The Fifteenth International Symposium on Artificial Life and Robotics 2010 (AROB 15th '10),
B-Con Plaza, Beppu,Oita, Japan, February 4-6, 2010

An improvement of MSEPF for visual tracking

Yuki Nakagama, Masahiro Yokomichi
University of Miyazaki, 1-1,Gakuenkibanadainishi,Miyazaki,Miyazaki,Japan
(0985-58-7420)
(tge813u@student.miyazaki-u.ac.jp)

Abstract: Recently, many approaches on applying particle filter to visual tracking problem have been proposed. However, it is hard to implement it to the real-time system because it requires a lot of computation and resources in order to achieve higher accuracy. As a method for reduce the computation time, Shan and coworkers proposed combining particle filter and Mean-Shift in order to keep the accuracy with small number of particles. In their approach, the state of each particle moves to the point in the window with the highest likelihood value. It is known that the accuracy of estimation depends on the size of the window, but the larger window size make the computation slower. In this paper, the authors propose method for exploring the highest likelihood more quickly by means of random sampling. Moreover, our approach defines likelihood in terms of not only color cue but also motion cue for higher-accuracy object tracking. The effectiveness of the proposed method is evaluated by real image sequence experiments.

Keywords: Particle filter; Real-time visual tracking; MSEPF

I. Introduction

Visual tracking is one of the important researches in the autonomous mobile robot. Visual tracking requires high-accuracy tracking and real-time processing. To achieve high accuracy tracking, many approaches have been studied. . Particle filter is one of the robust tracking approaches in visual tracking, which has recently been developed. It performs a random search guided by a stochastic motion model to obtain an estimate of the posterior distribution describing the object's configuration. However, it is known that the degeneracy is one of the difficult problems inherent in particle filter. Degeneration problem is a phenomenon of the tracking accuracy's decreasing because most particles may have very low likelihood. One of approaches that deal with it is to use very large number of particles, but it is hard to implement it to real-time systems because it requires a lot of computation times and resources. Shan and coworkers proposed combining particle filter and Mean-Shift (MSEPF) in order to keep the accuracy with small number of particles. In their approach, the state of each particle moves to the point in the window with the highest likelihood value. In general, MSEPF overcomes the degeneration problem because each particle has higher likelihood. In addition, the accuracy of estimation depends on the size of the window, but the larger window size makes the computation slower.

In this paper, we propose a method for exploring the highest likelihood more quickly by means of random sampling. Using random sampling in Mean-Shift, computing time will be considerably reduced keeping accuracy. When total of likelihoods is less than threshold, particles do relocation using extended reset method. Moreover, likelihood function is also modified such that the computing time reduces. In general, visual tracking uses only color cue to define the likelihood function. In the proposed approach, the likelihood function is defined in terms of not only color cue but also motion cue for higher-accuracy object tracking.

II. The Mean Shift Embedded Particle Filter

1. Particle Filter

Particle filter is model estimation technique based on simulation. A continuous state vector of a target object at time step t is denoted by x_t. Dynamic model is assumed to be represented as a temporal Markov chain:

$$p(\mathbf{x}_t \mid \mathbf{x}_1,\ldots,\mathbf{x}_t) = p(\mathbf{x}_t \mid \mathbf{x}_{t-1}), \tag{1}$$

and

$$p(\mathbf{z}_1,\ldots,\mathbf{z}_t \mid \mathbf{x}_1,\ldots,\mathbf{x}_t) = \prod_{i=1}^{t} p(\mathbf{z}_i \mid \mathbf{x}_i). \tag{2}$$

Particle filter aims to estimate the sequence of hidden parameters $\mathbf{x}_t$ based only on the observed data $\{\mathbf{z}_1,\ldots,\mathbf{z}_t\}$. According to the Bayes rule, the prior density is then given by

$$p(\mathbf{x}_t \mid \mathbf{z}_1,\ldots,\mathbf{z}_{t-1}) = \int p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} \mid \mathbf{z}_1,\ldots,\mathbf{z}_{t-1}) dx_{t-1} \tag{3}$$

and posterior is given by

$$p(\mathbf{x}_t \mid \mathbf{z}_1,\ldots,\mathbf{z}_t) = k_t p(\mathbf{z}_t \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{z}_1,\ldots,\mathbf{z}_{t-1}), \tag{4}$$

where k_t is the normalization term.

