

Research Article

Moving Vehicle Detection and Tracking Based on Optical Flow Method and Immune Particle Filter under Complex Transportation Environments

Wei Sun ^{1,2}, Min Sun,¹ Xiaorui Zhang,^{2,3} and Mian Li¹

¹School of Automation, Nanjing University of Information Science & Technology, Nanjing 210044, China

²Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, Nanjing 210044, China

³Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing 210044, China

Correspondence should be addressed to Wei Sun; wsun@nuist.edu.cn

Received 23 January 2020; Accepted 6 March 2020; Published 28 April 2020

Guest Editor: Dou Wanchun

Copyright © 2020 Wei Sun et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Video-based moving vehicle detection and tracking is an important prerequisite for vehicle counting under complex transportation environments. However, in the complex natural scene, the conventional optical flow method cannot accurately detect the boundary of the moving vehicle due to the generation of the shadow. In addition, traditional vehicle tracking algorithms are often occluded by trees, buildings, etc., and particle filters are also susceptible to particle degradation. To solve this problem, this paper proposes a kind of moving vehicle detection and tracking based on the optical flow method and immune particle filter algorithm. The proposed method firstly uses the optical flow method to roughly detect the moving vehicle and then uses the shadow detection algorithm based on the HSV color space to mark the shadow position after threshold segmentation and further combines the region-labeling algorithm to realize the shadow removal and accurately detect the moving vehicle. Improved affinity calculation and mutation function of antibody are proposed to make the particle filter algorithm have certain adaptivity and robustness to scene interference. Experiments are carried out in complex traffic scenes with shadow and occlusion interference. The experimental results show that the proposed algorithm can well solve the interference of shadow and occlusion and realize accurate detection and robust tracking of moving vehicles under complex transportation environments, which has the potentiality to be processed on a cloud computing platform.

1. Introduction

In recent years, the popularization of vehicles has caused severe traffic accidents and traffic congestion [1], and it has become necessary to relieve traffic pressure during peak period. Therefore, in a complex transportation environment, detecting the moving vehicles and counting them can be used to reasonably regulate the traffic flow of a certain section of the road or quickly arrange the traffic police to deal with the traffic problems. For many public places with a number of people [2], such as bus stations, railway stations, and airports, if a terrorist incident occurs, it will have unimaginable effects. To prevent the occurrence of dangerous events, such as vehicle intrusion, sudden acceleration, and

overspeed, it is necessary to detect and track the vehicles with abnormal behavior in these public areas and transfer data information to a cloud platform for storage and further analysis by IoT [3–5].

In general, there are mainly four kinds of methods to detect moving vehicles, i.e., interframe difference method, background subtraction method, loop detectors, and optical flow method [6].

For the interframe difference method, Ma Yongjie, et al. used the interframe difference method for vehicle detection [7]. The key to this method is to select an appropriate threshold for image binary obtained. If the threshold is not chosen properly, it will have severe effects on the test results. If it is too big, it cannot detect the moving vehicle, and if it is

too small, it cannot effectively suppress noise in the image. This is one of the disadvantages of the interframe difference method, i.e., it is sensitive to noise. In addition, in the process of moving vehicle detection, the interframe difference method will also cause a “cavity” phenomenon. The reason of the so-called “cavity” is that the speed of the moving vehicle is too slow or the detected vehicle is not moving.

Although the computational complexity of the interframe difference method is low in implementation and it is insensitive to illumination changes, it is only sensitive to moving targets. Therefore, it is often combined with other methods such as the background subtraction method to achieve the desired detection effect. The background subtraction method takes a still image as a comparison image and then implements difference operation between the still image and each frame of image subsequently taken to obtain a difference image, thereby detecting the moving vehicle from the image. Gao Tao et al. used the background subtraction method to detect moving vehicles [8]. Background modeling and updating are the key for the background subtraction method [9, 10]. The background subtraction method is simple to implement, and the threshold can be set according to actual conditions. In addition, the detection results of the method can well reflect various situations of the moving target, such as orientation and size. However, this method is sensitive to external environments and is susceptible to illumination, wind, background motion, and camera shake, which will result in inaccurate detection results. In addition, the threshold in difference operation requires artificial setting, and there is no uniform method for every specific setting. Therefore, this is also one of the disadvantages of the background subtraction method.

For loop detectors [11], the detection of true loop closure in visual simultaneous localization and mapping (vSLAM) can help in relocalization and map registration algorithms, which improves the accuracy of the map and obtains more accurate and consistent results. However, the loop closure detection [12–14] may be affected by many factors, including illumination conditions, seasons, different viewpoints, and mobile objects.

The so-called optical flow refers to the moving speed of pixels of moving objects in a grayscale image. Assume that in the image sequence, every pixel of previous frames of images moves to the same position of current frame image with a certain velocity vector, that is, every pixel on every frame image has a velocity vector. Ideally, the background is constant; therefore, the optical flow of its pixels is zero, and the part with nonzero pixels is the moving target to be detected. The optical flow method has high detection accuracy and can accurately analyze moving targets [15, 16]. At the same time, the optical flow method can also detect moving targets in the case of background motion. Compared with the three methods described above, the optical flow method can obtain more information about moving targets. However, although the optical flow method has high detection accuracy, it cannot obtain an accurate contour of a moving target due to the generation of a shadow.

Although the research on moving vehicle detection has aroused great interest at home and abroad, existing methods cannot still solve the problem of how to accurately detect moving vehicles. Major reason is that the real-world environment is complicated, such as lightning change, roadside railings, and shaking trees, which will have severe effects on the detection results. In addition, in an open environment, because of complex targets and backgrounds, most algorithms cannot meet the requirements of real-time performance and accuracy.

For vehicle tracking, many researchers have conducted in-depth research and have proposed many target tracking algorithms, such as traditional Kalman filter algorithm [17], deep learning [18], Meanshift algorithm [19], clustering algorithm [20], Camshift algorithm [21], aggregation algorithm [22], and particle filter algorithm [23]. In recent years, as deep learning technology has shined in computer vision community, especially since 2015, GPUs have become widespread, and the deep learning algorithm has also been introduced into target tracking. Although many vehicle tracking algorithms have achieved good results, there are still various challenges in terms of the occlusion and changes of vehicle scale and illumination under the complicated traffic environment.

Cloud computing [24, 25] is a type of distributed computing, which refers to the decomposing of huge data computing processing programs into countless small programs through the network [26] “cloud” and then processing and analyzing these early stages of cloud computing through a system of multiple servers [27]. With this technology, we can complete the processing of tens of thousands of data in a short time, thereby achieving powerful network services [28, 29]. The cloud computing not only is a kind of distributed computing but also can provide the facilities of utility computing, load balancing, parallel computing, collaborative filtering [30], network storage, hot backup redundancy, and virtualization result [31].

