Joint tracking algorithm using particle filter and mean shift with target model updating

Bo Zhang (张 波), Weifeng Tian (田蔚风), and Zhihua Jin (金志华)

Laboratory of Navigation and control, Department of Instrumentation Engineering, Shanghai Jiao Tong University, Shanghai 200240

Received May 23, 2006

Roughly, visual tracking algorithms can be divided into two main classes: deterministic tracking and stochastic tracking. Mean shift and particle filter are their typical representatives, respectively. Recently, a hybrid tracker, seamlessly integrating the respective advantages of mean shift and particle filter (MSPF) has achieved impressive success in robust tracking. The pivot of MSPF is to sample fewer particles using particle filter and then those particles are shifted to their respective local maximum of target searching space by mean shift. MSPF not only can greatly reduce the number of particles that particle filter required, but can remedy the deficiency of mean shift. Unfortunately, due to its inherent principle, MSPF is restricted to those applications with little changes of the target model. To make MSPF more flexible and robust, an adaptive target model is extended to MSPF in this paper. Experimental results show that MSPF with target model updating can robustly track the target through the whole sequences regardless of the change of target model.

OCIS codes: 100.0100, 100.2960, 330.0330.

Visual tracking of moving objects in the presence of background clutter is now an active area of research in computer vision because of the large number of applications. The goal of visual tracking is to find out the location of the object in each frame of the whole image sequences.

Existed tracking algorithms generally fall into two classes: deterministic tracking and stochastic tracking. Mean shift, firstly introduced into visual tracking by Dorin Comaniciu[1], is an excellent deterministic tracking algorithm. Its strength is computational effective and suitable to real time application. However, it is easy to lose the object due to its intrinsic limitation of exploring local maxima especially when the tracked object is with quick movement. Moreover, mean shift is hard to recover from a total occlusion. Different from mean shift, particle filters[2,3], a kind of stochastic tracking algorithms that use multiple discrete "particles" to represent the distribution over the location of the target, have shown to be very suitable for performing tracking in cluttered environments due to their ability of maintaining multiple hypothesis of probability distribution. More importantly, particle filters exhibit superior characteristic of recovering from the temporary lost track. But the drawback that particle filters require a large amount of number particles for accurately representing the probability distribution limits their applications to real time occasion.

To achieve good tracking performance, a novel method has been proposed in Refs. [4, 5] by combining the merits of mean shift and particle filter (MSPF) in a unified framework. The kernel of MSPF is to generate fewer discrete particles by stochastic motion model of particle filters and then each particle is shifted to its local position which holds a maximum matching score with the target model. Unfortunately MSPF fails in when the tracked target suffers from partial rotation or total rotation which giving rise to the change of model features.

In this paper, a modified MSPF is proposed to solve the problem above mentioned. Our improvement is to use an adaptable model instead of the fixed model.

Given the model of the object, object tracking problem can reduce to search the new target location where similarity between the target candidate and the target model is maximum. Generally, the exhaustive search[6] for the state space can attain an optimized solution at the cost of speed. A faster method than the exhaustive search is mean shift based on gradient descent. The aim of mean shift is to maximize the similarity function along the gradient direction of the function, thus reducing the search time greatly.

The target is represented by probability density functions (PDFs) of color feature due to its simplicity and robustness to changes in pose and illumination. To facilitate the application of mean shift, a continuous kernel is evoked into target representation. The target candidate centered at y in current frame is described using the normalized color distribution $p_y = \{p_y^1, \cdots, p_y^m\}$, m denotes the number of bins ($m = 16 \times 16 \times 16$ in our experiments).

$$p_y^u = f \sum_{i=1}^{n_h} k\left(\frac{\|y - x^i\|}{h}\right) \delta[b(x^i) - u], \ u = 1, \cdots, m, \quad (1)$$

where k is the kernel profile with bandwidth h, $f = \frac{1}{\sum_{i=1}^{n_h} k\left(\frac{\|y-x^i\|}{h}\right)}$ is the normalization constant factor to ensure $\sum_{u=1}^{m} p_y^u = 1$; the function b represents the color bin assigned at pixel x^i; δ is the Kronecker delta function; n_h is the number of pixels inside the candidate area. The same form can be applied for the color distribution of the target model $q = \{q^1, \cdots, q^m\}$.

