

The Study of Improved Particle Filtering Target Tracking Algorithm Based on Multi-features Fusion

Hongxia Chu^{1,2(✉)}, Zhongyu Xie², Du Juan², Rongyi Zhang²,
and Fanming Liu¹

¹ College of Automation, Harbin Engineering University,
Harbin, Heilongjiang, China
chx0420@163.com

² College of Electrical and Information Engineering,
Heilongjiang Institute of Technology, Harbin, Heilongjiang, China

Abstract. In view of the shortcomings of traditional particle filter which is lacking of utilizing current observational information, this paper proposes a multi-featured fusion tracking algorithm based on simulated annealing to improve particle filter. The proposed method solves the problem of large amount of computation and lack of particle number in high dimensional state. A hierarchical random search annealing method is used to generate a better proposal distribution in the Monte Carlo importance sampling. In the likelihood approximation, this paper integrated image feature attribute of colors and edges to generate weight function in the different annealing layer by weighting. Using this method to track the moving objects with complex background and occlusion, the experimental results show that the proposed method has high tracking accuracy and strong stability.

Keywords: Particle filtering · Proposal distribution · Simulated annealing · Multi-features fusion

1 Introduction

Research of target tracking in video sequence is an important and challenging task within the field of computer vision field. The core thought is to detect, track and distinguish targets as well as to describe and analyze them with computer vision technique in image sequences. The research is widely used in intelligent video surveillance, robot vision, human-computer interfaces and safety examination areas. The difficulty of target tracking centralizes on picture-noise influence, illumination variation, clutter, unstable target features, occlusion, posture variation and so on. So it is a challenge to design a high-speed robust target tracking algorithm. The representative target algorithm is particle filtering [1]. The key procedure is the confirmation of proposal distribution. The closer proposal distribution is from the posterior probability distribution, the better properties of the particle filtering. Traditional particle filter utilizes priori probability density function as the proposal distribution. It has the advantages of easy calculating weight and convenient sampling of the proposal probability density. Because the

proposal probability density has no relations with present quantity-measurements, efficiency of the particle is low. It can't solve the problem of calculating a large amount of particles and particle number degeneration under high-dimensional conditions. Some scholars combined particle filter with mean-shift to restrain the particle degeneration in some extent. A large number of particles are required in particle tracking, so it is difficult to assure real-time [2]. Some scholars combined multi-features within the frame of particle filter to improve accuracy and stability of the tracking [3–5]. Also, some scholars integrated color information with motion direction and other information within the frame of particle filtering, and it gets well experiment tracking effects [6]. Simulated annealing algorithm is a multi-mode random optimization method based on a probabilistic search approach and is seldom used in tracking documents. Jonathan Deutscher [7–9] used the information integration thought and applied annealing into particle filter. Then the performance of particle filter has been improved significantly for tracking the human body. Simulated annealing (SA) is a probabilistic method proposed by Metropolis for finding the global minimum of a cost function that may possess several local minima. SA, as an extension of partial search algorithm, produced a new state model randomly in the process of amending models. Nature mechanism introduction not only lets simulated annealing receive target function “better” test point in iteration procedure, but also lets simulated annealing receive target function “poor” test point according to certain probability. States in iteration process are random, and not demand the later states should be better than the former ones. Therefore, SA is easy to intervene to the existing model. It is extensibility and easy to combine with other technology. The idea of SA algorithm is introduced to PF. One-time state of the PF algorithm was transferred for changing process of particle state under the control of the temperature. Overall energy state of the PF system is an equilibrium with mutual restraint and mutual of the thermal motion effect inside the particle.

On the basis of the above analysis, in the light of defect of traditional particle filtering proposal distribution, which lacks the utilizing of current observation information, a kind of improved multi-feature integration annealing proposal distribution methods is proposed within the frame of particle filter video tracking application. Weighting function is produced by applying the image feature properties of the fusion between colors and edges to weight in different annealing layers. The combination bond is the calculation of particle weighting value. The comparison of experimental effects between traditional particle filter and improved annealing particle filter tracking was provided.

