Distributed Color-Based Particle Filter for Target Tracking in Camera Network

Yueqing Jing and Yanming Chen(✉)

Anhui University, Hefei 230000, China
jingyueqing@stu.ahu.edu.cn, cym@ahu.edu.cn

Abstract. Color-based particle filters have appeared in some literatures. However, there are still some important drawback in tracking targets, such as illumination changes, occlusion and low tracking accuracy. To solve these problems, in this paper, we propose a distributed color-based particle filter (DCPF) for target tracking, which can track targets accurately in a large-scale camera network with less data transmission and less computation. Compared with the previous algorithms, the algorithm proposed in this paper has two obvious advantages. First, the DCPF framework merges color features into the target's state to obtain better robustness. Second, it considers the situation where the target is disappear in some cameras because of limited field of view (FoV). Convincing results are confirmed that the performance analysis of the proposed algorithm in this paper is very close to the centralized particle filter method.

Keywords: Consensus algorithm · Particle filter · Camera network

1 Introduction

With the decrease of cost for surveillance equipments, cameras have been deployed on a large scale in our city. They have been widely used in the wide-area surveillance, disaster response, environmental monitoring, etc. [1]. Multiple cameras can observe a wider area and provide different angles of view, which make information fusion possible. In the target tracking application, it may be got a more robust tracking method by information fusion among different cameras. There are two types information fusion methods: the centralized solution and the distributed solution. In the wide-area surveillance, the distributed solution is usually selected because of its scalability to a large number of sensors, easy replacement, and strong robustness. In view of the above reasons, this paper mainly proposes a distributed color-based particle filter (DCPF) for target tracking in camera networks, which can track targets accurately in a large-scale camera network with less data transmission and less computation. our method can deal with target occlusion and target disappearing in some cameras. The term 'distributed' means that each camera processes its own data and interaction information among neighboring nodes. There is no centralized unit.

H. Gao et al. (Eds.): CollaborateCom 2020, LNICST 350, pp. 396–406, 2021.
https://doi.org/10.1007/978-3-030-67540-0_24

There are already some distributed methods, which may encounter some difficulties for applying to camera networks. For example, the data obtained by the camera are video streaming, which are relatively large and not easy to transmit. If a centralized method is adopted, it will be difficult to process, transmit, and store data in a large scale camera network. Therefore the centralized method is not suitable for a large scale camera network. In addition, there is another problem. The camera has a limited field of view. When the data from a single camera, the target may be lost because of the limited field of view. Therefore, we need fuse data from different cameras. However, due to limited processing power, bandwidth, and real-time processing requests, etc., we need adopt a distributed method with less data transmission and less computation. The method proposed in this paper adopts the idea of combining local filtering and fusion filtering, and uses local filtering to process the video to get smaller data. Therefore, it can effectively alleviate the data congestion phenomenon in terms of data transmission, storage, and operation.

Figure 1 depicts a scene application for tracking people in the camera networks. Although some cameras do not observe the target, which can also obtain the estimated state of the target by the distributed algorithm.

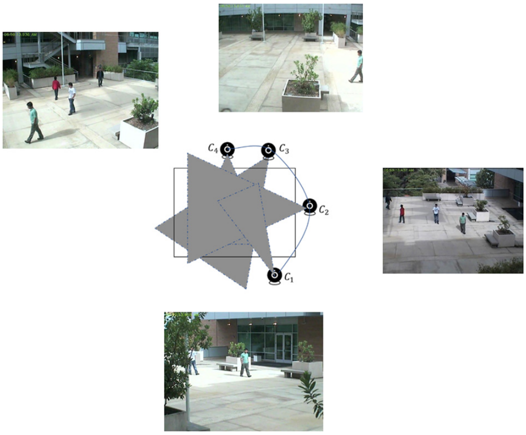

Fig. 1. Cameras' FoVs and network connectivity

The **main contributions** of this paper are proposing a distributed color-based particle filter for target tracking in camera networks, taking special care of the issues of huge data transmission and less computation, and using the proposed algorithm to track target in camera networks.

