HIGH TECHNOLOGY LETTERS | Vol. 29 No. 2 | Jun. 2023 | pp. 113-121

doi:10.3772/j.issn.1006-6748.2023.02.001

Vehicle recognition and tracking based on simulated annealing chaotic particle swarm optimization-Gauss particle filter algorithm①

WANG Weifeng(王伟峰)*, YANG Bo*, LIU Hanfei*, QIN Xuebin②**

(*School of Safety Science and Engineering, Xi'an University of Science and Technology, Xi'an 710054, P. R. China)

(**School of Electrical and Control Engineering, Xi'an University of Science and Technology, Xi'an 710054, P. R. China)

Abstract

Target recognition and tracking is an important research filed in the surveillance industry. Traditional target recognition and tracking is to track moving objects, however, for the detected moving objects the specific content can not be determined. In this paper, a multi-target vehicle recognition and tracking algorithm based on YOLO v5 network architecture is proposed. The specific content of moving objects are identified by the network architecture, furthermore, the simulated annealing chaotic mechanism is embedded in particle swarm optimization-Gauss particle filter algorithm. The proposed simulated annealing chaotic particle swarm optimization-Gauss particle filter algorithm (SA-CPSO-GPF) is used to track moving objects. The experiment shows that the algorithm has a good tracking effect for the vehicle in the monitoring range. The root mean square error (RMSE), running time and accuracy of the proposed method are superior to traditional methods. The proposed algorithm has very good application value.

Key words: vehicle recognition, target tracking, annealing chaotic particle swarm, Gauss particle filter (GPF) algorithm

0 Introduction

In the intelligent monitoring system, the video data flow for image processing and target analysis are executed by computer vision technology, which is further used to identify the target's actions and track the target. Multi-target tracking technology has become one of the hot spots in computer vision. The research involves the computer image processing, pattern recognition, artificial intelligence, automatic control, and many other related fields of knowledge. Multi-target tracking technology in the intelligent video surveillance system requires high real-time and stability of the system. How to adapt to a variety of complex scenes, high robustness, real-time multi object tracking technology has become a problem which needs to be solved urgently in the current research field.

In this paper, deep learning is used to identify vehicle, and then achieves multi-target tracking. The current vehicle tracking algorithm is mainly divided into the following categories. (1) Region matching tracking method[1], which is required to provide a target template before tracking. (2) Feature matching tracking algorithm[2], which requires feature extraction and feature matching algorithm. (3) Contour matching tracking algorithm[3], the closed curve is used to express the target profile. (4) Optical flow tracking algorithm[4], which needs to establish motion difference of the target and the background to carry on the optical flow segmentation for the target. (5) Model matching tracking algorithm[5]. (6) 3D-tracing method[6], which needs to construct the 3D model to track the target. However, the complexity of the algorithm is relatively high. At present, network construction and feature extraction based on deep learning is one of the hot spots research in target and behavior recognition. A vehicle color recognition method using lightweight convolutional neural network is proposed[7]. Deep learning technology has been rapidly developed. In the industrial sector, Google, Microsoft, Facebook, Baidu, Alibaba, Tencent and other domestic and foreign well-known IT companies in deep learning field

① Supported by the National Key R & D Plan of China (2021YFE0105000), the National Natural Science Foundation of China (52074213), Shaanxi Key R & D Plan Project (2021SF-472) and Yulin Science and Technology Plan Project (CXY-2020-036).

② To whom correspondence should be addressed. E-mail: qinxb@xust.edu.cn.

Received on July 8, 2022

have invested a lot of manpower and material resources, and have acquired great effectiveness. Deep learning emphasizes the deep of the model structure, such as, optimized GoogleNet architecture[8], 16-layer VGG network architecture[9], YOLO v3 network architecture[10]. Through layer by layer feature transform, the samples in the original space of the feature are mapped to a new feature space, so that the classification or prediction is more accurate. Compared with the traditional construction methods, the use of deep learning combined with the characteristics of large data learning is able to portray the inherent information of the data. Deep learning mainly includes a supervised learning machine learning model and an unsupervised learning machine learning model.

Ref. [11] proved that the chaotic particle swarm optimization (CPSO)-based Gaussian graph is the most convenient optimization method in the formation of hybrid neural networks, and has good reliability convergence in the optimization of global functions. Ref. [12] proposed a local chaotic particle swarm optimization algorithm based on a piecewise polynomial interpolation function. This algorithm improves the local convergence problem of the traditional particle swarm optimization algorithm and the improved learning factor particle swarm optimization algorithm while completing the convergence in the early or middle stage of the search. Ref. [13] proposed a switched iterative square root Gauss-Hermite filter, and this algorithm can quickly obtain more accurate state estimation and has a low computational cost and strong robustness. Ref. [14] proposed an improved flight conflict detection algorithm based on the Gauss-Hermite particle filter (GHPF), which outperforms the standard particle filter in conflict detection and tracking accuracy. Ref. [15] proposed a residual life prediction method for lithium-ion batteries based on the GHPF. The results show that the remaining useful life (RUL) prediction method has better prediction performance and higher prediction accuracy than the standard particle filter (PF) based method.