In the particle filter, by using a set of samples and the corresponding weights $\{(\mathbf{s}_t^n, \pi_t^n\}_{n=1}^N$ at time step t (where N is the number of particles), the posterior is approximated as

$$p(\mathbf{x}_t \mid \mathbf{z}_{1,\ldots}, \mathbf{z}_t) \approx \sum_{n=1}^{N} \pi_t^n \delta(\mathbf{x}_t - \mathbf{s}_t^n) \tag{5}$$

where $\delta(\mathbf{x}_t - \mathbf{s}_t^n)$ is the Dirac's delta function. Then, the prior is approximated as

$$p(\mathbf{x}_t \mid \mathbf{z}_{1,\ldots}, \mathbf{z}_{t-1}) \approx \sum_{n=1}^{N} \pi_{t-1}^n p(\mathbf{x}_t \mid \mathbf{s}_{t-1}^n). \tag{6}$$

The weights π_t^n is determined such $\pi_t^n \propto p(\mathbf{z}_t \mid \mathbf{x}_t^n)$, $\sum_{n=1}^{N} \pi_t^n = 1$. If a sufficiently large number of particles can be prepared, Eq. (5) and (6) are accurate. In reality, however, using an infinite number of particles is not allowed, especially for real-time processing.

2. Mean-Shift

The mean shift algorithm is a non-parametric technique that climbs the gradient of a probability distribution to find the nearest dominant mode (peak). In the search window, the mean position of the target object is computed and the search window is centered at that position. The mean position can be obtained as follows,

$$x_c = \frac{M_{10}}{M_{00}}, y_c = \frac{M_{01}}{M_{00}}, \tag{7}$$

where

$$M_{00} = \sum_x \sum_y I(x,y),$$
$$M_{10} = \sum_x \sum_y xI(x,y),$$
$$M_{01} = \sum_x \sum_y yI(x,y).$$

Here, (x_c, y_c), $I(x,y)$ denote the mean search window position, pixel (probability) in the position (x,y) in the image. Position of the target object is tracked by iterating this mean position calculation until shift length converges.

3. MSEPF

MSEPF is novel algorithm that incorporates Mean-Shift into particle filter. In MSEPF, Mean-Shift analysis is applied to each particle based on observation density, after each particle was measured by likelihood function. MSEPF can keep the accuracy using fewer particles because particles converge to the local maximum. Therefore, MSEPF can reduce particles than particle filter. Figure1 shows a graphical representation of the MSEPF. It is known that the accuracy of estimation depends on the size of the window. However, the larger window size requires additional computation time.

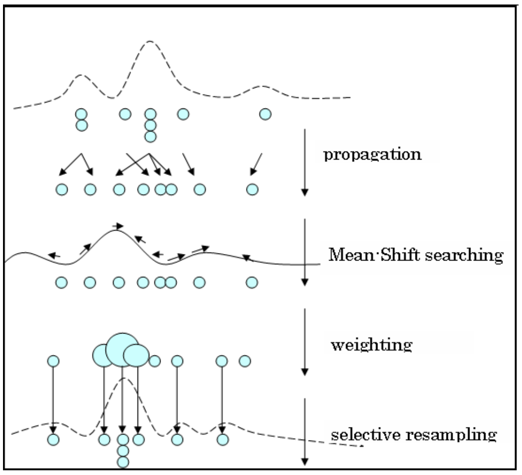


Fig.1. a graphical representation of MSEPF

III. An improvement of MSEPF and the observation model

1. An improvement of MSEPF

As described in the previous section, the accuracy of estimation depends on the size of the window, but the larger window size requires additional computation time. As pointed out by Yang and Duraiswami [3], the computational complexity of traditional Mean-Shift tracking is quadratic in the number of samples, making real-time performance difficult. To deal with their problem, it was proposed new algorithm using random sampling in Mean-Shift. In addition, particle filter embedded it. The approach can define more complex likelihood function because the computational complexity is considerably reduced. To summarize, algorithm of proposed MSEPF was outlined in Fig.2.