1.1 Description and Solution of the Problem

Many distributed methods can obtain good tracking accuracy with the observations of each node in sensor networks [2–4], but they cannot handle the scenario shown in Fig. 1, where each node can only observe a small part of the target trajectory. The method proposed by Kamal et al. [5] does not process video data. The method proposed by Mohammadi et al. [6] use Kalman to track multiple targets in video data. The experimental results are quite good. However, this method may not be suitable when the state model is nonlinear. The method proposed by Kamal et al. [7–9] use a single video to track objects, the experimental results are particularly good, and the robustness is extreme. But it is necessary to have observation data for the target, which is very challenging. Video target tracking method needs to solve following issues in camera networks, such as limited field of view, transmission delay, data fusion, and computational overhead for processing video data. The method proposed in this paper directly uses the color feature as the measurements. After local calculation, the processed data is transmitted, which is much smaller than the video data. The method significantly eases the pressure of transmission, storage, bandwidth, and delay. The experimental results show that the accuracy is very close to the centralized particle filter without the communication delay [4].

1.2 Related Work

Distributed particle filter has attracted the attention of many scholars in recent years. It has better performance than the centralized particle filter. The centralized particle filter requires a fusion centre [10], which receive the measurements from all sensor nodes and use them to update their weights. But it has a severe problem that once the fusion centre is broken down, the entire sensor network will be paralyzed. Distributed particle filtering is different. Each sensor node is a fusion centre. It communicates with its neighbors and transmits the processed data. After multiple iterations [5], consensus results are obtained. The consensus method turns each sensor node into a fusion center. Even if a sensor node is broken down, the entire sensor network still maintains regular operation. Many tracking methods based on distributed particle filter have been proposed [3,11,12], but they cannot solve the particular situation related to the camera network. Such as some cameras have no target. Due to delays and weak processing requests, the previous expectations and variances are merged [6]. In cluttered environment, features need to be added to improve tracking accuracy. Such as color, texture, angle, gradient, motion [8,13–16], the color feature is a very convenient feature. The method proposed in this paper solves the problem of some cameras missing targets by adding color features.

The rest of the paper is organized as follows: the Sect. 2 introduces centralized particle filter, distributed particle filter consensus method, and color characteristics. The DCPF algorithm proposed in this paper is described in the Sect. 3. The Sect. 4 is the experimental results. Finally, we summarize the paper in the Sect. 5.

2 Background

We consider a camera network which composed of N nodes. The state variable is $x(k) = [X(k), Y(k), \dot{X}(k), \dot{Y}(k)]^T$. Where $(1 \leq l \leq N)$, T represents the transpose, $(X(k), Y(k))$ is the position, $\dot{X}(k), \dot{Y}(k)$ are the velocity, node l generates a measurement $z^{(l)}(k)$ at time k. The overall space is shown as follows:

$$\text{State model} : x(k) = f(x(k-1)) + \xi(k) \tag{1}$$

$$\text{Observation model} : z^{(l)}(k) = g(x(k)) + \zeta^{(l)}(k) \tag{2}$$

The $\xi(k), \zeta^{(l)}(k)$ represent the noise error of the state model and observation model, respectively.

2.1 Centralized Particle Filter (CPF)

In CPF [4], observations from all sensors are transmitted to a centralized unit. The centralized unit contains n weighted particles $(\chi^{(j)}(k-1), \omega^{(j)}(k-1))(j = 1, ...n)$, where n is the number of particles. It merging the measurements of multiple sensors by increasing the dimensions of the measurements when the centralized unit receives the measurements of the sensor $z^{(l)}(k)$ at time k. It uses CPF to predict particles $\chi^{(j)}(k)$ and update weights $\omega^{(j)}(k)$.

2.2 Distributed Particle Filter

In DPF [6], all sensor nodes are connected to form a network. Each sensor node uses its measurements to update the weight of the particle on the local filter and calculates the expectation and covariance. Transmit $p^{-1} \cdot u$ and covariance p^{-1} to the nodes who direct connected to it. After multiple iterations, the value on each node is consensus. The average consensus algorithm is a popular distributed algorithm to calculate the arithmetic average of $(a_l)_{l=1}^N$ values [5]. In the average consensus algorithm, each node initializes its consensus state $a_l(0) = a_l$, and runs the following equation iteratively.

$$a_l(t) = a_l(t-1) + \epsilon \sum_{j \in \aleph_l} (a_j(t-1) - a_l(t-1)) \tag{3}$$

where $\aleph_l$ represents the set of neighbor nodes for node l, t is the number of iterations, and the parameter ϵ is a number between 0 and $\frac{1}{\triangle_{max}}$. $\triangle_{max}$ is the maximum degree in the sensor networks topology graph. At the beginning of the iteration, node l sends its previous state $a_l(t-1)$ to its neighbor nodes, and also receives the previous state $a_j(t-1)$ of other nodes. It use Eq. (3) to update its state. And the state values of all nodes will converge. It use the consensus values to update the particles weights on the fusion filter. After the weights are updated, the estimated value of the target can also be obtained by calculating the expectations. Distributed particle filtering mainly has the following three advantages: (1) There is no centralized unit; (2) The sensor node does not need global knowledge of the networks topology; (3) Each node only needs to exchange information with its neighboring nodes.