2 Particle Filtering

Particle filter draws out N individual distribution samples $\{x_{0:k}^{(i)}\}$ by utilizing the Monte Carlo method from the posterior probability density function $P(x_{0:k}|z_{1:k})$ of state. Posterior density function (PDF) of state can be approached as by empirical distribution

$$\hat{p}(x_{0:k}|z_{1:k}) = \frac{1}{N} \sum_{i=1}^N \delta(x_{0:k} - x_{0:k}^{(i)}) \quad (1)$$

However, PDF is unknown in general. At this time, N samples $\{x_{0:k}^{(i)}\}$ are needed to be individually drawn out from an important distribution function $q(x_{0:k}|z_{1:k-1})$ which can easily be sampled. PDF can be similar formulated as:

$$\begin{cases} \hat{p}(x_{0:k}|z_{1:k}) = \sum_{i=1}^N \tilde{\omega}_k^{(i)} \cdot \delta(x_{0:k} - x_{0:k}^{(i)}) \\ \tilde{\omega}_k^{(i)} = \omega_k^{(i)} / \sum_{i=1}^N \omega_k^{(i)} \end{cases} \quad (2)$$

Where, $\omega_k^j = \omega_{k-1}^j \frac{p(z_k|x_k^j)p(x_k^j|x_{k-1}^j)}{q(x_k^j|x_{k-1}^j, z_{1:k})}$ can be regarded as important weight value.

System state estimation on K time is

$$\hat{x}_k = \sum_{i=1}^N \tilde{\omega}_k^{(i)} x_k^{(i)} \quad (3)$$

$q(x_k^i|x_{k-1}^i, z_{1:k})$ is proposal distribution (important density) function. Selecting proposal distribution is very important in the whole process. The most simple and easy to implement approach is to make it equal to the prior density, that is $q(x_k^i|x_{k-1}^i, z_{1:k}) = p(x_k^i|x_{k-1}^i)$. At this time, $\omega_k^i = \omega_{k-1}^i p(z_k|x_k^i)$. It's obvious that the method hasn't considered the latest observation value. There is a comparatively big deviation between the samples drawn from the important function and the one generated by true posterior distribution. When the distribution of the likelihood function is narrow or there are a few overlaps between the distribution of prior density and the measurement likelihood function, only a small number of particles can get bigger weight values. So it makes more particles abandoned in the re-sampling procedure and aggravates the particle degeneration. It maybe leads to the failure of particle filtering. Aim to this defect, the proposal distribution function is selected through simulation annealing thought. After improvements, annealing particle filtering will not depend on the model. Even though model lacks of precision or observation noise becomes louder, this reference distribution can also effectively express the real distribution. Meanwhile, the linearism of systematic state equation is unnecessary when updating samples. Particle filter can really accomplish non-linearity and solve the problem of particle degradation.

2.1 System State Description and State Transferring Model

The purpose of video track is to get position coordinate and dimensional information of motive target. The rectangle can be used to describe an interesting area. For motive targets, it is difficult for a random model to satisfy motive description, so introducing speed weight. At the same time, the components of the width, height, and weight of the target are introduced in order to meet target change. Then target state vector can be expressed by one six-dimensional vector. It can be parameterized as $x = \{x, y, \dot{x}, \dot{y}, s_x, s_y\}$.

Where, x and y is centroid coordinates of the rectangle. \dot{x} and \dot{y} is the velocity of targets along x and y . s_x and s_y is the width and height of targets. We use first-order auto-regressive (AR) equation to define dynamic models. It can be formulated as

$$x_k = Ax_{k-1} + v_k \tag{4}$$

Where, A is a systematic state transferring matrix. v_k is process noise.

3 Annealing Particle Filtering

3.1 The Weighting Function

A number of factors must be taken into account when deciding which image features are to be used to construct the weighting function. Firstly, the used image features should be invariant under a wide range of conditions so that the same tracking framework will function well in a broad variety of situations. Secondly, in an effort to make the tracker as efficient as possible the used features must be easy to extract. Three image features were chosen to construct the weighting function: color and edge features and texture.