To evaluate the distance between the model and the candidate, a similarity function based on Bhattacharyya similarity coefficient is defined by

$$\rho[p_y, q] = \sum_{u=1}^{m} \sqrt{p_y^u q^u}. \quad (2)$$

Thus the distance can be expressed by $D(p_y, q) = \sqrt{1-\rho[p_y, q]}$. The smaller the distance is, the larger the possibility that the target appears in the candidate area centered at y is. Under the condition that the displacement between two consecutive frames is with small change, the similarity function can be approximated by Taylor expansion around the value p_{y_0} which is the color distribution of the candidate area centered at the location y_0 in previous frame

$$\rho[p_y, q] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{p_{y_0}^u q^u} + \frac{C_h}{2}\sum_{i=1}^{n_h} w_i k\left(\left\|\frac{y-x_i}{h}\right\|^2\right), \quad (3)$$

where

$$w_i = \sum_{u=1}^{m}\sqrt{\frac{q_u}{p_{y_0}^u}}\delta[b(x_i)-u]. \quad (4)$$

Because of the differentiable property of kernel profile, the above approximation of similarity function $\rho[p_y, q]$ also become differentiable, thus facilitating the application of mean shift. Maximizing the Taylor expansion of similarity function amounts to a steepest ascent procedure along the gradient direction by iterative mean shift using kernel g

$$y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{y-x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{y-x_i}{h}\right\|^2\right)}. \quad (5)$$

After several iterations, mean shift converges to a stable position which is the finally location of target in current frame. The severe drawback of mean shift is the slow convergence speed.

From the probabilistic point of view, visual tracking comes down to estimate the object's state x_k by combining all the measurements $y_{1:k}$ up to time k as effectively as possible. In brief visual tracking aims at deriving the posterior probability density function $p(x_k\,|y_{1:k}\,)$ representing the configuration distribution of the target object in image coordinate.

Particle filters can be used to solve the above presented estimation problem and have been proved successful in recent year since the pioneered work[7]. In essence, particle filters are those recursive Bayesian filters dealing with more challenging situations (e.g. non-linear and non-Gaussian) encountered in real world. The basic idea of particle filters is to approximate a posterior distribution over unknown state variables by a set of particles, drawn from this distribution. Each particle is composed of a state vector x_k^i and an associated weight w_k^i: $\left\{(x_k^i, w_k^i), i = 1, \cdots, N\right\}$. The posterior density can be approximated by

$$p(x_k|y_{0:k}) \approx \sum_{i=1}^{N} w_k^i \delta(x_k - x_k^i). \quad (6)$$

Based on the prior state transition $p(x_k^i|x_{k-1}^i)$ and the observation likelihood $p(y_k|x_k^i)$, the weight of each particle can be recursively computed by

$$w_k^i \propto \frac{p(y_k|x_k^i)p(x_k^i|x_{k-1}^i)}{q(x_k^i|x_{k-1}^i, y_{0:k})}, \quad (7)$$

where $q(x_k^i|x_{k-1}^i, y_{0:k})$ is the important proposal density which is usually chosen as $p(xx_k^i|x_{k-1}^i)$ in the context of visual tracking.

In sum, the basic particle filter algorithm consists in four key steps orderly: 1) important sampling step: sample from the state transition to obtain a new set of particles; 2) weighting step: once each particle has been sampled, its weight is proportional to the likelihood $w_k^i \propto p(y_k|x_k^i)$ if the important proposal density is taken as $p(x_k^i|x_{k-1}^i)$; 3) outputting step: the estimated location of target object in each frame can be approximately obtained by taking the expectation of the posterior density $\hat{x}_k \approx \sum_{i=1}^{N} w_k^i x_k^i$; 4) re-sampling step: to avoid the phenomenon that most particles collapse to one position over time, the particles are re-sampled according to their weights. This operation results in the same number of particles, but very likely particles are duplicated while unlikely ones are dropped. This step is not necessary in each particle filter loop. Reference [8] shows more details about particle filters.

Once we have understood the idea of particle filters and mean shift, it is easy to incorporate mean shift into particle filter. Mean shift analysis is applied to new particles which have been achieved by important sampling step in particle filter, thus shifting those particles to their nearby local maximum mode of the posterior density. The reminder steps are the same as the particle filter algorithms. Here, more detailed discussion is given on target model updating.