2.3 Color Characteristics

Feature-based color particle filtering is often used to track any type of target [17]. Color, gradient, and other features are often used to track targets. Specifically, color may be the most popular feature of video surveillance and tracking targets. Because it is more effective than motion or geometry under conditions of partial occlusion, rotation, range, and resolution changes. At the end of each filter iteration, its estimates will be used to update the model, and a parameter a is used to control the speed of model update.

$$hist(k+1) = (1-\alpha)hist(k) + \alpha hist_{E[x_k]}(k) \tag{4}$$

where $hist$ represents the color histogram.

3 Distributed Color Particle Filter

DCPF runs two localized particle filters at each node, one is a local filter, and the other is a fusion filter. The implementation of the local filter is very similar to the standard particle filter. The local filter only processes the local observation value $z^{(l)}(1:k)$, as described in Sect. 3.1. The fusion filter uses the local distribution $P(x(k) \mid z^{(l)}(1:k)$ and $P(x(k) \mid z^{(l)}(1:k-1))$ to estimate the global posterior distribution $P(x(0:k) \mid z(1:k))$, as described in Sect. 3.2.

3.1 Local Filter

The model of the local filter is shown as follows

$$\text{State model}: x(k) = f(x(k-1)) + \xi(k) \tag{5}$$

$$\text{Observation model}: z^{(l)}(k) = hist^{(l)}(k) \tag{6}$$

$$\begin{pmatrix} r \\ g \\ b \end{pmatrix} = frame \begin{pmatrix} yMin:yMax & xMin:xMax & 1 \\ yMin:yMax & xMin:xMax & 2 \\ yMin:yMax & xMin:xMax & 3 \end{pmatrix} \tag{7}$$

$$hist = [imhist(r, hist_{bin}); imhist(g, hist_{bin}); imhist(b, hist_{bin})] \tag{8}$$

where $xMin, xMax, yMin, yMax$ are the minimum and maximum values on the x and y axes of a frame. r, g, b represent the color channel of the frame respectively. And $hist_{bin}$ is the size of the bin on each channel. $Hist$ is the color histogram. Each node has a local filter. The details of the local filter are as follows:

(i) First, the particles $\chi_j^{(l,lf)}$, weights $\omega_j^{(l,lf)}$ and color histograms $hist_{target}$ on the local filter of each node are initialized. The particles are initialized in world coordinates instead of the coordinates in the video frame, and the state of the target in the video frame needs to be transferred to the world state according to the conversion matrix. And the conversion matrix H can be calculated by the position of the camera.

(ii) According to the proposal distribution $q(\chi_j(k) \mid \chi_j(0:k-1), z^{(l)}(1:k))$ propagate particles $\chi_j^{(l,lf)}(k-1) \rightarrow \chi_j^{(l,lf)}(k)$

(iii) Estimate the position of the target in this node according to the propagated particle $\chi_j^{(l,lf)}(k)$ and the not updated weight $\omega_j^{(l,lf)}(k-1)$

$$\text{Expection}: v^{(l)}(k) = \sum_{j=1}^{n} \chi_j^{(l,lf)}(k)\omega_j^{(l,lf)}(k-1) \tag{9}$$

$$\text{Covariance}: r^{(l)}(k) = \sum_{j=1}^{n} \omega_j^{(l,lf)}(k-1)(\chi_j^{(l,lf)}(k)-v^{(l)}(k))(\chi_j^{(l,lf)}(k)-v^{(l)}(k))^T \tag{10}$$

(iv) Use the measurements $z^{(l)}(k)$ to update the weight of node l. Because of its importance, many literatures have been devoted to optimizing these weights. The method proposed by Zhang et al. [8] use correlation filter to optimize the weight of particles. The method proposed by Rincon et al. [15] uses linear Kalman to optimize the target color coefficient to get the desired weight. The method proposed by Zhang et al. [18] using the particle swarm genetic optimization method to optimize particles and remove useless particles. In this paper, the color histogram is used to optimize the weight of particles.

