Visual Tracking by Sequential Cellular Quantum-Behaved Particle Swarm Optimization Algorithm

Junyi Hu, Wei Fang(✉), and Wangtong Ding

School of IoT Engineering, Jiangnan University, Wuxi, Jiangsu, China
fangwei@jiangnan.edu.cn

Abstract. Visual tracking is a very important application in the field of computer vision. The tracking process can be formulated as a dynamic optimization problem, which can be solved by particle swarm optimization (PSO) algorithms. PSO algorithm with particle filter (PF) has been actively used in visual tracking. In this paper, we propose an improved resampling cellular quantum-behaved PSO (RScQPSO) algorithm, which is a probabilistic variant of PSO, and combine the PF to solve the tracking problem. The cQPSO algorithm can better keep the population diversity and balance the global and local search than PSO algorithm. For better tracking performance, we further improve the tracking algorithm by improving the particle initialization approach in cQPSO, resampling technique in PF as well as using the Gaussian mixture model in fitness assessment. Experimental results demonstrate that the proposed tracking algorithm is more effective and accurate, especially for the cases that the object has an arbitrary motion or undergoes large appearance changes, than the compared algorithms.

Keywords: Visual tracking · Quantum-behaved particle swarm optimization · Particle filter

1 Introduction

Visual tracking has been a core issue in many applications, such as monitor, visual-based control, human-machine interface, intelligent transportation and augmented reality, etc. Robust target tracking algorithm needs to undergo a more stringent test of complex conditions (such as sudden movement, fast-moving, shape variation, and occlusion, etc.). Particle Filter (PF) [1,2] is an effective target tracking technology. Compared with traditional filtering method, PF has unique advantages in dealing with the parameter estimation and filtering problem of non-Gaussian and nonlinear time-varying system. In PF, when the importance weight variance increasing over the time, the weight only concentrates on a small number of particles so that the particle collector cannot express the true posterior probability distribution, which is termed as the 'degradation of particles'. Resampling method [3] can inhibit the degradation of particles, but it

M. Gong et al. (Eds.): BIC-TA 2016, Part II, CCIS 682, pp. 86–94, 2016.
DOI: 10.1007/978-981-10-3614-9_11

may cause the loss of particle diversity. Particle Swarm Optimization (PSO) is a new optimization algorithm proposed in recent years, which is inspired by the social behavior of bird flocking [4]. PSO algorithm has been used with PF for visual tracking problems in order to improve the diversity of particles and the stability of particle filter. In [5], a target tracking method based on PSO-PF is proposed to solve the problem of particle diversity recession. However the precision of this PSO-PF target tracking algorithm still depends on the accuracy of PSO. Similarly, In [6], another method called PPF can also deal with the particle diversity recession to some extent with the help of PSO. However, the standard PSO algorithm has some shortcomings, such as low precision, easily getting into local optimum and slow convergence. A lot of PSO variants have been proposed to address the shortcomings. Quantum-behaved particle swarm optimization (QPSO) algorithm is one of the PSO variants and is proposed based on the trajectory analysis of particles in PSO [7] and the quantum potential well model [8,9]. QPSO algorithm has been used for solving many optimization problems with satisfying performance [10]. In [11], a decentralized form of QPSO (cQPSO) is proposed by using the cellular structured population. In cQPSO algorithm, the local attractor in the sub-population is also modified in order to accelerate the diffusion of the best solution. The cQPSO algorithm can better keep the population diversity and balance the global and local search than QPSO algorithm. In this paper, we propose the tracking algorithm based on the cQPSO and combining the resampling section of PF. For better tracking performance, we further improve the tracking algorithm by improving the particle initialization approach in cQPSO, resampling technique in PF as well as using the Gaussian mixture model in fitness assessment.

The rest of this paper is organized as follows. Section 2 introduces related work on particle filter and QPSO algorithm. In Sect. 3, the improved tracking algorithm, which is called improved resampling cQPSO(RScQPSO), is proposed. Experimental results are given in Sect. 4. Section 5 concludes this paper.