To address above challenges, we propose a new moving vehicle detection and tracking method based on optical flow and immune particle filtering under complex transportation environments. Firstly, the moving vehicle is roughly detected by the optical flow method, and then the shadow detection algorithm based on HSV color space is used to detect the shadow position by threshold segmentation. Then, according to the detected shadow area, accurate moving vehicle detection is realized by removing shadow area. Finally, immune particle filtering is used as the algorithm framework [32] to realize adaptive motion vehicle tracking [33]. Based on the proposed method, we can do big data analysis [34–37] for vehicle monitoring and transportation management. The algorithm flowchart is shown in Figure 1.

The main contributions of this work are summed up as follows:

- (i) This paper proposes a moving vehicle detection method based on the optical flow method with shadow removal, which can improve the accuracy of moving vehicle detection.

- (ii) A new vehicle tracking framework based on the immune particle filter is proposed, which improves the reliability and robustness of vehicle tracking by improving the particle resampling process.

This paper is organized as follows. Section 2 presents the moving vehicle detection based on the optical flow method. Section 3 explains the proposed moving vehicle detection based on shadow removal. Section 4 discusses the improved immune particle filter for vehicle tracking. Finally, Sections 5 and 6 present a conclusion and some experiment results.

2. Moving Vehicle Detection Based on the Optical Flow Method

There are many ways to calculate the optical flow. According to the theory of optical flow field and mathematical calculation methods, Barron, et al. divided them into four types: gradient-based methods, matching-based methods, energy-based methods, and phase-based methods [38]. Considering that the proposed algorithm is based on the classical Horn–Schunck (HS) algorithm, the HS algorithm is introduced first. The flowchart of the HS algorithm is shown in Figure 2.

The HS method has four assumptions: (i) the grayscale value of the image is always constant; (ii) the change of the optical flow satisfies certain constraints; (iii) the pixel's changes in the image are slow or unchanged, that is, the change of the optical flow in the image is smooth; and (iv) the overlap of objects in the image is ignored.

Based on the assumptions above, the optical flow equation can be obtained as follows:

$$f_x u + f_y v + f_t = 0, \quad (1)$$

where $f_x = \partial f(x, y, t)/\partial x$, $f_y = \partial f(x, y, t)/\partial y$, $f_t = \partial f(x, y, t)/\partial t$, $u = \partial x/\partial t$, and $v = \partial y/\partial t$.

Next introduce the velocity smooth condition, as shown in the following equation:

$$\zeta_d^2 = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2. \quad (2)$$

Based on the optical flow constraint equation (1) and the velocity smoothing condition, the minimization equation is established:

$$\zeta^2 = \iint \mu^2 \zeta_d^2 + (f_x u + f_y v + f_t)^2 dx dy, \quad (3)$$

where μ represents the smoothing weight coefficient [39], which refers to the proportion of the speed smooth term.

Then, using the variational calculation [40], the Laplacian operator can be obtained approximately by making difference between the velocity of a pixel point and the average velocity of its surrounding pixel points. Thus, we can use the iterative method to solve the optical flow equation (1), where n represents the number of iterations.

But before solving equation (1), we need to determine two parameters, \bar{u} and \bar{v} . The two parameters can be calculated by the nine-point difference algorithm [41]. Then, calculate the grayscale gradient values, f_x , f_y , and f_t :

$$\begin{aligned} u^{n+1} &= \bar{u}^n - \frac{f_x(f_x \bar{u}^n + f_t + f_y \bar{v}^n)}{\lambda + (f_x^2 + f_y^2)}, \\ v^{n+1} &= \bar{v}^n - \frac{f_y(f_x \bar{u}^n + f_t + f_y \bar{v}^n)}{\lambda + (f_x^2 + f_y^2)}, \\ \bar{u}_{i,j,k} &= \frac{1}{6}(u_{i,j-1,k} + u_{i-1,j,k} + u_{i,j+1,k} + u_{i+1,j,k}) \\ &\quad + \frac{1}{12}(u_{i-1,j-1,k} + u_{i-1,j,k} + u_{i+1,j+1,k} + u_{i+1,j-1,k}), \\ \bar{v}_{i,j,k} &= \frac{1}{6}(v_{i,j-1,k} + v_{i-1,j,k} + v_{i,j+1,k} + v_{i+1,j,k}) \\ &\quad + \frac{1}{12}(v_{i-1,j-1,k} + v_{i-1,j,k} + v_{i+1,j+1,k} + v_{i+1,j-1,k}), \\ f_x &= \frac{1}{4}(f_{i,j+1,k} - f_{i,j,k} + f_{i+1,j+1,k} - f_{i+1,j,k} \\ &\quad + f_{i,j+1,k+1} - f_{i,j,k+1} + f_{i+1,j+1,k+1} - f_{i+1,j,k+1}), \\ f_y &= \frac{1}{4}(f_{i+1,j,k} - f_{i,j,k} + f_{i+1,j+1,k} - f_{i,j+1,k} \\ &\quad + f_{i+1,j,k+1} - f_{i,j,k+1} + f_{i+1,j+1,k+1} - f_{i,j+1,k+1}), \\ f_t &= \frac{1}{4}(f_{i,j,k+1} - f_{i,j,k} + f_{i,j+1,k+1} - f_{i,j+1,k} \\ &\quad + f_{i+1,j,k+1} - f_{i+1,j,k} + f_{i+1,j+1,k+1} - f_{i+1,j+1,k}). \end{aligned} \quad (4)$$

After determining all the parameters, we can solve the velocity field based on the current-frame and previous-frame grayscale images.