1) First, use the conversion matrix H to find the particle $\chi_j^{(l,lf)}$ in the video frame.
2) Using the converted particle state Eq. (5) and measurement model Eq. (6) to obtain the color histogram of the particle and normalize it.

$$hist = hist/sum(hist) \tag{11}$$

3) Calculate the particle color histogram and target color histogram according to the proposed distribution. Then combine the following equation to calculate the particles' weight.

$$\omega_j^{(l,lf)}(k) \propto \omega_j^{(l,lf)}(k-1)\frac{P\left(z^{(l)}(k) \mid \chi_j^{(l,lf)}(k)\right)P\left(\chi_j^{(l,lf)}(k) \mid \chi_j^{(l,lf)}(k-1)\right)}{q\left(\chi_j^{(l,lf)}(k) \mid \chi_j^{(l,lf)}(0:k-1), z^{(l)}(1:k)\right)} \tag{12}$$

(v) Estimate the position of the target in node l according to the propagated particle $\chi_j^{(l,lf)}(k)$ and the updated weight $\omega_j^{(l,lf)}(k)$.

$$\text{Expection}: u^{(l)}(k) = \sum_{j=1}^{n} \chi_j^{(l,lf)}(k)\omega_j^{(l,lf)}(k) \tag{13}$$

$$\text{Covariance}: p^{(l)}(k) = \sum_{j=1}^{n} \omega_j^{(l,lf)}(k)(\chi_j^{(l,lf)}(k) - u^{(l)}(k))(\chi_j^{(l,lf)}(k) - u^{(l)}(k))^T \tag{14}$$

3.2 Fusion Filter

The process of obtaining the global posterior distribution $P(x(0:k) \mid z(1:k))$ by the fusion filter is similar to the local filter.

(i) First, the local filter on each node is initialized. The particle $\chi_j^{(l,ff)}$ and the weight $\omega_j^{(l,ff)}$ are similar to the local filter.

(ii) According to the proposal distribution $q(\chi_j(k) \mid \chi_j(0:k-1), z^{(l)}(1:k))$ propagate particles $\chi_j^{(l,ff)}(k-1) \rightarrow \chi_j^{(l,ff)}(k)$

(iii) When all nodes get $u^{(l)}(k), p^{(l)}(k), v^{(l)}(k), r^{(l)}(k)$ at time k, the consensus algorithm is executed.
Initialize the consensus state for node l $X_{c1}^{(l)}(0) = (p^{(l)}(k))^{-1}, x_{c2}^{(l)}(0) = (p^{(l)}(k))^{-1}u^{(l)}(k), X_{c3}^{(l)}(0) = (r^{(l)}(k))^{-1}, x_{c4}^{(l)}(0) = (r^{(l)}(k))^{-1}v^{(l)}(k)$. Then use Eq. (3) to achieve consensus. Once the consensus is reached, the average value of $u^{(l)}(k)$, $p^{(l)}$ (k), $v^{(l)}(k)$, $r^{(l)}(k)$ can be obtained by the following equation

$$p(k) = \frac{1}{N} \lim_{t \to \infty} \left\{ X_{c1}^{(l)}(t)^{-1} \right\} \tag{15}$$

$$u(k) = \lim_{t \to \infty} \left\{ X_{c1}^{(l)}(t)^{-1} \times x_{c2}^{(l)}(t) \right\} \tag{16}$$

$v(k)$ and $r(k)$ are calculated in the same way as $u(k)$ and $p(k)$.

(iv) The calculation of the weight is shown as follow

$$\omega_j^{(l,ff)}(k) = \omega_j^{(l,ff)}(k-1) \frac{\mathcal{N}\left(\chi_j^{(l,ff)}(k); u(k), p(k)\right) P\left(\chi_j^{(l,ff)}(k) \mid \chi_j^{(l,ff)}(k-1)\right)}{\mathcal{N}\left(\chi_j^{(l,ff)}(k); v(k), r(k)\right) q\left(\chi_j^{(l,ff)}(k) \mid \chi_j^{(l,ff)}(k-1), z(k)\right)} \tag{17}$$

It can be seen from Eq. (17) that the calculation of the weight must know the proposal distribution. Use $P(\chi_j^{(l,ff)}(k)\chi_j^{(l,ff)}(k-1))$ as the proposal distribution, which is called SIR filter. It is also the most common and most convenient method. This article uses this proposal distribution.