Given particle set $S_{t-1} = \{s_{t-1}^i, \pi_{t-1}^i \mid i = i, \cdots N\}$ at time $t-1$, perfume the following step:

1. Propagation: propagate each particle by dynamic mo del to obtain the sample set $\tilde{S}_t$
2. Mean-Shift searching: apply Mean-Shift that used random sampling to each particle
3. Weighting: weight each particle using likelihood functio n
4. Resampling: obtain new particles S_t
5. Estimate: obtain estimate state using ohtsu's method [4]

Fig.2 An iteration step of proposed MSEPF

2. Dynamic model

The state transition model characterizes the motion change between frames. In a visual tracking problem, it is ideal to have an exact motion model governing the kinematics of the object. In practice, however, it is hard to get at accurate state transition model. Moreover, object motion is irregular movement. Therefore, our approach approximates state transition model by simple model

$$\begin{aligned} x_t &= x_{t-1} + w_x \\ y_t &= y_{t-1} + w_y \end{aligned} \tag{8}$$

where w_x and w_y are each uniform pseudorandom number.

3 Likelihood function

Particle filter requires a likelihood function of the target object. In general, visual tracking apply color cue to likelihood function. Our approach uses not only color cue but also motion cue. Our approach takes the statistic of chromatins of target object in advance. Here, likelihood of color cue $\pi_c(x,y)$ at x is defined as

$$\pi_c = \frac{1}{2\pi\sigma_r} e^{-\frac{(\mu_r - R(x,y))}{\sigma_r^2}} + \frac{1}{2\pi\sigma_g} e^{-\frac{(\mu_g - G(x,y))}{\sigma_g^2}} + \frac{1}{2\pi\sigma_b} e^{-\frac{(\mu_b - B(x,y))}{\sigma_b^2}} \quad . \qquad (9)$$

Here, μ_r, μ_g, μ_b, σ_r, σ_g and σ_b are red, green, and blue components of color saturation of mean or variance, respectively. In addition, $R(x,y)$, $G(x,y)$ and $B(x,y)$ are defined

$$R(x,y) = \frac{r(x,y)}{r(x,y)+g(x,y)+b(x,y)}$$

$$G(x,y) = \frac{g(x,y)}{r(x,y)+g(x,y)+b(x,y)}$$

$$B(x,y) = \frac{b(x,y)}{r(x,y)+g(x,y)+b(x,y)},$$

where $r(x,y)$, $g(x,y)$ and $b(x,y)$ are respectively intensity in the position (x,y).

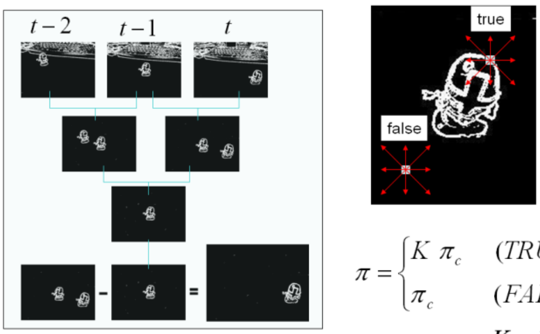
Next, likelihood used motion cue was outlined below. First, edge images were extracted from real image sequences. Second, only moving object were extracted using emporal differecing. If particles exist near edge, likelihood of samples are defined

$$\pi(x,y) = k\pi_c$$

where, k is invariable ($k \geq 1{,}0$). If it does not exist, there are defined

$$\pi(x,y) = \pi_c .$$

Figure.3 shows a graphical representation of the likelihood function.