First of all, target content is identified by YOLO v5 network[16-17] in this paper, particle swarm optimization algorithm (PSO) is an effective global optimization based on swarm intelligence to get optimal solution by iterative search, which is widely used in engineering[18-19]. Gaussian particle filtering (GPF) is a resembling particle filter algorithm. In the PSO-GPF algorithm, the GPF sampling process introduces the latest measurements by using PSO algorithm, the sampling distribution moves to the high posterior probability region, and then improves the exchange of information between particle swarm and increases the estimation accuracy of the multi-model distribution[20]. In order to accelerate the convergence of the model, the simulated annealing chaos mechanism is embedded in PSO-GPF, so the method is defined as simulated annealed chaotic particle swarm optimization algorithm (SA-CPSO)-GPF, which is used to optimize the Gauss particle filtering and realize the multi-object tracking.

The rest of this paper is organized as follows. In Section 1, the related research is introduced, including introduction of YOLO v5 and Gaussian particle filtering-particle swarm optimization (GPF-PSO). In Section 2, the proposed method is elaborated in detail. The experimental results are given in Section 3. Section 4 concludes the research.

1 Related research

1.1 YOLO v5 network structure

YOLO v5 network not only inherits the previous YOLO series algorithms, but also puts forward corresponding improvements in some structures. Input: in the model training stage, some improvement ideas are proposed, mainly including mosaic data enhancement, adaptive anchor box calculation and adaptive picture scaling; Benchmark network: integrate some new ideas in other detection algorithms, mainly including focus structure and CSP structure; Neck network: in the target detection network, some layers are often inserted between the backbone and the last head output layer, and the FPN + PAN structure is added in YOLO v5; Head output layer: the anchor frame mechanism of the output layer is the same as that of YOLO v4. The main improvement is the loss function GIOU during training process.

Input represents picture. The input image size of the network is 608×608 pixels. This stage usually includes an image preprocessing stage, that is, the input image is scaled to the input size of the network and normalized. Benchmark network is usually a network of classifiers with excellent performance. This module is used to extract some general feature representations. Neck network is usually located in the middle of refer-

ence network and head network. It can further improve the diversity and robustness of features. Head output is used to complete the output of target detection results. An example of YOLO v5 structure is shown in Fig. 1.

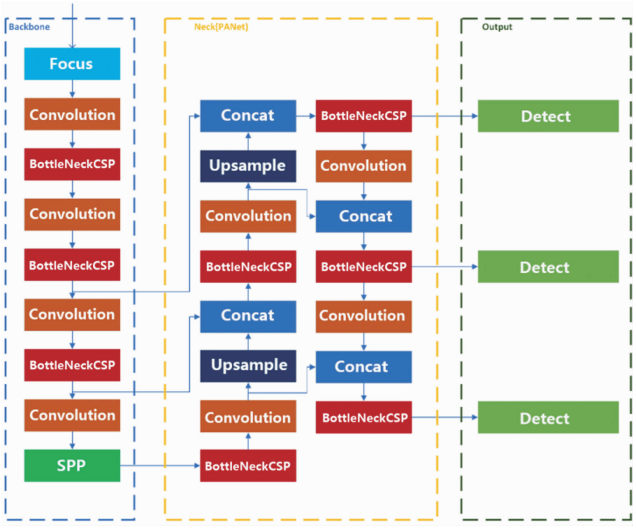


Fig. 1 The structure of YOLO v5 network

1.2 SA-CPSO

1.2.1 PSO

Conventional PSO is a computational method which obtains global optimization.

PSO is expressed as follows: presume a space is D-dimension and the scale of particles are m. The position of the ith particle is $X_i = (X_{i1}, X_{i2}, \cdots, X_{iD})$. The best position of the ith particle in the 'flying' history is $p_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$, and the best value of $p_i (i = 1, 2, \cdots, m)$ is located at p_g, the velocity of the ith particle is the vector $\boldsymbol{v}_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$; the position of ith particle will change according to the following equations.

$$v_{id}(t+1) = wv_{id}(t) + c_1 r_1 (p_{id}(t) - X_{id}(t)) + c_2 r_2 (p_{gd}(t) - X_{id}(t)) \quad (1)$$

$$X_{id}(t+1) = X_{id}(t) + v_{id}(t+1) \quad (2)$$

Rand function is uniform distribution of random number between (0, 1). w is inertia coefficient, c_1 and c_2 are positive learning factor. All the particles move to the position of the best particle by a fitness function. The fitness function is defined as

$$fitness = \exp\left[-\frac{1}{2R_k} (Z_{\text{New}} - Z_{\text{Pred}})^2 \right] \quad (3)$$

where, R_k is measurement noise variance, Z_{New} and Z_{Pred} are the newest measurement value and predictive value.