Color is an important feature of the target. This article uses the “kernel” concept referred in mean shift algorithm to create a color histogram [10]. The weighted color histogram is used as a color distribution model of target. In RGB space, color histograms are calculated with many small color bits. In our experiments, $8 \times 8 \times 8$ bits are sufficient to represent the color distribution for pixels with 8-bit color depth in each channel. When constructing color distribution models for a half-length and half-width rectangular area, weighting function is selected according to the different contributions pixel in different areas towards color histograms.

$$k(r) = \begin{cases} 1 - r^2 & r < 1 \\ 0 & r \geq 1 \end{cases} \tag{5}$$

In this, r is the distance from some points to the regional center. Thus, particle color histograms which regard y as the center candidate region can be expressed as:

$$p_u(y) = C_h \sum_{i=1}^N k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \delta[b(x_i) - u] \tag{6}$$

Where, C_h expresses normalization constant. N is the total pixel number of the target region. u is the index value of histogram segments. $b(x_i)$ means the instruction function of pixel point x_i in its histogram. $\delta(\cdot)$ is the Kronecker delta function. $h = \sqrt{h_x^2 + h_y^2}$ describes the size of the target area.

Similarity is measured by the Bhattacharyya distance between color histogram $p_u(y)$ in candidate model and color histogram q_u in the target model. That is

$$d_c = \sqrt{1 - \rho(p_u, q_u)} \quad (7)$$

$$\rho(p_u, q_u) = \sum_{u=1}^M \sqrt{p_u(y), q_u} \quad (8)$$

Equation 7 is the discrete Bhattacharyya coefficient. Color histogram is more similarity between candidate model and target model when d_c becomes smaller gradually. Similarity likelihood observation function distance for distance is modeled as a Gaussian distribution [11, 12].

$$w^{co} = w_{color}(Z_{color,k}|x_k) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{d_c^2}{2\sigma^2}\right) \quad (9)$$

3.1.1 Edge Feature

In complicated situations, single color feature information doesn't contain any motion and shape information. Also, it's easily influenced by illumination variation and clutter. Edge feature, as another important feature, can effectively adapt to illumination variation. Thus, we regard edge feature as the second tracking feature and add edge feature information into an object model. That means, oval or rectangular with parameters is regarded as shape models. Suppose in the experiment, we use rectangle contour to select target area, thus we need count similar function of shape cue according to the rectangle. The similar accountant of the rectangle is using one point p , and then drawing one measuring line from this p point towards the center of the rectangle. Along this line, there are n Fixed-interval sampling points around the center point p . On each point, similar function is count with the canny edge detector. Suppose real edge point distributions are standard, that the mean is zero and the variance is σ^2 . Then the similar function of observation sampling points are:

$$w_{shape}(z_{shape,k}^{(l)}|x_k) = 1 + \frac{1}{\sqrt{2\pi}\sigma h_0} \sum_{j=1}^{n_l} \exp\left(-\frac{(z_j - x)^2}{2\sigma^2}\right) \quad (10)$$

h_0 is priory probability of being measured unreal edge. Z_j is the distance from being measured feature point to the rectangle. Then all the similar functions m lines with even distribution around rectangle can be formulated as [13]

$$w^e = w_{shape}(z_{shape,k}|x_k) = \prod_{l=1}^m P_{shape}(z_{shape,k}^{(l)}|x_k) \quad (11)$$

3.1.2 Texture Feature

Texture feature is an important tracking feature of the target description. It reflects the properties of the image itself and has a strong mutability of the anti-light photo. But also texture features based on the gray level co-occurrence matrix have the ability of anti

1	2	3
4	5	6
7	8	9

Fig. 1. Pixel distribution map

noise and occlusions. Its extraction is simple and processing speed is fast. It meets the requirements of the target tracking in real time and accuracy. It can make up that the color feature is easily affected by illumination and occlusion by edge feature. The description of texture features based on the gray level co-occurrence matrix is shown in Fig. 1.