$$\pi = \begin{cases} K\ \pi_c & (TRUE) \\ \pi_c & (FALSE) \end{cases}$$

$$K > 1.0$$

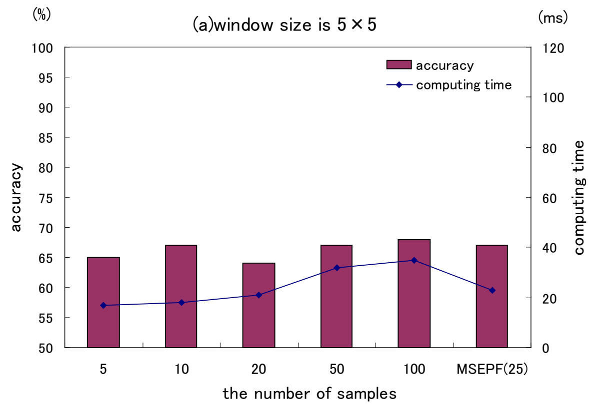
Fig.3. a graphical representation of our likelihood function

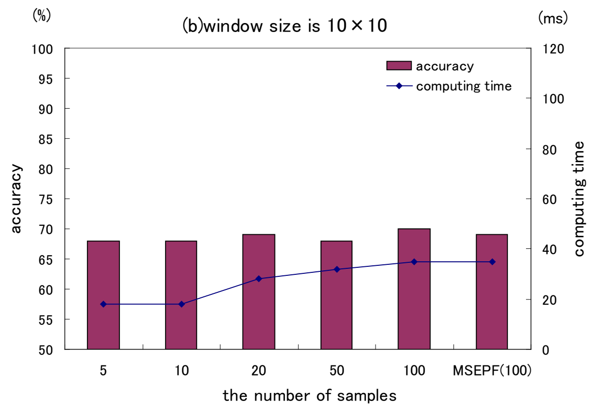
IV. Experiment

Aim of this section is evaluate proposed MSEPF. To illustrate the differences between conventional MSEPF and proposed MSEPF, same sequences ware applied. These video sequences are captured at 14 frames per second in a usual office environment with background clutters. The resolution of each image is 240×180 pixels, window size in Mean-Shift are 4 patterns of 5×5 pixels, 10×10 pixels, 15×15 pixels and 20×20 pixels. The numbers of samples change 5, 10, 20, 50, and 100 in our approach. The numbers of particles are 100 in all algorithms.

Figure.4 shows that computation time and tracking accuracy of conventional MSEPF and our MSEPF. Experimental results demonstrate that our algorithm outperforms conventional MSEPF. It understands that computing time greatly decreases as the number of samples increases. When window size is 20×20 pixels, conventional MSEPF spent 103 ms on average for each frame. Our proposed MSEPF of samples 10 when window size is 20×20 pixels, spent 33 ms on average for each frame, it is about one-third. Moreover, this experiment obtained the surprising result proposed MSEPF requires only very few samples. High accuracy tracking as equal as conventional MSEPF can be achieved with 10 less or equal random samples.

In addition, next experiment was performed to evaluate likelihood function. As discussed in preceding section, likelihood function was applied color cue and motion cue to. Some tracking results of two trackers are shorn Fig.5. In this experiment, computation speeds of each tracker are almost equal. As it can be seen in Fig.5 (a), tracker that uses likelihood function only of color cue tracks motionless object that is similar color of target object. The trackers with our proposed likelihood can track as shown in Figure.5 (b). This likelihood function will be able to track more effective and robust tracking than observation model only of color cue because likelihood make high when particle exist near motion object. Table.1 shows that tracking results of conventional MSEPF with likelihood function based on only color cue and our proposed MSEPF with likelihood function based on our observation model. Computing speed of each tracker is almost equal. Despite almost equal computation speed, our proposed method is dramatically highly accurate tracking than conventional method.