1.2.2 Simulated annealed chaotic function

The traditional neural network is probably not global-minima, but local-minima in the training process. So chaotic simulated annealing is used to escape from local-minima and get global-optimal solution by the annealed chaotic mechanism in neural network. The energy function of the neural network demonstrates the convergence process. The related research is proposed in Refs [21, 22]. A single chaotic dynamics neuron-annealing model is shown in Eqs (4 – 5). The nested loop process of p and q is shown in Fig. 2.

$$p(t) = \frac{1}{1 + e^{-q(t)/\lambda}} \quad (4)$$

$$q(t+1) = \mu q(t) - E + T(t)(p(t) - I_0) \quad (5)$$

where, $p(t)$ is transient state of the interconnection strength between input neurons and output neurons, $q(t)$ is internal state of the interconnection strength between input neurons and output neurons, I_0 is input bias for each neuron, μ is damping factor of nerve membrane ($0 \leqslant \mu \leqslant 1$), E is energy function of the neural network, and λ is steepness factor of the output function ($\lambda = 0$).

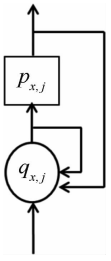
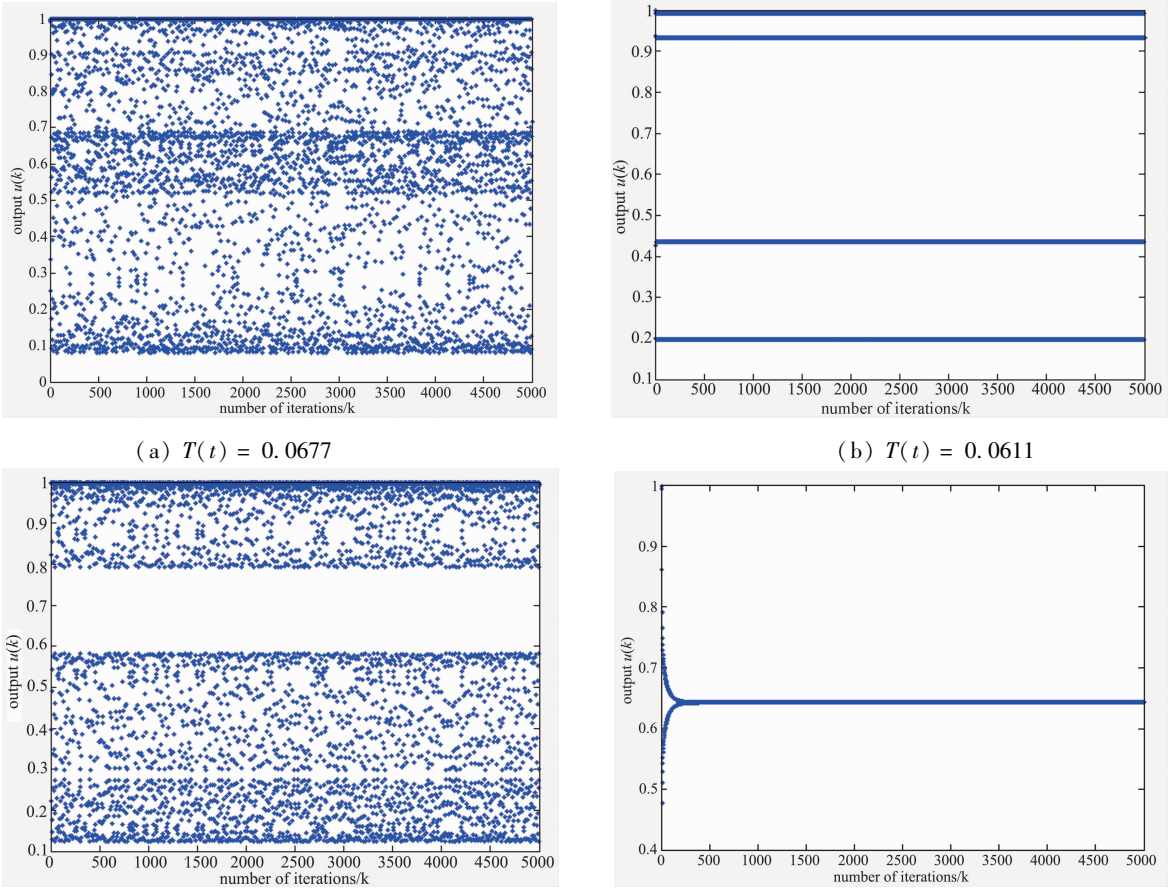