(v) If the particle degradation is severe, then resample.

Algorithm 1: Fusion filter

Input: given particle set $(\chi_j^{(l,ff)}(k-1), \omega_j^{(l,ff)}(k-1))_{j=1}^{n}$ –fusion filter's particles and their weights

Output: $(\chi_j^{(l,ff)}(k), \omega_j^{(l,ff)}(k))_{j=1}^{n}$ –the updated particles and weights of the fusion filter.

1.for $l = 1 : N$,

2.do $(u^{(l)}(k), v^{(l)}(k), p^{(l)}(k), r^{(l)}(k))$
=local filter$((\chi_j^{(l,ff)}(k-1), \omega_j^{(l,ff)}(k-1))_{j=1}^{n}, z^{(l)}(k))$

3.end for

4.fusion $u^{(l)}(k), P^{(l)}(k)_{l=1}^{N}$ to calculate $u^{(l,ff)}(k), P^{(l,ff)}(k)$

5.for $j = 1 : n, do$,

- Generate particles by proposal distribution sampling $\chi_j^{(l,ff)}(k)$
- Use weight update Eq (12) to calculate $\omega_j^{(l,ff)}(k)$

6.end for

7.observe $(\chi_j^{(l,ff)}(k), \omega_j^{(l,ff)}(k))_{j=1}^{n}$ –if degraded then
= resample $(\chi_j^{(l,ff)}(k), \omega_j^{(l,ff)}(k))_{j=1}^{n}$

8. $(\chi_j^{(l,lf)}(k), \omega_j^{(l,lf)}(k))_{j=1}^{n} = (\chi_j^{(l,ff)}(k), \omega_j^{(l,ff)}(k))_{j=1}^{n}$

4 Experimental Results

We track a single target in a distributed camera network. The number of camera nodes is $N = 4$, non-full connection mode. In some existing distributed particle filter target tracking methods, the processed data is not video data. There are other methods using video frame data for target tracking [9,14,19], the effect is excellent. However, this methods are not distributed, and only processes a single video. The method proposed by Kamal et al. [7] is distributed, the effect is excellent, but the method is based on Kalman, not particle filtering. We use a centralized CPF to evaluate the performance of the DCPF and track pedestrians in multiple environment. Set the target state $x(k) = [X(k), Y(k), \dot{X}(k), \dot{Y}(k)]^T$, the initial velocity is [1, 15], $\epsilon = 0.325$, Q_ξ is covariance of process noise. In order to simplify the calculation, a linear equation is used.

$$f = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad Q_\xi = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{18}$$

$z^{(l)}(k) = hist^{(l)}(k)$, $\quad hist^{(l)}(k)$ is the color histogram of node l at time k.

The video dataset in Scene 1 is provided by [9], where the size of the video frame is 640*480, a total of 58 frames of data are used, and track a person whose upper body is blue. After the 30th frame, the target is occluded by other objects in the C4 camera, and the target does not appear in the field of view of C3 after the 5th frame. The other observation is good. The video dataset in Scene 2 is provided by [20], the size of the video frame is 360*288, a total of 58 frames