Since MSPF without target model updating cannot address the scenario with changes of color feature of the tracked object, a model updating mechanism is here proposed to overcome the shortcoming. Unlike the fixed model, the changeable model is more suitable for the time-varying target. The center idea of model updating is to consider synthetically the impact of recent tracking results and the older target model. How to decide whether the recent tracking result is dominant in current frame becomes a crux. Here the judgment is realized by comparing the average likelihood of all particles with the given threshold τ in the sense that if the average likelihood is smaller than τ, it is shown that all target candidates in current frame have heavily deviated from the target model selected in the first frame and it is necessary to update the old model. Given the condition of model updating, the updating procedure is formulized as

$$\hat{q}_u^{\text{new}} = (1-\alpha)\hat{q}_u^{\text{old}} + \alpha p_u^{\text{s}}. \quad (8)$$

The superscripts of new and old denote the newly obtained target model and the older target model respectively, s represents the recent tracking result. α weights the contribution of the recent tracking result.

The whole implement procedure of MSPF with target model updating can be outlined as follows:

Step 1: important sampling: generate a new set of $N = 30$ particles: $x_k^i \sim p(x_k\,|x_{k-1}^i)$, $i = 1, \cdots, N$;

Step 2: mean shift search: apply mean shift for each particle until the stable position is attained. $\tilde{x}_k^i$ = meanshift(x_k^i);

Step 3: particle weighting: calculate the weight using

equation $w_k^i \propto p(y_k|\tilde{x}_k^i)$;

Step 4: result outputting: the same equation as particle filters;

Step 5: model updating: if the updating condition is satisfied, adjust the older model using the Eq. (8);

Step 6: re-sampling: the same as the particle filters.

In this section, some implementation issues are considered. The tracked target (the head) is modeled by an upright ellipse centered at (x, y) with minor half axis l. Then the state vector can be defined as $x = \{x, y, l\}$. Due to the random movement of the head, the motion model is represented by a simple random walk: $x_{k+1} = x_k + w_k$. The likelihood evaluation is based on Bhattacharyya similarity coefficient between the target model and the target candidate

$$p(y_k|x_k) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1-\rho[\hat{p}(x_k), \hat{q}]}{2\sigma^2}\right). \quad (9)$$

This section illustrates the performance of tracking algorithms by two real video sequences in lab environment. Specifically, the two videos focus on different aspects: one compares the effectiveness between particle filter, mean shift and MSPF; the other shows the superior performance of MSPF with model updating.

In Fig. 1, the comparison of the tracked result is presented using three algorithms: mean shift, particle filter, and MSPF respectively. All the programs are implemented in Matlab. In the experiment, we aim at tracking the face in our own lab with cluttered backgrounds (e.g. the box with similar color to the face) and image blur. The number of particles in particle filters is 200 while MSPF use only 30 particles.

From Fig. 1, it can be noticed that mean shift almost fails in tracking the face from the frame 136 when irregular and rapid movement of the face lead to overpass the bandwidth size of mean shift. Particle filters occasionally lose the face due to rapid movement of target. Meanwhile, particles filter are easily distracted by the box with similar color to the face at frame 141. Unlike mean shift and particle filters, MSPF can robustly track the face through the whole sequences and save much time with fewer particles than particle filter. Of course, MSPF is time-consuming comparing to mean shift, however it achieves robust tracking result at the expense of speed.

The second experiment shows the ability of MSPF with model updating for dealing with the drastical change of the target model. When the face turns around, MSPF loses the tracked object and converge to a wrong object with similar color to the target model. However MSPF with model updating can adjust the model according to the current situation, thus locating the head successfully. Some selected tracking frames are presented in Fig. 2.