Fig. 2 The relationship between transient state and internal state

$T(t)$ is self-feedback connection weight for input neurons and output neurons. $p(t)$ is expressed by a value of the self-feedback connection weight $T(t)$ in Eqs (4) and (5). The various bifurcation states are demonstrated for the weight $T(t)$ during 5000 iteration in Fig. 3. The initial value of weight T is 0.0677. While $T <$

0.0677, the transient state $p(t)$ shows a process from chaotic state through periodic bifurcation to a steady-state. The chaotic function converges with the decrease of $T(t)$ gradually, where the initial condition is $\lambda = 0.004$, $\mu = 0.899$, $I_0 = 0.649$. The chaotic behavior is used in a neural network. An annealed function is used to converge to a stable equilibrium point for a dynamic weight $T(t)$.

Fig. 3 shows the time evolution of output $p(t)$ and annealing process $T(t)$. The initial value for single neuron is $I_0 = 0.65$, $T_0 = 0.09$, $y_0 = 0.51$, $\lambda = 0.004$, $\mu = 0.899$, $\alpha = 0.9998$, $\beta = 450$[21]. $p(t)$ can converge to a steady-state value. It shows the process of a number of iterations and bifurcation of chaotic dynamics. Exponential damping of $T(t)$ is a process of simulated annealing Eq. (6). The dynamic structure embeds into the competitive learning network in the experiment. Furthermore, the initial value of the parameters influences the dynamics process in training network. The above selected parameters were valid for all the bifurcation processes. The experiment shows the annealed chaotic mechanism can converge rapidly in the competitive network. Fig. 4 (a) demonstrates the output of a single neuron $p(t)$. Fig. 4 (b) demonstrates annealing process of damping variable.



(a) $T(t) = 0.0677$ (b) $T(t) = 0.0611$

(c) $T(t) = 0.0533$ (d) $T(t) = 0.0328$

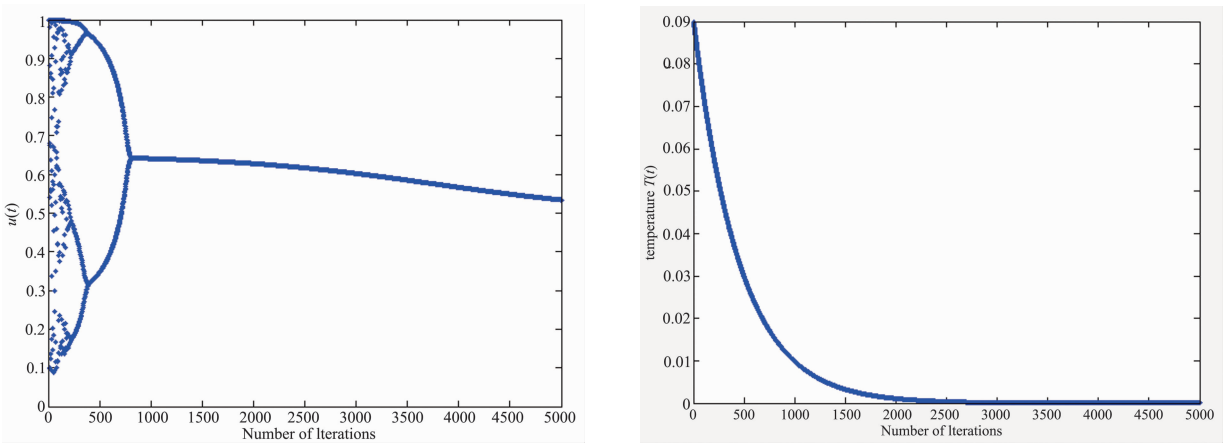
Fig. 3 The various bifurcation states for different $T(t)$ during 5000 iterations

$$T(t+1) = \frac{1}{\beta+1}[\beta + (\tanh(\alpha))^t]\,T(t)$$

$t = 0, 1, 2, 3, \cdots$ (6)

T is self-feedback connection weight or refractory strength ($T > 0$). α and β are constant. In Eq. (6), α is a constant near to 1, t is numbers of iteration. The cooling speed of Eq. (6) is much quicker than the cooling speed of traditional cooling process function[20]. ω in Eq. (1) is replaced with Eq. (6), the timing inertia factor is constructed in Eq. (1).