2 Related Work

2.1 Particle Filter

Particle filter is commonly used in solving Bayesian probability, and it is a technique to estimate target motion state from data with noise. The main idea of PF is generating a random sample called particles in the state space, combing the observation of each moment to adjust the weight and position of each particle. Through continuous searching, decision-making and resampling, particle filter replace integral calculation with sample mean as estimation value of system state. This algorithm only requires simple iteration, and can be applied to predicting and smoothing problems. It can also handle non-Gaussian noise, and allow the use of non-linear system equations. Particle filter is flexible and easy to be implemented. Therefore it is widely used in target tracking. Specific processes of particle filter includes initialization, sampling, state transition, and systematic observation.

2.2 Cellular QPSO (cQPSO) Algorithm

The particle's position $X_{i,j}$ of QPSO algorithm updates according to the following equations:

$$X_{i,j}(t+1) = p_{i,j}(t) \pm \alpha \cdot |C_j - X_{i,j}(t)| \cdot \ln\left(\frac{1}{u_{i,j}(t)}\right) \tag{1}$$

$$p_{i,j}(t) = \varphi_{i,j}(t) \cdot P_{i,j}(t) + (1 - \varphi_{i,j}(t)) \cdot G_{i,j}(t) \tag{2}$$

$$C_j(t) = \frac{1}{M}\left(\sum_{i=1}^{M} P_{i,1}(t), \sum_{i=1}^{M} P_{i,2}(t), \cdots, \sum_{i=1}^{M} P_{i,N}(t)\right) \tag{3}$$

where $p_{i,j}(t)$ is the local attractor position, $P_{i,j}(t)$ is the personal best position of the ith particle at the tth iteration, $G_{i,j}(t)$ is the global best position of the population, $\varphi_{i,j}(t)$ is a random number being in the range of [0, 1], M and N represent the population size and problem dimension respectively.

We expanded the QPSO algorithm to cQPSO algorithm by modifying the neighborhood structure according to the cellular automata (CA) [11]. In cQPSO algorithm, the particles distributed in the two dimensional toroidal mesh and they can only interact with the neighbors. Six kinds of neighborhood structure have been studied in cQPSO as shown in Fig. 1.

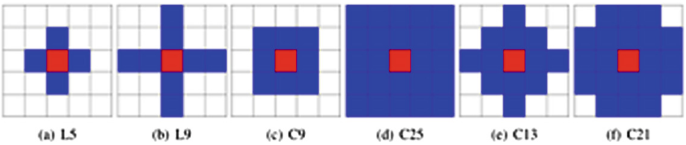


Fig. 1. Six kinds of neighborhood structure

In this paper, we use the C9 structure in cQPSO and then the average position equation $C_j(t)$ is changed to

$$lmbest_j(t) = \frac{1}{9}\left(\sum_{i=1}^{9} P_{i,1}(t), \sum_{i=1}^{9} P_{i,2}(t), \cdots, \sum_{i=1}^{9} P_{i,N}(t)\right) \tag{4}$$

Therefore the particle in cQPSO updates according to the following equation

$$X_{i,j}(t+1) = lbest_{i,j}(t) \pm \alpha \cdot |lmbest_j(t) - X_{i,j}(t)| \cdot \ln\left(\frac{1}{u_{i,j}(t)}\right) \tag{5}$$

where $lbest_{i,j}(t)$ is best $P_{i,j}$ position of the particles in the C9 neighborhood.

3 Improved Sequential cQPSO Algorithm for Target Tracking

For better tracking the object with higher accuracy, we improve the tracking algorithm which is based on the sequential cQPSO algorithm in this section. There are four improvements on the proposed sequential cQPSO algorithm, including the improvement on the particle initialization for cQPSO, resampling technique learnt from PF as well as the application of Gaussian mixture model.

3.1 The Improvement on the Particle Initialization

Since the video sequence is continuous, we can model the target movement according to previous motion results. The particles are distributed within the scope where the target is most likely to appear in the next frame. Let the successfully searching target position of the present frame and the previous frame be $g(t)$ and $g(t-1)$, the $v_{pre}(t+1)$ can be predicted by the following formula:

$$v_{pre}(t+1) = g(t) - g(t-1) \tag{6}$$

We should preset a v_{min}, when $v_{pre}(t+1) < v_{min}$, $v_{pre}(t+1) = v_{min}$. Next, use the following Gaussian distribution $x_{i,0}(t+1) \sim N(g(t), \Sigma)$ or motion model $x_{i,0}(t+1) = Ax_{i,0}(t) + Bw_{i,0}(t)$ to make particles evenly distributed around the current target. Σ is a covariance matrix of Gaussian distribution in which the diagonal elements is the $v_{pre}(t+1)$.A generally take 1, and B is aforementioned $v_{pre} \cdot k_v$ (k_v is a predefined constant). $w_{i,0}(t)$ is a random number in $[-1, 1]$.