3. Moving Vehicle Detection Based on Shadow Removal

The changes outside of the vehicle in the complex transportation environment can affect the detection of moving vehicles, for example, the wind blows the leaves and causes changes in the background. Although the optical flow method is suitable for the detection of all moving objects on an image sequence, it is sensitive to noise and light sources, such as shadows around the vehicle, which adversely affect the accurate detection of moving vehicles. Therefore, how to effectively remove the shadow is an important factor for optical flow detection. A large number of experiments have shown that shadows can be well detected in the HSV color space [42]. Therefore, this paper combines the shadow removal with the optical flow method to eliminate the interference of shadows and make the moving vehicle detection more accurate. The detailed steps are as follows:

- (i) Read the video of moving vehicle detection, determine the shadow areas after converting each

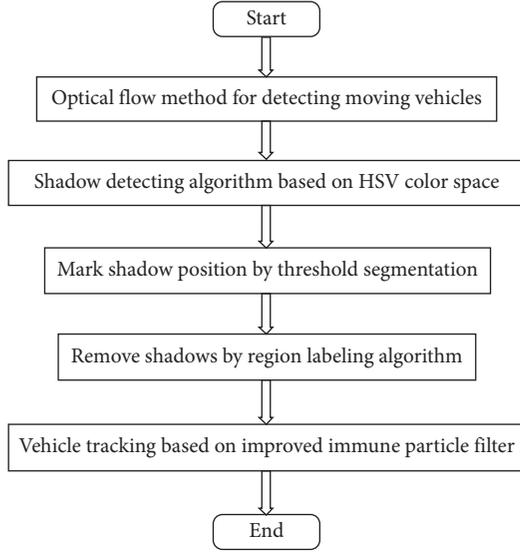


FIGURE 1: Algorithm flowchart.

frame image of the video two times in the HSV color space, and then convert the obtained image into a grayscale image.

- (ii) Calculate the optical flow field vector and add the optical flow field vector to the video frame.
- (iii) Calculate the average amplitude value of the optical flow vector to obtain the speed threshold; then, extract the moving object according to the speed threshold; finally, remove the noise by the median filter.
- (iv) Remove the shadow areas on the moving vehicle according to the shadow areas detected by the shadow detection algorithm.
- (v) Remove the road area by morphological erosion algorithm and then fill the “cavity” areas of the vehicle by morphological close operation. Repeat the next frame until all frames are completed.

The flowchart of the improved algorithm is shown in Figure 3.

3.1. Shadow Detection Based on HSV Color Space. In order to obtain accurate moving vehicle detection, shadows need to be removed. Commonly, there are two kinds of shadow detection methods, one is based on shadow feature and the

other is based on the geometric model. This paper uses the color features of shadows in the HSV color space to detect shadows.

The pipeline of shadow detection in the HSV color space is as follows: first, input a frame of image and convert the RGB image into a gray image by color space transformation; second, use the Otsu threshold detection method to obtain the threshold of image and binarize the image; third, remove noise by filter; fourth, the shadow area is detected. The flowchart is shown in Figure 4.

3.1.1. HSV Color Model. The HSV color model is similar to an inverted hexagonal pyramid model [43], where V, H, and S represent the brightness, color, and saturation, respectively. The transformation from RGB to HSV is shown in equations (5)–(7):

$$H = \begin{cases} \frac{60(G - B)}{V - \min(R, G, B)}, & V = R, \\ \frac{120 + 60(B - R)}{V - \min(R, G, B)}, & V = G, \\ \frac{240 + 60(R - G)}{V - \min(R, G, B)}, & V = B, \end{cases} \quad (5)$$

$$S = \begin{cases} \frac{V - \min(R, G, B)}{V}, & V \neq 0, \\ 0, & \text{others,} \end{cases} \quad (6)$$

$$V = \max(R, G, B). \quad (7)$$

3.1.2. Shadow Detection. The HSV color space is closer to human vision and can accurately reflect the information of the target. The shadow of the moving vehicle is detected by the nature of the various parameters on the background by the shadow in the HSV color space. In the detection, the V component will become smaller and change a lot relative to the background area, and it will be an important parameter for discriminating the shadow. For the S component, the shadow has a lower value and the difference from the background is negative. The H component usually does not change [39]. Based on this feature, the shadow can be detected. The specific algorithm is as follows:

$$SP(i, j) = \begin{cases} \text{shadow points,} & \text{if } \alpha \leq \frac{T_{i,j}^v}{B_{i,j}^v} \leq \beta, (I_{i,j}^S - B_{i,j}^S) \leq T^S, \\ & \text{and } \text{abs}(I_{i,j}^H - B_{i,j}^H) \leq T^H, \\ \text{nonshadow points,} & \text{otherwise,} \end{cases} \quad (8)$$

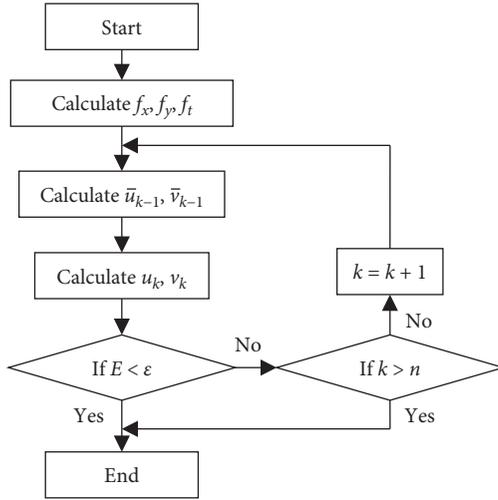


FIGURE 2: Flowchart of the HS algorithm.

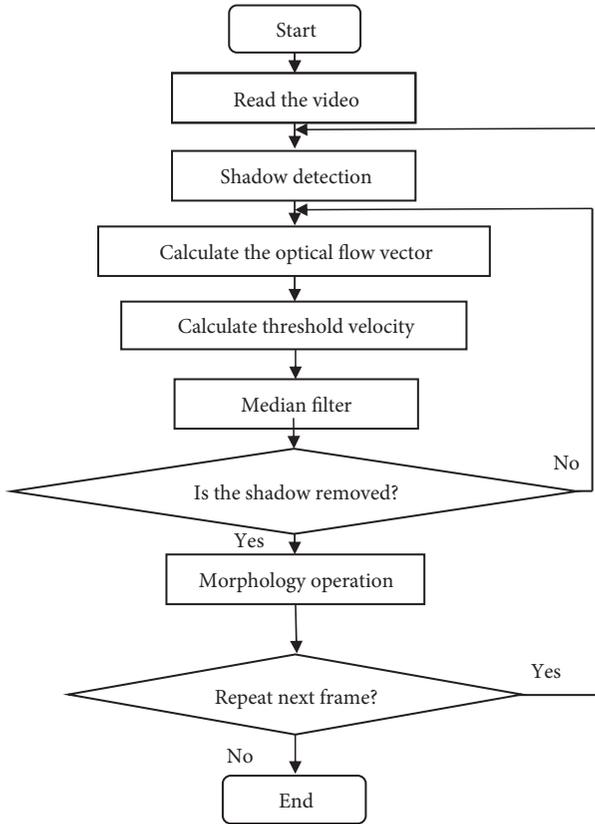


FIGURE 3: Flowchart of the improved algorithm.

where α and β represent the threshold values of luminance; T^S and T^H represent the threshold values of saturation and color, respectively; and $0 < \alpha < 1, 0 < \beta \leq 1$. T^S, T^H, α , and β are determined by many experiments, and in the paper their values are 15, 23, 0.15, and 0.47, respectively.