Simulated annealing mechanism is widely used in iterative heuristic algorithm, which is a process of tending to steady state gradually. On basis of the ideas of simulated annealing, a annealing process function is introduced to give a quantitative description that the process is tending to steady state with the reduction of numbers of iteration.



(a) Convergence process of single neuron $p(t)$ (b) Self-feedback connection weight $T(t)$

Fig. 4 $T(t)$ during 5000 iteration, or the damping variable corresponding to the temperature in the annealing process ($\alpha = 0.9998$, $\beta = 450$, $E = 0$)

1.3 PSO-GPF

Particle swarm optimization-Gauss particle filter algorithm (PSO-GPF) uses particle swarm optimization algorithm to update the parameters of the Gaussian suggested distribution and solve particle degradation and particle accuracy[23]. The implementation steps are as follows.

Step 1 Obtain measurement value by fitness function.

Step 2 Initialize the position and velocity for each particle.

Step 3 Update time. Extract samples from transition density function.

Step 4 Update measurement value. Calculate fitness value for each particle, the position and fitness of each particle are stored.

Step 5 Update the speed and position of each particle according to optimal value, and those particles are close to the true state.

Step 6 Compare the current fitness value of each particle with the fitness value of the individual history best position.

Step 7 Compare the individual optimum and global optimum of the current fitness value, and update the global optimum value.

Step 8 If the stop condition is satisfied, the search stops, otherwise returns to Step 5, continue to search the optimal value.

Step 9 Importance sampling: use the PSO algorithm to generate the parameters of the proposed distribution and extract samples.

Step 10 Calculate the weight of each particle.

Step 11 Normalize weight for each particle.

Step 12 Estimate the weight of the mean, covariance, and double Gaussian sum of the filter distribution.

2 Proposed method

In this section, first of all, the moving object-vehicles are identified by YOLO v5, and then, multi-object tracking SA-CPSO-GPF algorithm is proposed. At last, the details on the implementation of the proposed algorithm are explained.

2.1 Image preprocessing

In this section the targets are identified by YOLO v5. The target recognition regions are extracted firstly. Image preprocessing is the basic step of the whole algorithm. The specific implementation is divided into the following steps.

Step 1 Monocular camera calibration, that is, internal parameter calibration[24].

Step 2 The video stream is obtained by monocular camera.

Step 3 Gaussian filtering is used to smooth each frame of video stream.

2.2 Target recognition by YOLO v5

The preprocessing result is used in this section. That is, the detected rectangle region is regarded as input layers of YOLO v5, which is fine tuning the trained YOLO v5 model in ILSVRC 2012 dataset. The output categories of YOLO v5 is set to 2, which expresses positive example and negative example of vehicle. When the output sample is positive, the target is a vehicle, otherwise, it is not a vehicle.

2.3 The proposed SA-CPSO-GPF tracking algorithm

Conventional particle filter is expressed by suboptimal gain function. So the sampling process of particles are suboptimal. In the case of multi-peak distribution, Gauss particle filter has accuracy problems. In order to optimize the sampling process of Gauss particle filter and improve the estimation accuracy, PSO algorithm is embedded into the Gauss particle filter in this study. In the experiment, the simulated annealing chaotic process is embedded into PSO-GPF algorithm, which can effectively solve the problem of the global and local optimization and converge to the optimal solution quickly. Furthermore, the method can simplify the filtering structure and reduce the degradation problem of the particle by Gauss distribution. Implementation steps are as follows.

Step 1 Obtain measurement value by fitness function Eq. (3).

Step 2 Initialize the position and velocity for each particle, set the initial value of the annealing function.

Step 3 Update time. Extract D-dimensional samples of group $N\{X_{k-1}^{ij}\}$ $(i=1, 2, \cdots N; j=1, 2, \cdots, D)$ from transition density function $p(X_k \mid X_{k-1}=X_{k-1}^{ij})$.

Step 4 Update measurement value. Calculate fitness value for each particle according to Eq. (3), the position and fitness of each particle are stored in the P_{best} of the corresponding particle, select the best particle and save the location D_{best}.

Step 5 Each component of chaotic variable X_k makes chaotic motion by Eqs(4 – 5) to obtain a new position point.

Step 6 Update the speed and position of each particle according to optimal value by Eqs(1 – 2), so the particles are close to the true state.

Step 7 Compare the current fitness value of each particle with the fitness value of the individual best position in history, If the current fitness value is better, then the fitness value is regarded as the optimal value of individual history for particle, the current position of particle is regarded as the best position of individual history for the particle.

Step 8 Compare all current P_{best} value and D_{best} value, update D_{best}.

Step 9 If the stop condition (precision or maximum iteration number) is satisfied, the search stops, otherwise returns to Step 5, continue to search the optimal value.