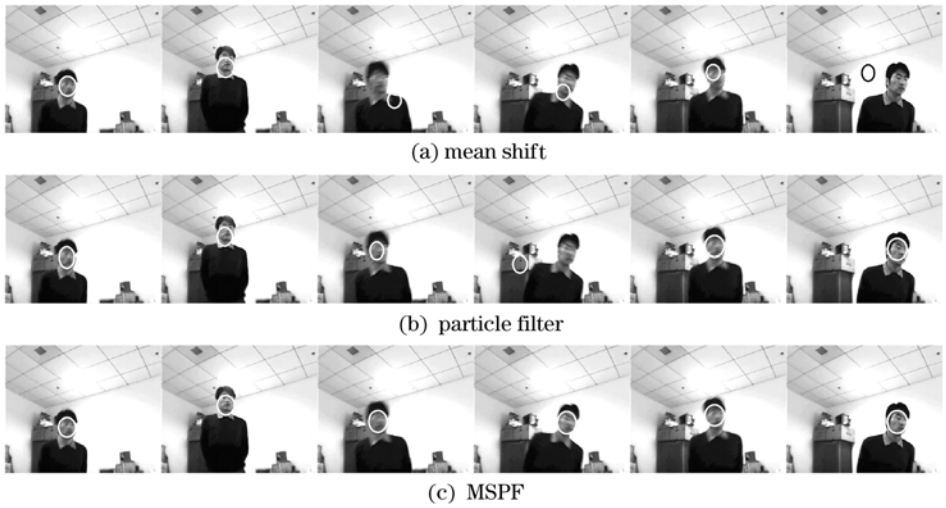

Fig. 1. Tracking result using three different algorithms: mean shift (a), particle filter (b), and MSPF (c). The sequences include 222 frames with the resolution 320×240. From left to right, the frame numbers are indexed as 1, 6, 136, 141, 163, 166.

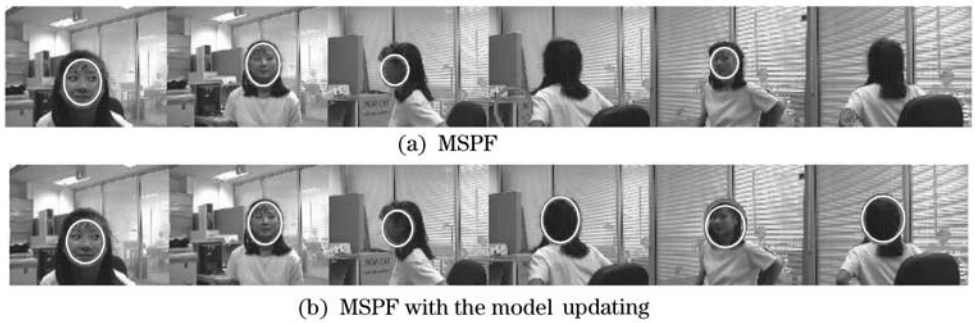

Fig. 2. Tracking the head. a) MSPF; b) MSPF with target model updating. From left to right, the frame numbers are 45, 60, 88, 100, 165, 182 in turn. The video is recorded in office with cluttered backgrounds and is composed of 500 frames with the size 128×96. The white ellipse denotes the estimated position of the head in image coordinate. The original sequences can be obtained from http://www.ces.clemson.edu/∼stb/.

In conclusion, mean shift is embedded into particle filter framework and thus the advantage of each method can be integrated for robust tracking. More importantly a target model updating mechanism is applied for widening the application of MSPF. Experimental results demonstrate that MSPF with model updating shows good performance for color changes, similar color appearance and cluttered backgrounds. Further work should aim at pursuing more suitable motion model. Furthermore, the accelerated method for solving the slow convergence problem of mean shift should also be investigated.

B. Zhang's e-mail address is zhang_bo@sjtu.edu.cn.

References

1. D. Comaniciu, V. Ramesh, and P. Meer, IEEE Trans. Patt. Analy. and Mach. Intell. **25,** 564 (2003).
2. M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, IEEE Trans. Signal Processing **50,** 174 (2002).
3. K. Nummiaro, E. Koller-Meier, and L. V. Gool, Image and Vision Computing **21,** 99 (2003).
4. C. Shan, Y. Wei, T. Tan, and F. Ojardias, in *The Proceedings of 6th International Conference on Automatic Face and Gesture Recognition* (2004).
5. E. Maggio and A. Cavallaro, in *Proceedings of IEEE Signal Processing Society International Conference on Acoustics, Speech, and Signal Processing* Philadelphia USA (2005).
6. S. Birchfield, in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* California, USA (1998).
7. M. Isard and A. Blake, Int. J. Computer Vision **29,** 5 (1998).
8. P. Li, T. Zhang, and A. E. C. Pece, Image and Vision Computing **21,** 111 (2003).