3.1.3. Automatic Threshold Determination Based on Otsu. The Otsu algorithm [43] is used in the HSV color space for shadow detection [44], which is a self-adaptive threshold

determination method. The larger the variance between the foreground and the background is, the greater the difference between them is. Assuming t is the set threshold, the gray level of the image is L ; then traverse t from 0 to L ; if $t = t^*$, the variance between foreground and background becomes the largest, and then t^* is regarded as the required threshold, where the variance ξ is calculated as follows:

$$\xi = \sqrt{A_0 \times (B_0 - B)^2 + A_1 \times (B_1 - B)^2}, \quad (9)$$

where $B = A_0 \times B_0 + A_1 \times B_1$, A_0 is the proportion of the moving target pixels to the total image pixels, B_0 is the average grayscale value of the moving target pixels, A_1 is the proportion of the background pixels to the total image pixels, B_1 is the average grayscale value of the background pixels, and B is the total average grayscale value of the total image.

3.2. Shadow Removal. Remove the shadow of the moving vehicle based on the detected shadows area and obtain accurate vehicle detection. Then, use morphology operation [45] to remove the road area. Finally, obtain complete and accurate detection area of the moving vehicle by filling the ‘‘cavity’’ via close operation [46].

4. Improved Immune Particle Filter for Vehicle Tracking

This study tracks vehicle targets based on the particle filter framework under the complex transportation environments. To reduce the occurrence of particle degradation and maintain the diversity of particles as much as possible, inspired by the literature [32], a vehicle tracking framework based on immune particle filter is proposed, which improves the reliability and robustness of vehicle tracking by improving the particle resampling process.

During vehicle tracking, the goal is a rigid object. Therefore, the algorithm uses a rectangular box with a constant ratio of width and height to indicate the target state, i.e., $x_t = (x_c, y_c, w, h, s)^T$, where x_c, y_c, w, h , and s represent the center point abscissa, center point ordinate, rectangle width, rectangle height, and scale factor of the rectangular box, respectively.

4.1. Particle Filter. Particle filtering is a computational method for discretely approximating Bayesian posterior probability density [33]. Particle filtering refers to estimating the current state of a target by previously determined states. The current posterior probability of the target is calculated based on the detected data and the posterior probability of the previous moment. Particle filter tracking is to first spread the particles by resampling the distribution of particles. Then, redetect the target, and finally update the state by normalization [33]. The advantage of the particle filter tracking is that it runs fast and can solve the problem of partial occlusion. Therefore, we use the particle filter to track the vehicle in the paper.

According to Monte Carlo algorithm [47], the mathematical expectation $E(g(a_{0:t})) = \int g(a_{0:t})p(q_{0:t}|b_{1:t})da_{0:t}$ can be obtained, and the expectation value is used to express the estimated weight of the particle. In addition, during the calculation process, the weights of particles are generally obtained from an approximate posterior probability density. According to the Bayes formula, we get the probability density $p(q_{0:t}|b_{1:t}) = (p(b_{1:t}|a_{0:t})p(a_{0:t})/p(b_{1:t}))$; let $w(a_{0:t}) = (p(b_{1:t}|a_{0:t})p(a_{0:t})/q(a_{0:t}|b_{1:t}))$; then, the particle sample $E(g(a_{0:t})) = (\int g(a_{0:t})w_t(a_{0:t})q(a_{0:t}|b_{1:t})da_{0:t})/\int w_t(a_{0:t})q(a_{0:t}|b_{1:t})da_{0:t}$ can be obtained.

4.2. Color Distribution Model. The observation model is based on the color feature during particle filtering process. The dimension of the HSV color histogram for statistical calculation is set as $L = n_H \cdot n_S + n_V$ [48], where n_H , n_S , and n_V represent the dimensions of the H component, S component, and V component, respectively. Compared to methods based on other color spaces, the method based on HSV color histogram can reduce the impact of lighting change on tracking accuracy to some extent. On the other hand, due to the frequently occurring occlusion in the vehicle tracking process, the HSV color histogram is calculated by the blockwise statistic way in the algorithm, which can effectively reduce the probability of target loss in this case with the help of the likelihood calculation method. The color histogram based on blockwise way can be defined as

$$H(k_{ij}) = \frac{m_{k_{ij}}}{N_{\text{block}}}, \quad k_{ij} = i \times J + j + k', \quad (10)$$

where $k' = 0, 1, \dots, L-1$ represents the dimension of HSV histogram of every block; $i = 0, 1, \dots, I-1$ and $j = 0, 1, \dots, J-1$, which represent the target rectangular box are equally divided into $I \times J$ identical blocks; $m_{k_{ij}}$ represents the number of pixels in the block of i th row and j th column that fall within the color interval corresponding to k_{ij} ; and N_{block} represents the total number of pixels in every block. Therefore, the proposed histogram based on blockwise way can be regarded as a series of $I \times J$ normalized subhistograms.

4.3. Artificial Immune Algorithm. Although the classical immune resampling algorithm optimizes overall particle filtering algorithm frame by making particles maintain diversity to a certain extent, it does not consider scene changes and interference of similar objects in practical applications. The algorithm uses the anti-interference affinity calculation and the improved mutation function of antibody to make the particle filter algorithm have certain adaptivity and robustness to scene interference in the vehicle tracking process.

4.3.1. Affinity Calculation. Affinity is used to measure the probability that the region determined by the particle at time t coincides with the target, defined as

$$p(z_t | x_t^i) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left\{-\frac{1}{2\sigma^2}D_i^2\right\}, \quad (11)$$

where σ represents variance of Gaussian distribution and D_i represents the distance function. To reduce the tracking error caused by partial occlusion, the study further makes an improvement for the distance function D_i . D_i is updated as follows:

$$D_i = \begin{cases} 1, & N_O > N_{\text{th}}, \\ \bar{d}, & N_O \leq N_{\text{th}}, \end{cases} \quad (12)$$

where N_O represents the number of the blocks where objects are occluded; N_{th} represents the given threshold (typically, let $N_{\text{th}} = I \times J/2$); and \bar{d} is the mean value of the distance values between the occluded blocks and the corresponding target templates, $\bar{d} = (1/N_O)\sum_{i=1}^{N_O}d_i$, in which d_i is the Bhattacharyya distance between the histogram $H(b)$ of every occluded block and the corresponding target template $T(b)$ [49], $d_i = \sqrt{1 - \sum_{b=0}^{L-1} \sqrt{H(b) \cdot T(b)}}$.