Step 10 Importance sampling. $N(x_k \mid \mu_k, \sum_k)$ is regarded as proposal distribution, and use the parameters of PSO algorithm, extract sample $N\{X_{ij,k}(i=1, 2, \cdots N; j=1, 2, \cdots, D)\}$.

Step 11 Calculate the weight of each particle.

$$w_k^{ij}=p(Z_k \mid x_k^{ij}) \tag{7}$$

Step 12 Normalize weight for each particle.

$$\tilde{w}_k^{ij}=w_k^{ij}\cdot\left(\sum_{i=1}^{N}\sum_{j=1}^{D}w_k^{ij}\right)^{-1} \tag{8}$$

Step 13 Estimate the mean μ_k and covariance $\sum_k$ of particle filter.

$$\mu_k=\sum_{i=1}^{N}\sum_{j=1}^{D}\tilde{w}_k^{ij}x_k^{ij} \tag{9}$$

$$\sum_k=\sum_{i=1}^{N}\sum_{j=1}^{D}\tilde{w}_k^{ij}(x_k^{ij}-\mu_k)(x_k^{ij}-\mu_k)^{\mathrm{T}} \tag{10}$$

3 Experiment results

In the section, first of all, camera's parameters are calibrated by Ref. [25]. A piece of vedio is captured by the sensor. The size of frame is 480 × 640 pixels. The routines were developed on a PC with Inter (R) Core(TM) 33.00 GHz and 3.25 GB RAM, using Ubuntu Linux 14.04 operation system.

3.1 Vehicle identification by YOLO v5

The differential image is obtained, that is, each frame of the captured video subtracts background image. The detected block regions are regarded as input layers of YOLO v5. The method of detected block regions is elaborated in subsection 1.1. YOLO v5 network is used to recognize vehicle, and classification results of YOLO v5 is 2 categories. Here, the output of the YOLO v5 network is adjusted to 2 categories, that is, positive sample and negative sample of vehicle. A sample of image is shown in Fig. 5. The content of the rectangle is recognized as a vehicle by YOLO v5 network. The other detected block regions is regarded as non-vehicle content by the network.

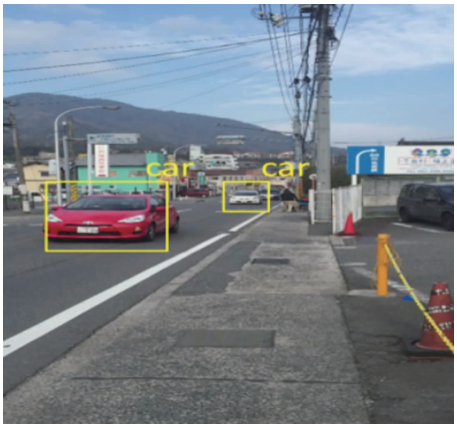


Fig. 5 The result of vehicle recognition by YOLO v5

3.2 Tracking the recognized vehicles by SA-CPSO-GPF

Annealing process is used to obtain the global optimum in the experiment. The detected vehicles are tracked by SA-CPSO-GPF in this section. The different color particles are tracked by SA-CPSO-GPF algorithm, which is elaborated in subsection 3.3. These parameters are set as follows: process noise variance of GPF and PSO-GPF $Q=10$, observed noise variance $R=1$, initial estimation variance $p=5$, time step $t_f=50$, the sample particles of GPF $N=100$; set the parameter of SA-CPSO-GPF: $T_0=0.08$, the number of sample particles $N_p=30$, learning factors $c_1=2.8$, learning factors $c_2=1.3$, time interval 1 s, target initial state $x_0=0$. In order to verify the effectiveness of the proposed algorithm, one-dimensional nonlinear models for GPF, PSO-GPF and SA-CPSO-GPF algorithms are simulated respectively. The discrete state equation and measurement equation of the system are defined as

$$x_k=\alpha x_{k-1}+\beta\frac{x_{k-1}}{1+x_{k-1}^2}+\gamma\cos(1.2(k-1))+w_k \tag{11}$$

$$y_k=\frac{x_k^2}{20}+v_k,\ n=1,2,\cdots,N \tag{12}$$

where, $v_n\sim N(0,\sigma_v^2)$, and the distribution of w_k is specified below. The initial value are $x_0=0.1$, $\sigma_v^2=1$, $\alpha=0.5$, $\beta=25$, $\gamma=8$, $N=100$.

Fig. 6 shows the comparison filtering results of different algorithms under the conditions of $N_P=100$, $D=30$, and $D_{T,\max}=1000$. Fig. 7 shows the comparison of the estimated value errors of different algorithms under the same conditions.