It assumes that the pixel value at position five is $G_{x,y}$. The transverse direction is x , and the longitudinal direction is y . Gray difference was calculated between each pixel and the pixel in the direction of 45° , 135° , 90° and 0° for the selected tracking target area.

$$\begin{aligned}
 G_1(x, y) &= G(x + 1, y + 1) - G(x - 1, y - 1) \\
 G_2(x, y) &= G(x - 1, y + 1) - G(x + 1, y - 1) \\
 G_3(x, y) &= G(x, y + 1) - G(x, y - 1) \\
 G_4(x, y) &= G(x + 1, y) - G(x - 1, y)
 \end{aligned} \tag{12}$$

The extracted gray co-occurrence matrix is calculated through two order statistics, and mean matrix is used to compute G_5 :

$$G_5(x, y) = [G_1(x, y) + G_2(x, y) + G_3(x, y) + G_4(x, y)]/4 \tag{13}$$

Similar to color histogram, gray histogram of image based on texture feature is obtained. Then texture similarity d_t is obtained between the target template and the candidate target. So the similar observation likelihood function of texture feature is:

$$w^f = w_{texture}(Z_{texture,k}|x_k) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{d_t^2}{2\sigma^2}\right) \tag{14}$$

3.1.3 Multiple Cues Integration of Annealing Particle Filtering

Particle filter provides a good probability framework for integration tracking. Any probability observation models can accomplish tracking tasks within the framework. In order to satisfy robust tracking requirements, we use former color, edge and texture cues. Weighting function is generated by weighting the image features property of color and edge integrated in different annealing layers, so that the proposal distribution is improved. The counting integration link is the account of particle weighting values.

According to Reference [14, 15], we suppose the observation statistic of each cue is individual, then the integrated similar weighting function on the x_k state can be formulated as

$$w^{\beta(l)} = \exp\left(-\sum_{c=1}^C (\lambda_c^{co}(\beta(l), l)w^{co} + \lambda_c^e(\beta(l), l)w^e + \lambda_c^t(\beta(l), l)w^t)\right) \quad (15)$$

In Eq. (15) $\beta(l)$ is the annealing rate. The weight coefficient λ_c is calculated according to the annealing rate and characteristic importance. w^{co}, w^e, w^t is the observation function based on color, edge and texture. We call feature fusion based on the annealing. Its structure is shown in Fig. 2.

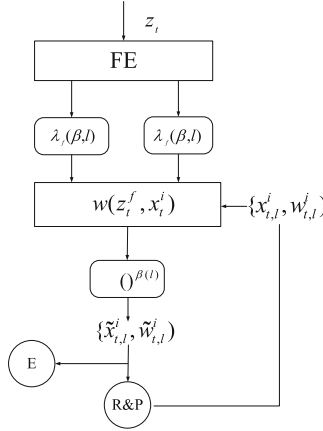


Fig. 2. Feature-based annealing PF

3.2 Annealing Procedure

Presented particle tracking method based on simulated annealing is different from traditional particle tracking method, this method generates partial particle using the posterior distribution of target color features through systematic state transferring equation. Meanwhile, it generates a weighting function with edge and texture particle features and applies image feature attribute of colors and edges to generate weight function at different annealing layer by weighing.

A series of weighting functions $w_0(Z, X)$ to $w_M(Z, X)$ are employed in which each w_m differs only slightly from each other. The function w_m is designed to be very wide, representing the overall trend of the search space while w_0 should be very peaked, emphasizing local features. Expression way can be formulated as:

$$w_m(Z, X) = w(Z, X)^{\beta_m} \quad (16)$$

In this formula, $\beta_0 > \beta_1 > \dots > \beta_M$, however, $w(Z, X)$ is the original weighting function.

One annealing procedure is achieved through image observation value z_k in each time t_k . The state of the tracker after each layer of an annealing procedure is represented

by a set of N weighted particles $S_{k,m}^\pi = \{(s_{k,m}^{(0)}, \pi_{k,m}^{(0)}) \dots (s_{k,m}^{(N)}, \pi_{k,m}^{(N)})\}$. But an unweighted set of particles can be described as $S_{k,m} = \{(s_{k,m}^{(0)}) \dots (s_{k,m}^{(N)})\}$. In $S_{k,m}^\pi$, each particle is regarded as a $(s_{k,m}^{(i)}, \pi_{k,m}^{(i)})$ pair. $\pi_{k,m}^{(i)}$ is the corresponding particle weight. Each annealing procedure can be described as follows:

- (1) About each time step t_k , annealing procedure begins on M layer, and $m = M$.
- (2) Each layer in annealing run is initialized by a set of un-weighted particles $S_{k,m}$.
- (3) Then, each of these particles is allocated with a weight

$$\pi_{k,m}^{(i)} \propto w_m(Z_k, s_{k,m}^{(i)}) \tag{17}$$

- (4) N particles are drawn randomly from $S_{k,m}^\pi$ with replacement and with a probability equal to their weight $\pi_{k,m}^{(i)}$. Particle $s_{k,m-1}^{(n)}$ is generated by choosing the n^{th} particle $s_{k,m}^{(n)}$. The formula can be described as:

$$s_{k,m-1}^{(n)} = s_{k,m}^{(n)} + B_m \tag{18}$$

where B_m is the multi-variate Gaussian stochastic variation. Variance is P_m and mean is 0.