4.3.2. Antibody Mutation Function Improvement. During vehicle tracking, there is no need to perform antibody mutation operation for high-weight particles, and antibody mutation operation is performed only for the low-weight particles according to $U(-u, u)$ distribution. Here, we adopt a new antibody mutation function $F'(t)$, which is defined as follows:

$$F'(t) = \begin{cases} \left(\gamma + (1-\gamma) \frac{f_t^* - f_t^i}{f_t^* - \min(f_t^i)} \right) \times \mu \times x_t^i, & \omega_t^i \leq \omega_{\text{th}}, \\ 0, & \omega_t^i > \omega_{\text{th}}, \end{cases} \quad (13)$$

where f_t^i represents the affinity function, f_t^* represents the critical affinity of meeting the weight threshold condition, γ represents the dynamic adjustment factor, $\mu \sim U(-1, 1)$, x_t^i represents the particle state, ω_t^i indicates particle weight, and ω_{th} is a set particle weight threshold. Note that $\omega_{\text{th}} = 0.8$ in the paper, which is determined by many experiments.

4.4. Algorithm Process

- (i) Initialization: calculate the color histogram of object of template by equation (10). Taking the center of the target as the origin point, select N particles by random sampling based on Gaussian distribution as antibodies and assign the affinity value for each antibody to be $1/N$.
- (ii) Selection: N samples are randomly selected from the N input samples according to the weights of samples. The selected probability of the particles with high weights is high, and the probability of the particles with small weight is small.
- (iii) Cloning: the affinity $p(z_t | x_t^i)$ of each antibody is calculated from equation (11). The aim of cloning is to promote antibodies with high affinity and to

inhibit antibodies with low affinity. Therefore, the proposed algorithm can quickly converge to the global optimal solution by cloning operation.

- (iv) Variation: mutation calculation is performed for all antibodies according to equation (14), where $F'(t)$ is calculated by equation (13).

$$x_t^i = x_t^i + (1 - \tilde{f}_t^i) F'(t). \quad (14)$$

- (v) Maturity: calculate the affinity of all antibodies after mutation operation. After ranking according to the degree of affinity, reselect the top N antibodies as a new antibody memory set for further optimization.
- (vi) Estimation: calculate the sum of weights of the N antibodies in the memory set to obtain a new updated target value. Next, judge whether the program ending condition is satisfied, and if it is satisfied, then it ends; otherwise, the program turns to step (ii).

5. Experiment

Experiments are implemented on the computer with Intel(R) Core(TM) i5-2410M CPU 2.30 GHz and 4.00 G memory. The image size is 320×240 pixels, and the software is programmed by Matlab 2013. The experimental results are as follows.

5.1. Shadow Detection Results. The video of moving vehicles is captured by a camera fixed on an overpass. The video is in AVI format with a frame rate of 15 frames/s. The number of moving vehicles is sufficient for the experiment. The experimental results of shadow detection are shown in Figure 5, where the left column is the original images and the right column is the detection results of the shadow area.

On the right column, the white area indicates the shadow area detected by the shadow detection algorithm based on HSV color space. In order to obtain complete shadow area, the morphological close operation [46] is performed after the image binary based on the threshold segmentation.

5.2. Comparison of the Proposed Method and Traditional Optical Flow Method. In order to verify the effect of the proposed method, we compare the proposed method with the traditional optical flow method. Figure 6 shows the detection results based on traditional optical flow method. Figure 6(a) shows the original image, Figure 6(b) shows the grayscale image after gray processing, Figure 6(c) shows the optical flow vector after optical flow calculation, Figure 6(d) gives the image through morphological filtering, and Figure 6(e) shows the detection result of moving vehicle.

Figures 7 and 8 show the detection results based on the proposed method, where Figure 7 shows the results of frame #40 and Figure 8 shows the results of frame #100. Figures 7(a) and 8(a) show the original images, Figures 7(b) and 8(b) show the grayscale images after gray processing,

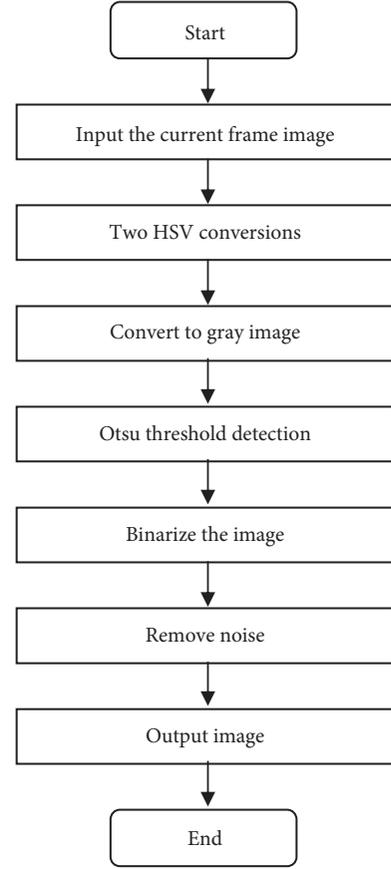


FIGURE 4: Flowchart of shadow area detection.

Figures 7(c) and 8(c) show the optical flow vectors after optical flow calculation, Figures 7(d) and 8(d) show the image through morphological filtering, and Figures 7(e) and 8(e) show the detection results of moving vehicle.

As shown in Figure 6, the shadow area is included into the detection box of moving vehicle detection based on the traditional optical flow method. However, from Figures 7 and 8, the shadow area is removed from the detection box of moving vehicle detection based on the proposed method. Therefore, the proposed method can more accurately detect moving vehicle than the traditional optical flow method.

We conduct an extensive experiment on a video with a total of 30 vehicles based on the proposed method and traditional optical flow method. The proposed method with shadow removal can accurately detect the moving vehicles and give correct number of the moving vehicles. However, the traditional method cannot correctly detect the number of the moving vehicles in the video. The reason is that a black car in the left lane of the frame #70 is not correctly detected, as shown in Figure 9, where the car is removed as a shadow because illuminance is poor, and the color of the car itself is also black. Therefore, the proposed method has strong robustness to shadow and similar-shadow objects compared with the traditional optical flow method.