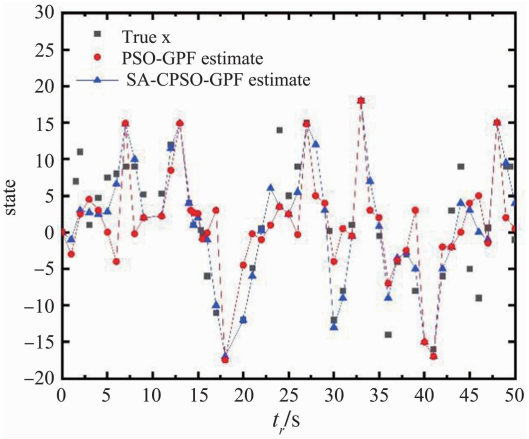


Fig. 6 Comparison of filtering results of different algorithms

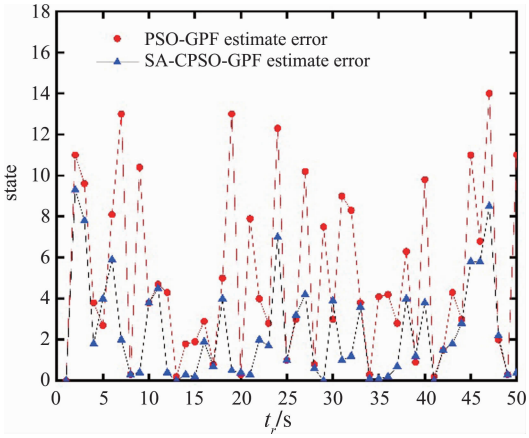


Fig. 7 Comparison of error values of different algorithms

In order to compare the performance of GPF, PSO-GPF and SA-CPSO-GPS in terms of estimation error, root mean square error (RMSE) is defined as the evaluation index, the RMSE formula is

$$RMSE=\sqrt{\sum_{i=1}^{t_t}(\mu_k-x_k)^2 t_f^{-1}} \tag{13}$$

In the experiment, a pieces of video is captured by calibrated camera. The video includes 520 frames. Without loss of generality, the muti-target moving objects start to be tracked from the 150th frame. In Fig. 8, the tracked results are shown, i. e. the 150th frame, the 165th frame, the 180th frame and the 195th frame, respectively. In the experiment, 100 particles are assigned randomly for each detected regions. Different vehicles are assigned different color particles, and then continuous tracking for the particles is achieved. The long line in Fig. 8 indicates tracking trajectories of the vehicles.

Under the conditions of number of sampled particles $N_p=100$, search space dimension $D=30$ and maximum iterations $D_{T,\max}=1000$, the results of different filter algorithms GPF, PSO-GPF and SA-CPSO-GPF are shown in Table 1.

Table 1 Comparison of filtering data among different filtering algorithm

Algorithm	RMSE	Standard deviation	Running time/s
GPF	5.3241	123.7848	0.1812
PSO-GPF	3.2512	90.6529	0.2024
SA-CPSO-GPF	3.0253	68.6576	0.2469

Table 1 shows mean value, standard deviation and running time of the algorithms. The RMSE values for GPF, PSO-GPF and SA-CPSO-GPF are 5.3241, 3.2512 and 3.0253, respectively. The RMSE of the

proposed method is the smallest of the three values. The variance of the covariance gradually decreases. The running times of three algorithms are 0.1812 s, 0.2024 s and 0.2469 s, respectively. The accuracy of SA-CPSO-GPF is higher than the other two methods.

Fig. 8 The vehicle-tracking result by SA-CPSO-GPF

4 Conclusion

In this paper, a multi-targets tracking is presented based on YOLO v5 network architecture with monocular camera in monitor fields. First of all, multi-target vehicles are identified by YOLO v5 network, and then those identified vehicles are tracked by the proposed SA-CPSO-GPF algorithm. The simulation experiment shows the standard deviation and RMSE of SA-CPSO-GPF method is better than traditional methods. In order to prove the validity of the algorithm, the experiment shows the nice performance in continuous video tracking.

In the future, preprocessing image will be refined, and the different moving objects are identified by YOLO v5 networks, and then multi-target and different content tracking will be achieved.

References

[1] ZHANG X X, ZHU X. Efficient and de-shadowing approach for multiple vehicle tracking in aerial video via image segmentation and local region matching[J]. Journal of Applied Remote Sensing, 2020, 14(1):014503.

[2] MATHIVANAN A, PALANISWAMY S. Efficient fuzzy feature matching and optimal feature points for multiple objects tracking in fixed and active camera models[J]. Multimedia Tools and Applications, 2019, 78(19): 27245-27270.

[3] LI C, GAO X. Adaptive contour feature and color feature fusion for monocular textureless 3D object tracking[J]. IEEE Access, 2018, 6(99):30473-30482.