- (5) $S_{k,m-1}$ which has been generated is used to initialize the $m - 1$ layer. The process is repeated until we arrive at the set $S_{k,0}^\pi$.
- (6) $S_{k,0}^\pi$ is used to evaluate the optimal model configuration X_k using

$$X_k = \sum_{i=1}^N s_{k,0}^{(i)} \pi_{k,0}^{(i)} \tag{19}$$

- (7) Then, the set $S_{k+1,m}$ is produced from $S_{k,0}^\pi$ using

$$s_{k+1,M}^{(n)} = s_{k,0}^{(n)} + B_0 \tag{20}$$

3.3 Annealing Rate of Tracking Parameter Setting

As stated previously the function $w_M(Z, X)$, used in each layer of the annealing process, is determined by Eq. 16 with $\beta_0 > \beta_1 > \dots > \beta_M$. The value of β_m will determine the rate of annealing at each layer. A large β_m will produce a peaked weighting function w_m resulting in a high rate of annealing. Small values of β_m will have the opposite effect. If the rate of annealing is too high the influence of local maxima will distort the estimate of X_k . If the rate is too low X_k will not be determined with enough resolution.

A good measure of the effective number of particles will be chosen for next layer propagation. We do not use the exact gradient descent method, but the survival rate is

adjusted by using annealing rate. It is a simple amendment on the basis of present survival rate, so the algorithm is simple and effective.

$$\beta(l) = \beta(l - 1) - \varepsilon(a_{target} - a(l - 1)) \quad (21)$$

Where a_{target} is the expected survival rate of particles on each layer. $a(l - 1)$ is the survival rate weighted on last layer. ε is the learning factor. It is usually set to $\frac{1}{l+1}$ and satisfied $\beta(l) \geq \beta(l - 1)$.

4 Experiment Results and Analysis

In order to verify the results of tracking method, a large amount of experiments is carried out. The sequences in experiments can be accessed human face tracking sequences [16, 17] Experiments are done on Pentium 2.4 GHZ CPU Common Configuration Computer. Image size is 320×240 . Video capture rate is 25 frames per second. The initial state of particle region is set to $x_0 \sim U(0, 320)$, $y_0 \sim U(0, 240)$, $\Delta x \sim N(0, 2)$, $\Delta y \sim N(-10, 6)$ and the number of particles is $N_S = 100$. Annealing layers $m = 3$, $T_0 = 1000$, $T_{min} = 50$, $intex_max = 10$, $\beta(l) = 0.99$, $a_0 = a_1 = a_2 = a_3 = 0.5$.

4.1 Real Time Tracking Experiment

The comparison of two tracking algorithms is given. In the first experiment, sequences one, which is constructed from the 84th frame to the 95th frame, experienced the procedure of face being sheltered and light becoming darker. From the 115th frame to the 130th frame, there was a fast squat downward procedure. From the 421th frame to the 430th frame, it's a full occlusion process. From the results of the experimental sequences, traditional particle filter tracking was failed. This is because it only uses color information. Then, when the target entered illumination area, the colors of the target changed dramatically, and the proposal distribution didn't utilize present observation information. Then, it could not capture color variation and lead the algorithm to fail. Current color, edge, and texture observation information can be utilized to the proposal distribution, so the target is well discriminated. Besides, edge information is not sensitive to the illumination variations. Thus, when the color feature lost its discrimination, edge and texture information played a leading role. It made the whole sequence reliably track the target. From the results of the two experiments, tracking of annealing particle filtering is effective and stable because the improvement of proposal distribution under complicated backgrounds situations, existing partial occlusion and similar target color (Figs. 3 and 4).

