-N}. \tag{16}$$

3.3. Analysis and Discussion

In real tracking, we have the following sampling scheme[1],

$$X_k = f(X_{k-1}) + v_k, \tag{17}$$

where $f(X_{k-1})$ is the estimation of the mean of the new samples, and v_k has the Gaussian distribution $v_k \sim N(0, \sum G)$.

According to equation (10), given the tracking dimension N and the average number of particles used among M frames n, the allocation of number of particles depends on σ_k and ε_k. It can be shown that the variance of error is also the variance of proposal density for sampling. As is shown in (8), ε_k is difficult to compute. However, we can take it independent of k when reasonably assuming $\beta(\widetilde{X})$, $\widetilde{X}_{max}$ and $w(\widetilde{X})$ are time independent. As a result, the number of particles for k^{th} frame corresponds only with the variance of proposal density σ_k^2. So, (10) becomes

$$n_k = n \frac{\sqrt[N+1]{N\sigma_k^2}M}{\sum_{k=1}^{M} \sqrt[N+1]{N\sigma_k^2}}. \quad (18)$$

We call the variance of proposal density as proposal variance. In traditional tracking, the proposal variance of all frames are the same, according to (18), $n_k = n$, which means the optimal particle allocation is using fixed number of particles for each frame. However, this does not consider the different characteristics of each frame. In some frames the objects moves fast, while in others moves slowly. The proposal variance should be bigger for fast moving frames because of the more movement, and the number of particles should also be more because of the larger variance.

So, we introduce the dynamic variance of v_k, and

$$X_k \sim N(f(X_{k-1}), \textstyle\sum_{G(\hat{X}_{k+1}-\hat{X}_k)}), \quad (19)$$

which means the variance of the proposal density is changing with $\Delta\widehat{X}_k = \widehat{X}_{k+1} - \widehat{X}_k$. Then the number of particles will be allocated according to different proposal variance.

4. EXPERIMENTS AND COMPARISONS

We first show the power of our equation by condensation particle filter based on off-line learning of $\Delta\widehat{X}_k$, and then give the complete algorithm for real application in the next section. We compare the tracking results of different algorithms, which are (a) Fixed proposal variance (choosing smallest variance of $\sum_{G(\Delta\hat{X}_k)}$), Fixed number of particles; (b) Fixed proposal variance (choosing biggest variance of $\sum_{G(\Delta\hat{X}_k)}$), Fixed number of particles; (c) Variant proposal variance, Fixed number of particles; (d) Variant proposal variance, Variant number of particles.

4.1. One-dimensional tracking

In this experiment, the average number of particles $n = 12$. And the proposal variance $\sigma^2 = 0.02$ if $|\Delta\widehat{X}_k| \leq 0.02$; $\sigma^2 = 3$ otherwise. The tracking results are given in Fig 1. The solid line represents the ground truth, while the dotted line denotes the tracking results. It can be shown that by using our equation, the tracking quality improves a lot while utilizing about the same CPU time(obtained by Matlab 6.5, CPU Pentium M 1.6GHZ). The average SNR in Table 1 is got by 200 trials.

4.2. Two-dimensional tracking

In two dimensional tracking, the tracking vector is comprised of the x, y coordinates of the center of the object. This experiment is carried out on 200 synthetic frames with average particle number n=9. The proposal variance of x, y component

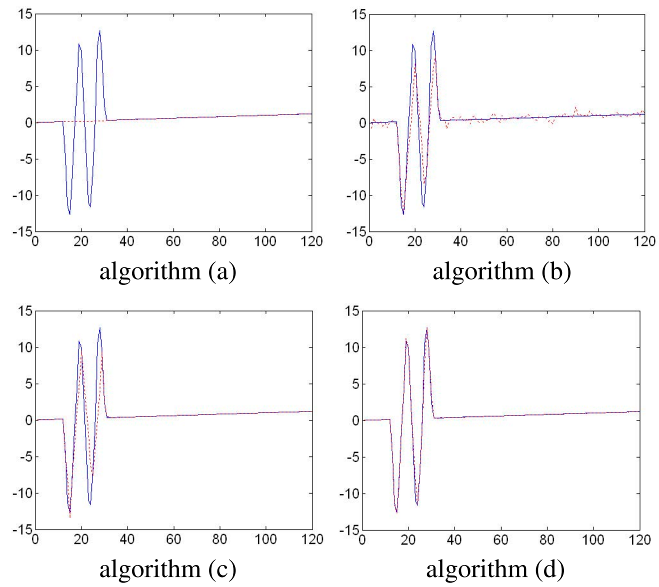

Fig. 1. 1D tracking results of different algorithms.

Table 1. $\overline{SNR}$ and CPU time in seconds of 1D tracking

Algorithm	a	b	c	d
$\overline{SNR}$	1.0441	4.3206	4.6872	37.1755
CPU time	0.07	0.07	0.07	0.07

$\sigma_x^2 = \sigma_y^2 = 0.02$ if the absolute value of either component of $\Delta\widehat{X}_k \leq 0.05$; $\sigma_x^2 = \sigma_y^2 = 2$ otherwise. Tracking results of different algorithms are shown in Fig 2. The CPU time of processing one frame of different algorithms is about 0.97 seconds. It can be seen that algorithm (a) (b) (c) completely lose tracking by frame 25, only (d) tracks well.

5. DYNAMIC PROPOSAL VARIANCE AND OPTIMAL PARTICLE NUMBER ALLOCATION ALGORITHM

Now, we demonstrate our dynamic proposal variance and optimal particle number allocation algorithm. We process 20 frames at one time. Since the time span of each frame is so little, we can still consider it real-time processing.