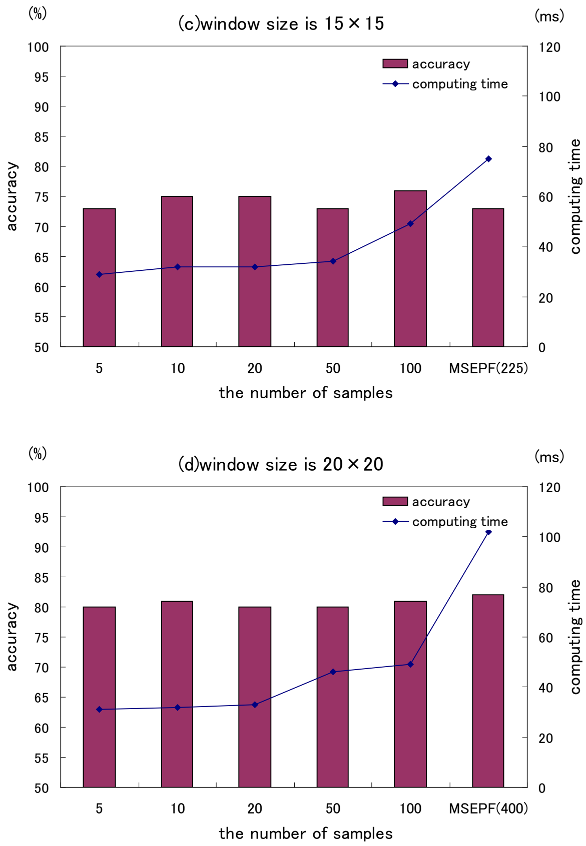


Fig.4. Computation time and tracking accuracy of conventional MSEPF and proposed MSEPF. (a), (b), (c), and (d) are window size of 5×5 pixels, 10×10 pixels, 15×15pixels and 20×20 pixels.

(a) Tracking using likelihood function based on only color cue.

(b) Tracking using likelihood function based on color and motion

Figure.5 Tracking using observation based on color and motion cues and based on color and motion cues. The frames 250, 260, and 270 are shown.

Table.1. tracking results of conventional MSEPF with likelihood function based on only color cue and our proposed MSEPF with likelihood function based on color and motion cue.

	accuracy	computing time
Conventional MSEPF	59.3%	36ms
Proposed MSEPF	75.1%	32ms

V. Conclusion

This article presented about improvement of MSEPF for visual tracking. Mean shift that used random sampling embedded particle filter can quickly track than conventional MSEPF. Using as few as even 10 random samples, our proposed MSEPF is almost equal to conventional MSEPF. Using color and motion cues in likelihood function our method can track accurately and robustly than observation model only of color cue. By incorporating accurate dynamic model, the performance of tracking will be improved greatly. We will apply EM algorithm to learn dynamic model, in our future work.

REFERENCES

[1] Caifeng Shan, et al.(2004), Real Time Hand Tracking by Combining Particle Filtering and Mean Shift, Automatic Faces and Gesture Recognition:669-674

[2]Ryuichi Ueda, at el. (2005), Recovery Methods for Fatal Estimation Errors on Monte Carlo Localization. JERC 23(4):466-473

[3] Changjiang Yang, et al. (2005), Efficient Mean-Shift Tracking via a New Similarity Measure(1):176-183

[4] N. Otsu (1995), Discriminant and Latest Squares Threshold
Selecton. *Proc of 4IJPCPR* 1978, pp 592-596.

[5] G. Kitagawa (1996), Monte Carlo filter and smoother for non-Gaussian nonlinear state space models, Journal of Computational and Graphical Statistics 5(1):1–25.

[6] T. Bando et al (2006), Switching particle filters for efficient visual tracking, Robotics and Autonomous Systems 54:873–884