Table 1 shows the performance comparison of the proposed method, background subtraction method, inter-frame difference method, and traditional optical flow

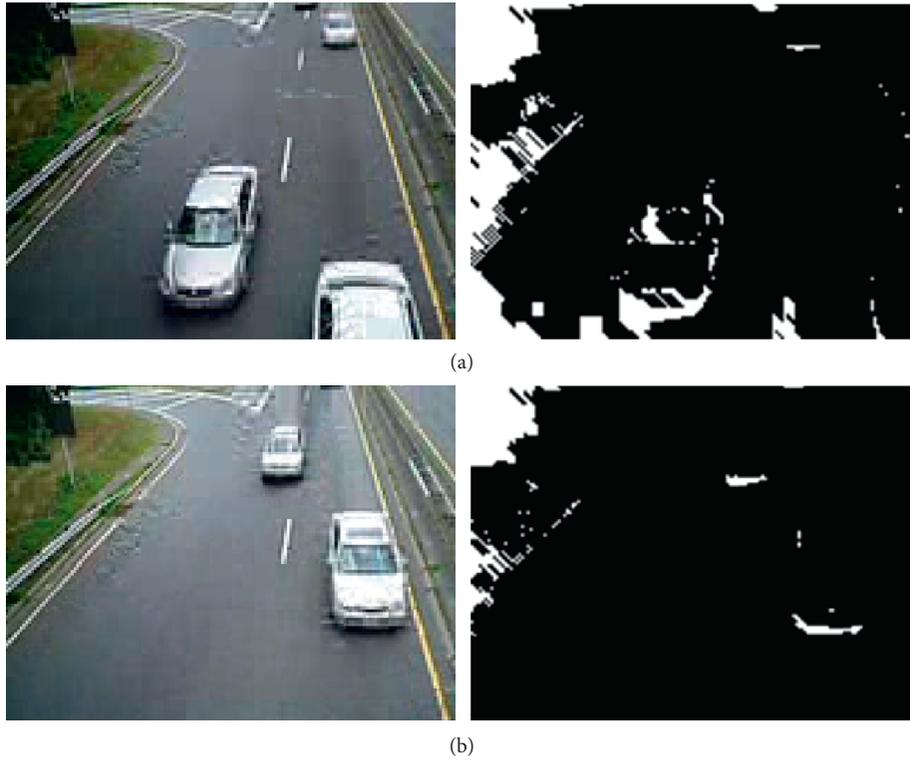


FIGURE 5: Shadow detection results based on HSV color space. (a) Frame #12. (b) Frame #23.

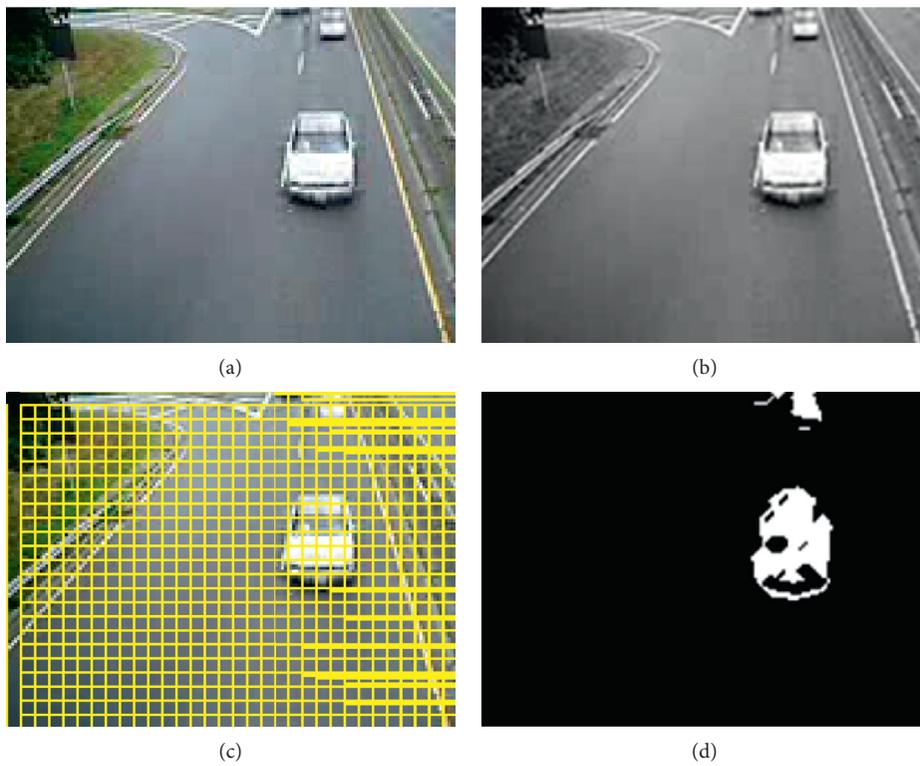
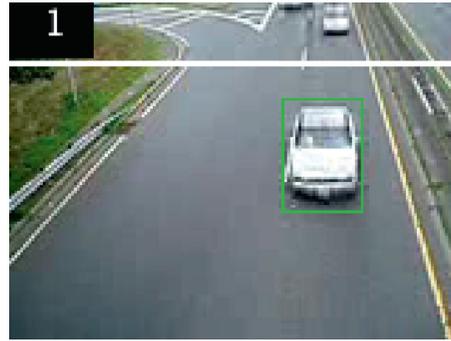
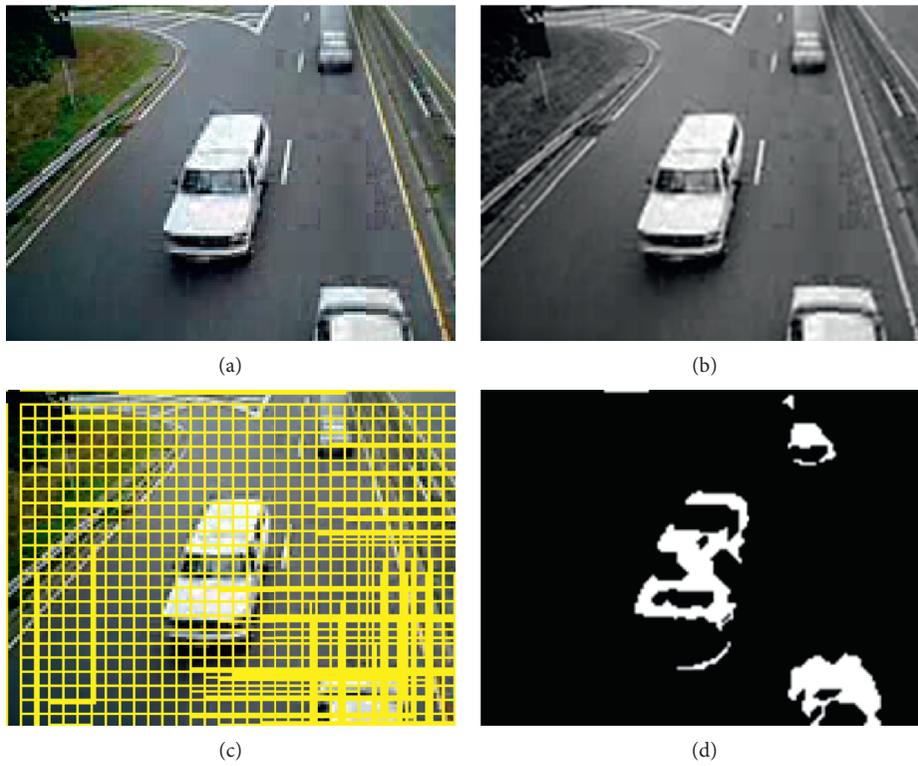


FIGURE 6: Continued.