Second experimental sequences are selected from <http://www.ces.clemson.edu/~stb/research/headtracker/>. The tracked target is in a more complex tracking environment with the similar background, light, rotation, occlusion and background interference. Because the traditional particle filter can not achieve tracking, the paper does not give the experimental results. And the algorithm can achieve tracking precision.

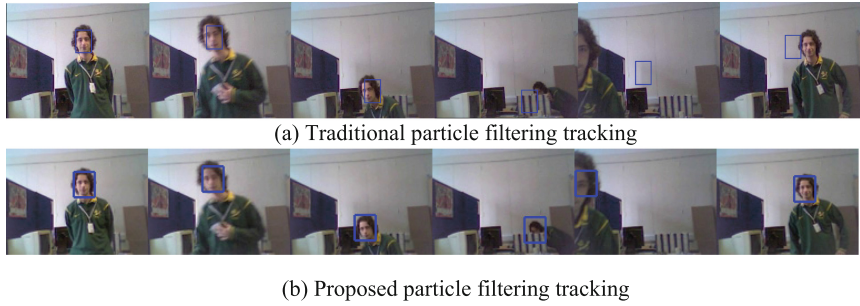


Fig. 3. The first experiment results (1th, 107th, 184th, 432th, 537th, 600th tracking results)



Fig. 4. The second experiment results (the 1th, 30th, 47th, 89th, 203th, 233th, 247th, 287th, 296th, 445th tracking results)

4.2 Performance Analysis

4.2.1 Tracking Precision

In order to better describe the performance of the improved algorithm, the latest improvements of the other PF algorithms are compared with the algorithm of this paper for the number of different particles, the results are shown in Fig. 5 and Table 1. The average root mean square error (RMSE) of the SAPF algorithm is significantly lower than that of the PF algorithm under the same conditions. The average RMSE value of the SAPF algorithm decreases significantly with the increase of the number of particles. It shows the probability of the optimal estimation of the tracking approach. At the same time, it can be seen that if the PF algorithm achieves the same tracking accuracy of that of the SAPF algorithm, we must increase the number of particles, which leads to a long run time, can not meet the needs of real-time target tracking

4.2.2 Test Tracking Speed

The total running time of the PF and SAPF algorithm was observed at 100 times one time, and the results are shown in Table 2. From Table 2, it is easy to see that the importance sampling step of the improved algorithm is slightly complicated, the time consumption of the algorithm is slightly longer than that of the common PF algorithm. But the increased time overhead is within the scope of acceptance. With the

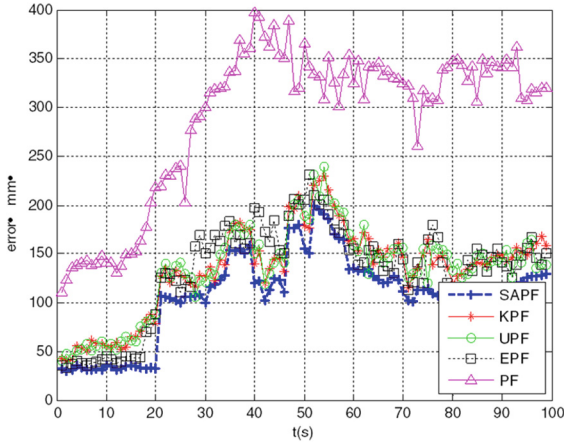


Fig. 5. RMSE of target tracking of PF, KPF, EPF, UPS, SAPF algorithms

Table 1. Average *RMSE* comparison results of PF, SAPF algorithms

<i>RMSE</i>	N = 50	N = 100
PF	12.5438	10.4623
SAPF	7.7842	6.3597

Note: 1 independent simulation

Table 2. Comparison results of the running time of PF, SAPF algorithms

TIME	N = 50	N = 100
PF	7.3250	12.2507
SAPF	7.9715	13.1782

Note: 1 independent simulation

improvement of the tracking accuracy, it can be concluded that the improved algorithm significantly improves the tracking performance of the system.

4.2.3 Comparison of Various Algorithms

Compared with other improved PF algorithms, the results are shown in Table 3.