1. Use ABM [3] as a motion estimation to obtain $\Delta\widehat{X}_k$ of the 20 frames;

2. Distribute the proposal variance $\sum_{G(\Delta\hat{X}_k)}$;

3. Use the optimal particle number allocation equation among 20 frames to get the optimal particle number for each frame n_k;

4. Turn back to do particle filter estimation with the optimal number of particles. The sampling scheme is $X_k^i = X_{k-1}^i + \Delta\widehat{X}_k + v_k \quad i = 1, 2, ..., n_k$;

5. After processing these 20 frames, continue to process

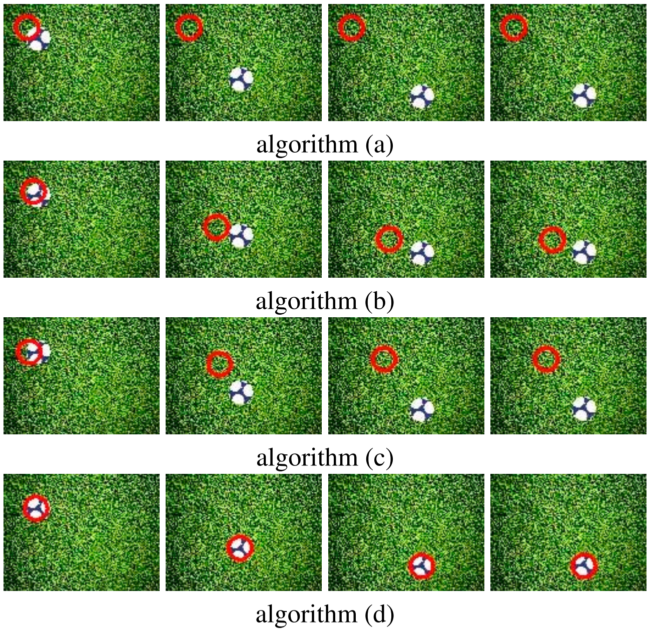

Fig. 2. Frame 14, 25, 112, 200 of 2D tracking results

Table 2. $\overline{MSE}$ and CPU time per frame in seconds of complete algorithm application

Algorithm	a	b	c	d
$\overline{MSE}$	3.8224	3.7449	2.5441	1.6222
CPU time	1.5778	1.5758	1.5853	1.5882

another 20 frames till all of the video sequences have been processed.

We apply our algorithm to a video sequence to see the improved performance. Besides the variance detector, ABM motion is also used to make the mean of the estimate accurate, so the improved performance is shown by mean-square error. The average number of particle n=10. The variance of x, y component $\sigma_x^2 = \sigma_y^2 = 0.3$ if the absolute value of either component of $\Delta\widehat{X}_k \leq 1$; $\sigma_x^2 = \sigma_y^2 = 3$ otherwise. The person in the video walks slowly and then suddenly bends and jumps. The ground truth is obtained by circling the head by hand frame by frame. Since it is not easy to get the ground truth of all frames precisely, we only consider the 20 frames which contains bending and jumping. Fig 3 shows the tracking results of different algorithms. Table 2 shows the average MSE got by 100 trials and CPU time for processing a frame.

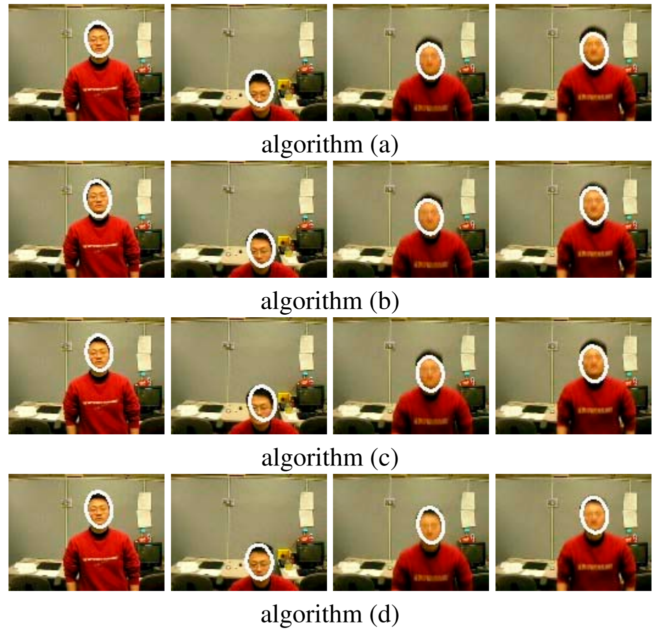

Fig. 3. Frame 142, 154, 158, 159 of tracking results of complete algorithm application.

6. CONCLUSION

In this paper, we propose to minimize the total tracking distortion by considering the variant proposal variance and number of particles per frame at the same time. By using the optimal particle number allocation equation, we propose a complete algorithm for real tracking. Our algorithm has the following advantages: 1. It can minimize the total tracking distortion while using the same total number of particles as traditional particle filters; 2. Given the same power, our algorithm achieves the best tracking quality; 3. For the same tracking quality, our algorithm uses the least CPU time and least power.

7. ACKNOWLEDGEMENT

We would like to thank our labmate Nidhal Bouaynaya for helping with the implementation of the particle filtering codes.

8. REFERENCES

[1] A. Doucet, "On sequential simulation-based methods for bayesian filtering," Tech. Rep., Signal Processing Group, Department of Engineering, University of Cambridge, 1998.

[2] M. Isard and A. Blake, "Condensation conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[3] N. Bouaynaya and D. Schonfeld, "Complete system for head tracking using motion-based particle filter and randomly perturbed active contour," in *Proceedings of SPIE,Image and Video Communications and Processing*, March 2005, vol. 5685, pp. 864–873.