3.2 The Improvement on the State Representation of Particles

The particles state of the proposed target tracking algorithm is represented by $par_i = (X, S, \theta)$. $X = (x, y)$ represents the corresponding position of the particles in images to be tracked, $S = (sx, sy)$ represents scaling ratio between the images corresponding to particles and the original image of the target, θ is the deflection angle of the image. Under the initial state, the location of corresponding pixels can be obtained by the following transformation and the tracking result is shown in Fig. 2.

$$(x, y) = X + T \times \begin{vmatrix} sx & 0 \\ 0 & sy \end{vmatrix} \times \begin{vmatrix} \cos(\theta) & sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{vmatrix} \tag{7}$$

Fig. 2. Affine result

3.3 The Improvement on MOG Appearance Mode

The model used in the proposed algorithm is the MOG (mixture of Gaussian) model, which is used to evaluate the fitness value of particles. Similar to [12,13], three components, which are S, W, F, exist in the appearance model. The S component describes temporarily stable images, the W component characterizes the two-frame variations, and the F component is a fixed template of target which is used to prevent the model from drifting away. Based on this appearance model, we can calculate individual's fitness value according to the following equation,

$$f(x(t)) = p(o(t)|x(t)) = \Pi_{j=1}^{d} \left\{ \Sigma_{l=s,w,f} \pi_{l,j}(t) N(o_j(t); \mu_{l,j}(t), \sigma_{l,j}^2(t)) \right\} \tag{8}$$

where $\{\pi_{l,j}(t), \mu_{l,j}(t), \sigma_{l,j}^2(t), l = s, w, f\}$ respectively represent mixture probabilities, mixture centers and mixture variances of these three components, $o_j(t)$ is the candidate region corresponding to state of particle $x(t)$ and d is the number of pixels inside $o_j(t)$. $N(\cdot)$ is a Gaussian density defined as follows,

$$N(x; \mu, \sigma^2) = (2\pi\sigma^2)^{-1/2} exp\{-\frac{(x-\mu)^2}{2\sigma^2}\} \tag{9}$$

More importantly, we use a forgetting factor to avoid keeping all the data from the previous frame. This makes the model parameters be more dependent on recent observations. An online EM algorithm is used as follows:

E-step:

The ownership probability of each component is computed as:

$$m_{l,j}(t) \propto \pi_{l,j}(t) N(o_j(t); \mu_{l,j}(t), \sigma_{l,j}^2(t)) \tag{10}$$

which fulfills $\Sigma_{l=s,w,f} m_{l,j}(t) = 1$

The mixing probability of each component is computed as:

$$\pi_{l,j}(t+1) = \partial m_{l,j}(t) + (1-\partial)\pi_{l,j}(t); l = s, w, f \tag{11}$$

And the first- and second-moment images are evaluated as (14)

$$M_{k,j}(t+1) = \partial o_j^k(t) m_{s,j}(t) + (1-\partial) M_{k,j}(t); k = 1, 2 \tag{12}$$

where $\partial = 1 - e^{-1/\tau}$ acts as a forgotten factor, and τ is predefined.

M-step:

The mixture centers and the variances are estimated in the M-step:

$$\mu_{s,j}(t+1) = \frac{M_{1,j}(t+1)}{\pi_{s,j}(t+1)}, \sigma_{s,j}^2(t+1) = \frac{M_{2,j}(t+1)}{\pi_{s,j}(t+1)} - \mu_{s,j}^2(t+1) \tag{13}$$

$$\mu_{w,j}(t+1) = o_j(t), \sigma_{w,j}^2(t+1) = k_{MOG} \sigma_{s,j}^2(t+1) \tag{14}$$

$$\mu_{f,j}(t+1) = \mu_{f,j}(1), \sigma_{f,j}^2(t+1) = k_{MOG} \sigma_{s,j}^2(t+1) \tag{15}$$

3.4 The Improvement on Resampling

When cQPSO iteration comes into the later period, the convergence speed becomes slower. We introduce the resampling step of the particle filter into the proposed algorithm. In that way, we can reduce the search region as well as improve the convergence speed and the solution accuracy. In the proposed algorithm, when the number of iterations is greater than IF (IF is predefined), after each iteration, generate a random number $u \sim U(0,1)$. If $u > p$ (p is predefined), resample as follows.