(e)

FIGURE 6: Detection results of the traditional optical flow method. (a) The original image. (b) Grayscale image. (c) Optical flow vector. (d) Morphological filtering. (e) Test results.



(c)

(d)



(e)

FIGURE 7: Detection results of the 40th frame. (a) The original image. (b) Grayscale image. (c) Optical flow vector. (d) Morphological filtering. (e) Test results.

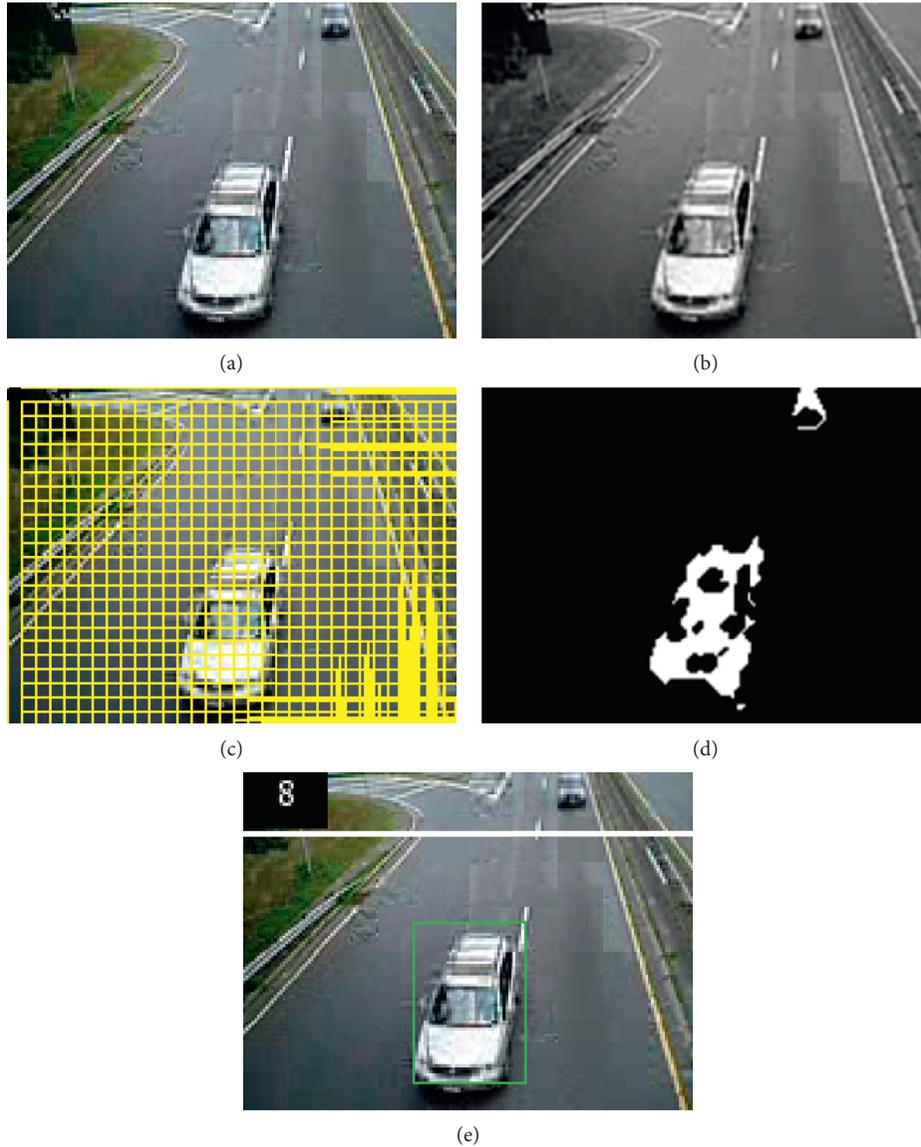


FIGURE 8: Detection result of the 100th frame. (a) The original image. (b) Grayscale image. (c) Optical flow vector. (d) Morphological filtering. (e) Test results.

method in terms of average computational time and accuracy. As shown in Table 1, although the background subtraction method, interframe difference method, and traditional optical flow method have fast detection speed, they have lower accuracy than the proposed method.

5.3. Tracking Results. Based on the proposed tracking algorithm, we perform the vehicle tracking experiments. Experimental results are shown in Figures 10–12. Figures 10–12 show the tracking results in the 2nd, 10th, and 20th frames in the captured video, respectively. The red points in Figures 10–12 indicate the detection result based on optical flow method, and the blue points in Figures 11 and 12 indicate the trajectory of the tracked moving vehicle.

The videos of the tracking experiments include the normal traveling condition of vehicles on the highway and abnormal conditions such as illumination change, similar neighboring objects, occlusion, and scale changes. To verify the effectiveness of the proposed tracking algorithm, experiments were performed under the same conditions, and the tracking results were compared with two typical target tracking algorithms, i.e., Kalman filter algorithm and Camshift algorithm.

In terms of parameter setting, the particle number of the particle filter is set as 100, and other corresponding parameters, such as system noise and observation noise, are selected as consistent as possible in the three algorithms. Since the tracking results of the test videos have been manually labeled, we refer to them as the ground

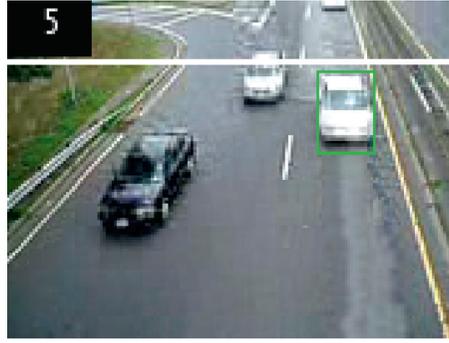


FIGURE 9: Detection result of frame #70.

TABLE 1: Performance comparison of the proposed method and other methods.