Table 3. Performance comparison of PF, KPF, EPF, UPF, SAPF algorithm Unit: s

Algorithms	PF	KPF	EPF	UPF	SAPF
Average <i>RMSE</i>	9.2760	5.0542	8.6535	6.5428	3.6949
Average <i>TIME</i>	7.8158	9.7621	8.5125	12.0746	8.3826

Note: 50 independent simulations, N = 50

We can observe from Table 3 that the average RMSE values of the improved PF algorithm were closer to the true value than the PF algorithm. It indicates that the tracking accuracy and reliability of the improved PF algorithm are better than that of the basic algorithm of PF. But the time cost of the KPF, EPF and the UPF algorithm is difficult to adapt to the real-time target tracking. SAPF algorithm gradually made the tracking accuracy approach to the optimum by improving the importance sampling density function. The computing time is also able to meet the time constraints of the real-time target tracking.

5 Conclusion

The paper presents a kind of multi-feature integration annealing particle filtering method. Weighting function is produced by applying the image feature properties of the fusion between colors and edges to weight in different annealing layers, so the proposed distribution is improved. It not only makes the target never excessively depends on the individual feature, but also effectively solves the particle degradation problem. Algorithm thus can effectively avoid illumination variation and complicated backgrounds influence and our method obtains well tracking results. The future work will increase the contour information and texture information, and so on. We will pay more attention on the effectiveness of integration method.

Acknowledgments. This research was supported by the project of research foundation of the talent of scientific and technical innovation of Harbin City (NO. 2014RFQXJ103).

References

1. Doucet, A., de Freitas, N., Gordon, N. (eds.): Sequential Monte Carlo Methods in Practice. Springer, New York (2001)
2. Maggio, E., Cavallaro, A.: Hybrid Particle filter and mean shift tracker with adaptive transition model. In: IEEE International Conference on Acoustics (2005)
3. Xin, Y., Jia, L., Pengyu, Z.: Adaptive particle filter for object tracking based on fusing multiple features. *J. Jilin Univ.* **45**(2), 533–539 (2015). (Engineering and Technology Edition)
4. Gu, X., Wang, H., Wang, L.: Fusing multiple features for object tracking based on uncertainty measurement. *Acta Autom. Sin.* **37**(5), 550–559 (2011)
5. Maggio, E.: Adaptive multi-feature tracking in a particle filtering framework. *IEEE Trans. Circuits Syst. Video Technol.* **17**(10), 1348–1359 (2007)
6. Xiaowei, Z.: Particle filter tracking algorithm combining the color and structural information. *Opto Electron. Eng.* **35**(10), 1–6 (2008)
7. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. In: Proceedings of Conference on Computer Vision and Pattern Recognition, Hilton Head, South Carolina, vol. 2, pp. 1144–1149 (2000)
8. Yang, S., Wu, T., Zhang, Y.: Particle filter based on simulated annealing for target tracking. *J. Optoelectron. Laser* **22**(8), 1236–1240 (2011)

9. Li, Y., Sun, Z., Chen, S.: 3D human pose analysis from monocular video by simulated annealed particle swarm optimization. *Acta Autom. Sin.* **38**(5), 732–741 (2012)
10. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(5), 564–577 (2003)
11. Nummiaro, K., Koller-Meier, E., Gool, L.V.: An adaptive color-based particle filter. *Image Vis. Comput.* **21**, 99–110 (2003)
12. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) *ECCV 2002*. LNCS, vol. 2350, pp. 661–675. Springer, Heidelberg (2002). doi:[10.1007/3-540-47969-4_44](https://doi.org/10.1007/3-540-47969-4_44)
13. MacCormick, J.: Probabilistic modeling and stochastic algorithms for visual localization and tracking. Ph.D. thesis, University of Oxford, UK (2000)
14. López Méndez, A.: Feature-Based Annealing Particle Filter for Robust Motion Capture. Image and Video Processing Group, pp. 1–72 (2009)
15. Shao, P., Wan, C.: Genetic-annealing algorithm for global optimization problems. *Comput. Eng. Appl.* **43**(12), 62–65 (2007)
16. <http://www.ces.clemson.edu/~stb/research/headtracker/>
17. <ftp://motinas.elec.9mul.ac.uk/pub/single-face>