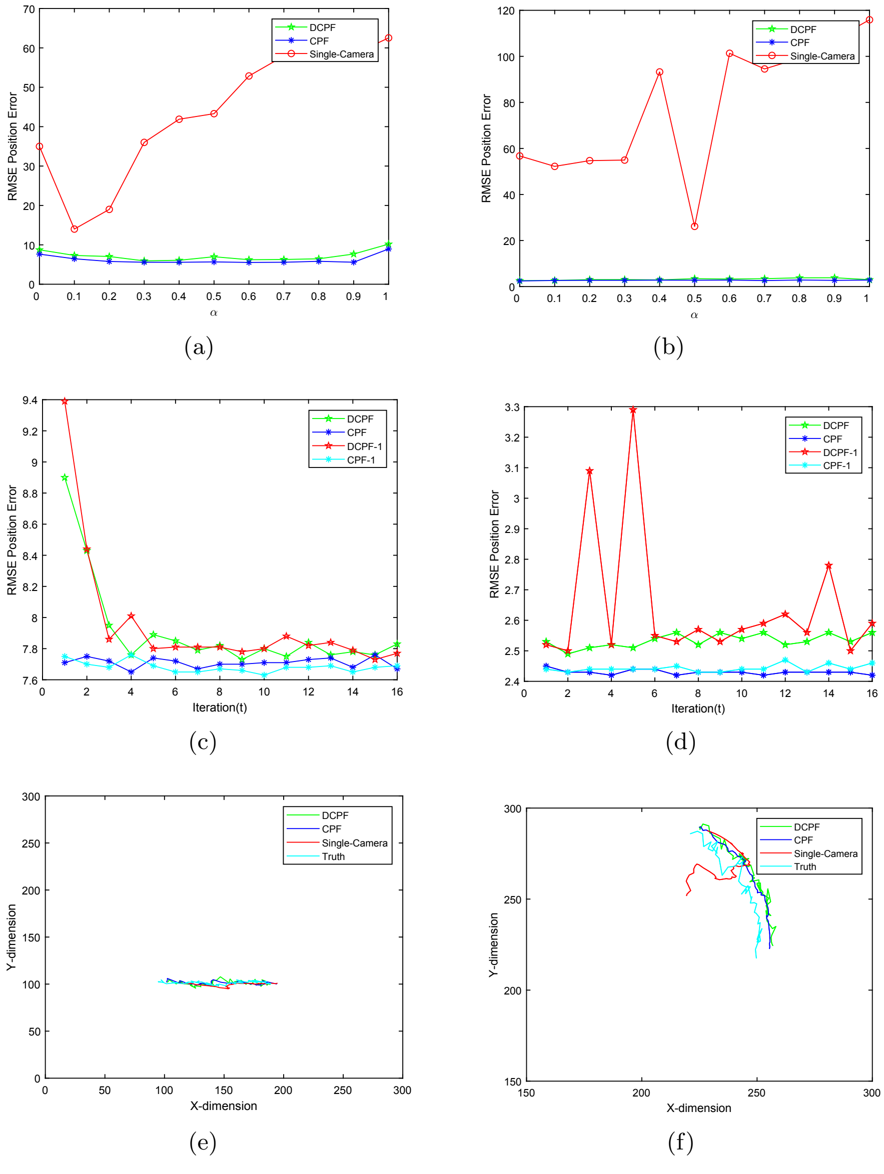

Fig. 2. Scene 1 are the Figs. 2(a), (c), (e) on the left, and Scene 2 are the Fig. 2(b), (d), (f) on the right. Figure 2(a) $n = 500, hist_{bin} = 8, t = 12$, different color coefficient α, error of different methods. Figure 2(c) $hist_{bin} = 8$, $\alpha = 0$, DCPF, CPF represent the error of $n = 500$ repeated running 10 times with different iterations. DCPF-1, CPF-1, represent the error of $n = 300$ repeated running 10 times with different iterations. Figure 2(e) $n = 500$, $hist_{bin} = 8, t = 12$, trajectories of different methods, among which single-camera selects the trajectory with the smallest error. Figure 2(b), (d), (f) is the same as Fig. 2(a), (c), (e).

of data are used. And the person is tracked whose upper body is white. The four cameras have good measurements, and there is no target lost. The tracking result is shown in Fig. 2.

As shown in Fig. 2(a,d), under the condition of different color coefficient α, DCPF has always maintained a low error, while the single camera is greatly affected. The number of DCPF iterations is 12, the number of network nodes is 4, and the result has converged, so the error of DCPF is close to CPF. As shown in Fig. 2(b,e), under the same conditions, the errors of DCPF are similar to CPF. The error among different particle numbers is small. DCPF is stable when the number of particles is 500. When the number of particles is 300, the DCPF algorithm is relatively unstable and error-prone. It can be seen from Fig. 2(c,f) that the single camera is easily affected by the environment, while the DCPF still maintain good performance. The curves of different colors represent the trajectories of the targets in real world coordinates which based on different algorithms. DCPF can keep consistent with the real trajectory in most α and environment, while the single-camera algorithm needs the optimal α and specific environment to keep consistent with the real trajectory. It can be seen from Fig. 2 that DCPF can keep stable and have small errors in various camera networks environment.