Firstly, calculate the weight of each particle (w_i) according to the distance between it and the global best of particles and normalize them. Let $C_i = C_{i-1} + w_i$. Then generate n random numbers $u_i \sim U(0,1)\, i \in [1,n]$. For each u_i, find the Minimal j that meet $C_j > u_i$. Finally, copy the j_{th} particle into the new particle population.

3.5 The Improved Sequential RscQPSO Based Tracking Algorithm

Above all, the process of the proposed algorithm is shown in Table 1.

Table 1. The process of the proposed algorithm

1. Load and display the original image, outline the target
2. Built the appearance models
3. Randomly initialize the particles $X_{i+1} = \left\{x^{i,0}\right\}_{i=1}^{N}$
4. For k=0 to T do
For i=1 to N do
5. Calculate particles' fitness by the SWF model according to Eq.(8)
6. Update particles, personal best and global best
7. If the particles are converged then Terminate the QPSO iteration
Else Resample
8. end for
end for
9. Update the appearance models
10. Output

4 Experimental Results

The proposed tracking algorithm is implemented in Opencv/Matlab on a PC with AMD A4 CPU (1.9 Ghz) and 2.74 GB memory. For each sequence, the state of the target object is manually set in the first frame. Parameters set as follows (Table 2).

Table 2. Parameter setting

Parameter	p	M	v_{min}	k_{MOG}
Value	0.75	25	0.04* image length/width	3

4.1 Experiment 1

In this section, we conduct the comparison experiments between QPSO based tracking algorithm and the proposed algorithm. One video sequence named "Dog", contains a dog face moving to the left and right with appearance and scale variation. The other dataset is called "Boy" in which a boy moves quickly. All datasets mentioned are available at http://cvlab.hanyang.ac.kr/tracker_benchmark/seq/name.zip. "name" in the URL need to be replaced with the dataset names when you download them.

As shown in Fig. 3, QPSO based tracking algorithm also successfully finish this tracking task. However, the proposed algorithm tracks the target more accurately, which shows a better performance in search for the global optimum in a long tracking task. Compared to the "Dog", the "Boy" dataset are more difficult to be tracked. As can be seen from Fig. 4, when the tracking window of the QPSO based tracker fails to cover the object after the frame 251, the proposed algorithm still locks the target. The result shows that the improved RScQPSO is a relatively robust tracking algorithm.

In order to comprehensively evaluate the result of the two algorithm, we calculate the MSE (mean square error) between the estimated position and the labeled ground truth as well as the rate of successful tracking. Here only the successful frame is taken into consideration to calculate MSE. The comparison result is shown in Table 3. We can see from the tables that our tracker achieves very favorable performance in terms of both accuracy and successful rate.

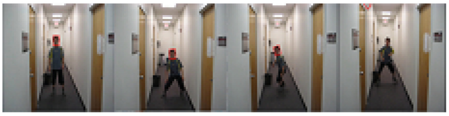
frame 1 frame 368 frame 1000 frame 1200
(a) QPSO based algorithm

frame 1 frame 368 frame 1000 frame 1200
(b) the proposed tracking algorithm

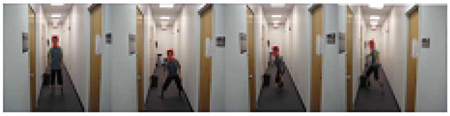
Fig. 3. Tracking performance of "Dog" sequence

4.2 Experiment 2

In order to further evaluate the performance of our algorithm, more dataset are used for testing. The first video "fish", shown in Fig. 5(a), contains a static object with camera motion, which means everything in the sequence is under illumination variation. Despite all this, our algorithm is able to track the target