[4] DU B, CAI S, CHEN W. Object tracking in satellite videos based on a multi-frame optical flow tracker[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2019, 12(8):3043-3055.

[5] LIU W, LIU D, FEI B. Optimal matching tracking algorithm based on discriminant appearance model[J]. Pattern Recognition and Artificial Intelligence, 2017, 30(9): 791-802.

[6] LIU D, LIU W, FEI b. Target tracking method based on location-category-matching model[J]. Acta Optica Sinica, 2018, 38(11):224-232.

[7] ZHANG Q, ZHUO L. Vehicle color recognition using multiple-layer feature representations of lightweight convolutional neural network[J]. Signal Processing the Official Publication of the European Association for Signal Processing, 2018, 147:146-153.

[8] CHEN J, LIN X, GAO S. A fast evolutionary learning to optimize CNN[J]. Chinese Journal of Electronics, 2020, 29(6):1061-1073. (In Chinese).

[9] CHEN Q, YAO D, HUANG S. Surveillance video fire detection based on convolutional neural network and optimized GoogleNet architecture[J]. Automation Technology and Application, 2018, 40(9):124-129.

[10] LIN H, YANG J. Ensemble cross-stage partial attention network for image classification[J]. IET Image Processing, 2022, 16(1):102-112.

[11] KOYUNCU H. GM-CPSO:a new viewpoint to chaotic particle swarm optimization via Gauss map[J]. Neural Processing Letters, 2020, 52(9):241-266.

[12] DU Y X, CHEN Y H. Time optimal trajectory planning algorithm for robotic manipulator based on locally chaotic particle swarm optimization[J]. Chinese Journal of Electronics, 2022, 31(5):906-914. (In Chinese).

[13] WANG Y, ZHANG H, MAO X. Switched and Iterated square-root Gauss-Hermite filter for passive target tracking[J]. Circuits, Systems, and Signal Processing, 2018, 37(12):5463-5485.

[14] MA L, GAO Y S, YIN T Y, et al. Improved flight conflict detection algorithm based on Gauss-Hermite particle filter[J]. Wuhan University Journal of Natural Sciences,

2017, 22(3):269-276(In Chinese).
[15] MA Y, CHEN Y, ZHOU X, et al. Remaining useful life prediction of lithium-ion battery based on Gauss-Hermite particle filter[J]. IEEE Transactions on Control Systems Technology, 2019, 27(4):1788-1795.
[16] LIU C, LI J. Self-correcting ship tracking and counting with variable time window Based on YOLO v3[J]. Computer System Application, 2021, 30 (11):240-246.
[17] YAN B, FAN P, LEI X, et al. A real-time apple targets detection method for picking robot based on improved YOLO v5[J]. Remote Sensing, 2021, 13(9): 1619-1627.
[18] GONG Y, ZHANG S, WANG M, et al. USV path planning method based on genetic particle swarm optimization algorithm[J]. Journal of Shandong Jiaotong University, 2019, 30(1):29-37(In Chinese).
[19] DONG J, ZHANG L, ZHANG Y, et al. Prediction of carbon content in BOF steelmaking based on PSO-BP neural network[J]. Journal of North China University of Science and Technology, 2018, 44(1):16-23(In Chinese).
[20] DAI M, LIN P. Gaussian particle filtering using particle swarm optimization[J]. Journal of Fuzhou University, 2015, 43(1):54-60(In Chinese).
[21] QIN X B, DENGJ, WANG M, et al. EEG feature extraction and recognition with different mental states based on wavelet transform and ACCLN network[J]. Journal of Technology, 2017, 32(4):261-274.
[22] LIN J S. Annealed chaotic neural network with nonlinear self-feedback and its application to clustering problem [J]. Pattern Recognition. 2001, 34 (5):1093-1104.
[23] DAI M N, LIN P J, CHENG S Y, et al. Gaussian particle filter using particle swarm optimization[J]. Journal of Fuzhou University: Natural Science Edition, 2015, 43 (1):54-60(In Chinese).
[24] HUANG L, CHEN Y, FAN Z, et al. Measuring the absolute distance of a front vehicle from an in-car camera based on monocular vision and instance segmentation[J]. Journal of Electronic Imaging, 2018, 27(4):043019.
[25] DONG Q, WANG L, FENG J. Confidence-based camera calibration with modified census transform[J]. Multimedia Tools and Applications, 2020, 79(31-32):23093-23109.

WANG Weifeng, born in 1982. He is a professor in the Department of Fire Protection Engineering, School of Safety Science and Engineering, Xi'an University of Science and Technology. He received his Ph. D degree in the Department of Safety Science and Engineering, Xi'an University of Science and Technology in 2019. His main research interests are secure Internet of Things, video recognition, etc.