5 Summary

We propose a new distributed color particle filter, which combines color features in the measurement model to obtain more accurate performance. We use local filters and fusion filters to propagate particles, update the weights, and estimate the position of the target. Our measurements is the video frame data. Using the color histogram, we can accurately estimate the position of the target in the picture. In this way, the position of the target in the video frame can be obtained at every moment. Besides, distributed color particle filtering can effectively solve the problems of occlusion, and the target is missing in some cameras. Although the performance of the distributed color particle filter proposed in this paper is excellent, it cannot effectively solve the scale problem. If the size of the target changes rapidly in all cameras, DCPF may fail. In the future, we will work to expand to the problem of target size changes.

Acknowledgment. This work is supported in part by the National Natural Science Foundation of China under Grand (No. 6180200).

References

1. Li, S., Da Li, X., Zhao, S.: 5G Internet of Things: a survey. J. Ind. Inf. Integr. **10**, 1–9 (2018)
2. Wolf, M., Schlessman, J.: Distributed smart cameras and distributed computer vision. In: Bhattacharyya, S.S., Deprettere, E.F., Leupers, R., Takala, J. (eds.) Handbook of Signal Processing Systems, pp. 361–377. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-91734-4_10

3. Cetin, M., et al.: Distributed fusion in sensor networks. IEEE Signal Process. Mag. **23**(4), 42–55 (2006)
4. Gu, D., Sun, J., Hu, Z., Li, H.: Consensus based distributed particle filter in sensor networks. In: 2008 International Conference on Information and Automation, pp. 302–307. IEEE (2008)
5. Kamal, A.T., Farrell, J.A., Roy-Chowdhury, A.K.: Information weighted consensus filters and their application in distributed camera networks. IEEE Trans. Autom. Control **58**(12), 3112–3125 (2013)
6. Mohammadi, A., Asif, A.: Distributed particle filter implementation with intermittent/irregular consensus convergence. IEEE Trans. Signal Process. **61**(10), 2572–2587 (2013)
7. Kamal, A.T., Bappy, J.H., Farrell, J.A., Roy-Chowdhury, A.K.: Distributed multi-target tracking and data association in vision networks. IEEE Trans. Pattern Anal. Mach. Intell. **38**(7), 1397–1410 (2015)
8. Zhang, T., Liu, S., Changsheng, X., Liu, B., Yang, M.-H.: Correlation particle filter for visual tracking. IEEE Trans. Image Process. **27**(6), 2676–2687 (2017)
9. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2544–2550. IEEE (2010)
10. Collins, R.T., Lipton, A.J., Fujiyoshi, H., Kanade, T.: Algorithms for cooperative multisensor surveillance. Proc. IEEE **89**(10), 1456–1477 (2001)
11. Presti, L.L., Sclaroff, S., La Cascia, M.: Path modeling and retrieval in distributed video surveillance databases. IEEE Trans. Multimed. **14**(2), 346–360 (2011)
12. Onel, T., Ersoy, C., Deliç, H.: On collaboration in a distributed multi-target tracking framework. In: 2007 IEEE International Conference on Communications, pp. 3265–3270. IEEE (2007)
13. Yu, J.Y., Coates, M.J., Rabbat, M.G., Blouin, S.: A distributed particle filter for bearings-only tracking on spherical surfaces. IEEE Signal Process. Lett. **23**(3), 326–330 (2016)
14. Han, B., Comaniciu, D., Zhu, Y., Davis, L., et al.: Incremental density approximation and kernel-based Bayesian filtering for object tracking. CVPR **1**, 638–644 (2004)
15. Martínez-del Rincón, J., Orrite, C., Medrano, C.: Rao-blackwellised particle filter for colour-based tracking. Pattern Recognit. Lett. **32**(2), 210–220 (2011)
16. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking. IEEE Trans. Signal Process. **50**(2), 174–188 (2002)
17. Khan, Z., Balch, T., Dellaert, F.: A rao-blackwellized particle filter for eigentracking. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004, vol. 2, pp. II. IEEE (2004)
18. Zhang, Y., Wang, S., Li, J.: Improved particle filtering techniques based on generalized interactive genetic algorithm. J. Syst. Eng. Electron. **27**(1), 242–250 (2016)
19. Han, B., Zhu, Y., Comaniciu, D., Davis, L.: Kernel-based Bayesian filtering for object tracking. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), vol. 1, pp. 227–234. IEEE (2005)
20. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multicamera people tracking with a probabilistic occupancy map. IEEE Trans. Pattern Anal. Mach. Intell. **30**(2), 267–282 (2007)