Frame 1 Frame 85 Frame 173 Frame 251
(a) QPSO based algorithm



Frame 1 Frame 85 Frame 173 Frame 267
(b) the proposed tracking algorithm

Fig. 4. Tracking performance of “Boy” sequence

Table 3. The tracking results of “Dog” and “Boy”

Tracking algorithm	MSE of position		Frame tracked	
	Dog	Boy	Dog	Boy
QPSO	5.951298	6.470549	1200/1200	251/267
Improved RScQPSO	4.837096	6.040748	1200/1200	267/267

(a) dataset: fish

(b) dataset: Jumping

(c) dataset: seq_sb

Fig. 5. The more performance of our algorithm

correctly. The second dataset “Jumping” is a jumping person with a vaguely face. Results in Fig. 5(b), demonstrate that the proposed improved RScQPSO algorithm performs well even though the target undergo fast and abrupt motion. The last dataset, “seq_sb”, describes a person undergoing large pose, appearance, and lighting changes, as well as partial occlusions. The result in Fig. 5(c) shows that our tracker finishes the task successfully.

5 Conclusion

In this paper, we proposed an improved tracking algorithm based on cQPSO and PF. The proposed tracking algorithm can deal with tracking tasks with objects undergoing various environments such as illumination, scale, appearance variation, partial occlusion, abrupt and fast motion. Compared with the color histogram, the MOG (mixture of Gaussian) based appearance model we use can make a better object description. From the experiment results, the improved

RScQPSO based tracker has a favorable performance compared with the QPSO based tracker both in terms of accuracy and efficiency.

Acknowledgement. This work was partially supported by the National Natural Science foundation of China (Grant Nos. 61673194, 61105128, 61170119, 61373055), the Natural Science Foundation of Jiangsu Province, China (Grant No. BK20131106), the Postdoctoral Science Foundation of China (Grant No. 2014M560390), the Fundamental Research Funds for the Central Universities, China (Grant No. JUSRP51410B), Six Talent Peaks Project of Jiangsu Province (Grant No. DZXX-025).

References

1. Kitagawa, G.: Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. J. Comput. Graph. Stat. **1**(1), 1–25 (1996)
2. Isard, M., Blake, A.: CONDENSATION-conditional density propagation for visual tracking. Int. J. Comput. Vis. **29**(1), 5–28 (1998)
3. Liu, J.S., Chen, R., Logvinenko, T.: A theoretical framework for sequential importance sampling with resampling. In: Doucet, A., de Freitas, N., Gordon, N. (eds.) Sequential Monte Carlo Methods in Practice, pp. 225–246. Springer, New York (2001)
4. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: IEEE International Conference on Neural Networks, Proceedings, vol. 4, pp. 1942–1948. IEEE (1995)
5. Zhang, L.: Application of Particle Filtering with Particle Swarm Optimization to Target Tracking. Lanzhou University of Technology (2010)
6. Zhang, C-q, Ge, L., Han, D.: Research on particle swarm particle filter algorithm based on target tracking. Comput. Simul. **31**(08), 392–396 (2014)
7. Clerc, M., Kennedy, J.: The particle swarm - explosion, stability, and convergence in a mul-tidimensional complex space. IEEE Trans. Evol. Comput. **6**(1), 58–73 (2010)
8. Sun, J., Fang, W., Wu, X., et al.: Quantum-behaved particle swarm optimization: analysis of individual particle behavior and parameter selection. Evol. Comput. **20**(3), 349–393 (2012)
9. Sun, J., Feng, B., Xu, W.: Particle swarm optimization with particles having quantum behavior. IEEE 2004 Congress on Evolutionary Computation, pp. 1571–1580 (2004)
10. Fang, W., Sun, J., Ding, Y., et al.: A review of quantum-behaved particle swarm optimization. IETE Techn. Rev. **27**(4), 336–348 (2010)
11. Fang, W., Sun, J., Chen, H., et al.: A decentralized quantum-inspired particle swarm optimization algorithm with cellular structured population. Inf. Sci. **330**, 19–48 (2016)
12. El-Maraghi, T.F.: Robust online appearance models for visual tracking. IEEE Trans. Pattern Anal. Mach. Intell. **25**(10), 1296–1311 (2003)
13. Sun, B., Wang, B., Shi, Y., et al.: Visual tracking using quantum-behaved particle swarm optimization. In: Control Conference. IEEE (2015)