Methods	Average computational time (millisecond)	Accuracy (%)
Background subtraction method	56	87
Interframe difference method	49	89
Traditional optical flow method	103	92
The proposed method	119	95

FIGURE 10: Tracking result of the 2nd frame. (a) The original image. (b) Tracked vehicle image.FIGURE 11: Tracking result of the 10th frame. (a) The original image. (b) Tracked vehicle image.

truth and define two evaluation criteria to compare the tracking results, namely, the centroid error and the target domain coverage accuracy. The centroid error and the

target domain coverage accuracy of single-frame image are expressed by E_{rr} and A_{cc} , as shown in equations (15) and (16), respectively:



FIGURE 12: Tracking result of the 20th frame. (a) The original image. (b) Tracked vehicle image.

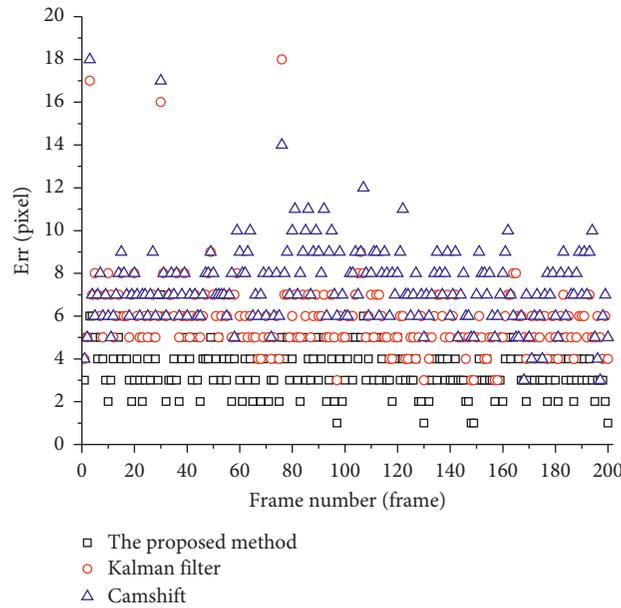


FIGURE 13: Comparison of centroid error for three algorithms in complex traffic scenarios.

$$E_{rr} = |x - x_G| + |y - y_G|, \quad (15)$$

$$A_{cc} = \frac{(R \cap R_G)}{(R + R_G - R \cap R_G)}, \quad (16)$$

where x and y indicate the horizontal and vertical coordinates of the center of tracked target vehicle, respectively; x_G and y_G indicate the horizontal and vertical coordinates of the ground truth; and $R \cap R_G$ expresses the overlap area between the area R determined by the proposed algorithm and the area R_G determined by the ground truth.

Based on the two evaluation criteria, we select the tracking results of 200 frames of continuous vehicle images to evaluate the performance of the proposed method. First, the tracking experiments are performed on a sunny day with good visibility. The average value of centroid errors of the proposed method is 2.1, and the average value of the target domain coverage accuracy is 93.6%. However, the average values of centroid errors of the Camshift algorithm and

Kalman filter algorithm are 4.2 and 3.8, respectively, and the average values of the target domain coverage accuracy of the two algorithms are 69.1%, and 71.3%, respectively. Experimental results show that in simple scenarios, the proposed algorithm in this paper can well adapt to vehicle changes and has higher tracking accuracy than other two algorithms.

In addition to the experiments in good traffic scenarios, we also perform vehicle tracking experiments under complex transportation environments with poor visibility and target-like interference around target vehicle on a rainy day. The average value of centroid errors of the proposed method is 3.5, and the average value of the target domain coverage accuracy is 75.3%. However, the average values of centroid errors of the Camshift algorithm and Kalman filter algorithm are 7.4 and 5.9, respectively, and the average values of the target domain coverage accuracy of the two algorithms are 60.8%, and 67.1%, respectively. Experimental results are shown in Figures 13 and 14. At the 30th frame image, an obvious turn of the tracked vehicle leads to the failure of tracking for the Camshift algorithm. Furthermore, at the

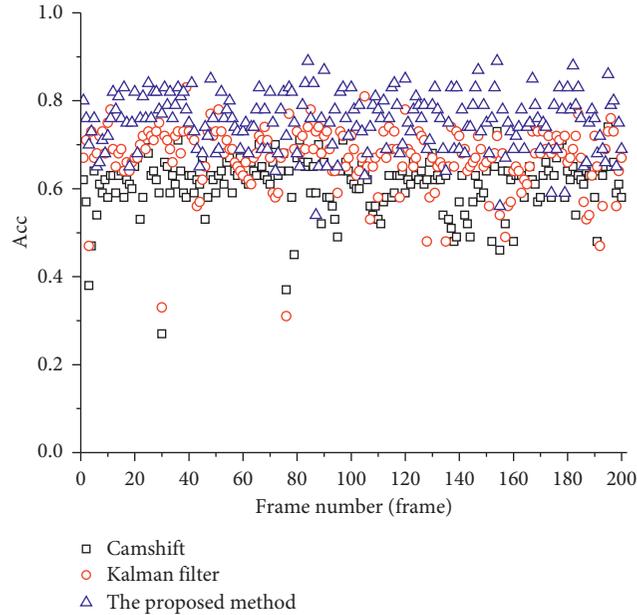


FIGURE 14: Comparison of target domain coverage accuracy for three algorithms in complex traffic scenarios.

76th frame image, the Kalman filter algorithm fails to accurately track the object vehicle due to a sudden acceleration of the tracked vehicle along with a poor illumination. Besides, at the 3rd frame of the video, because the object vehicle is small and a black sedan similar to the object vehicle in color just appears on its right side, the Camshift algorithm fails to track the object vehicle. However, thanks to the use of the anti-interference affinity calculation and the improved mutation function of antibody, the proposed method can make the particle filter algorithm have certain adaptivity and robustness to scene interference. Therefore, despite poor illumination, sudden acceleration of the tracked, and similar-color interference, the proposed method can still favorably track the object vehicle.

6. Conclusion

This study proposed a new moving vehicle detection and tracking method based on optical flow and immune particle filter under complex transportation environment. The proposed method firstly uses the optical flow method to roughly detect the moving vehicle and then uses the shadow detection algorithm based on the HSV color space to accurately detect the moving vehicle and finally robustly tracks the moving vehicle based on the proposed immune particle filter algorithm. The experiments under complex traffic scenes with shadow interference demonstrate that the proposed method can well solve the impact of shadow interference on moving vehicle detection and realize accurate detection as well as robust tracking of a moving vehicle. Due to adopting shadow removing and improved immune particle algorithms, the proposed method achieves higher accuracy of vehicle detection and tracking compared with existing algorithms of Camshift and Kalman filter. However, the proposed method in the paper is limited to daytime with

good and poor illumination; for the conditions at night, following research will be further done by considering to employ infrared image of moving vehicles.

In the future, the experimental results can be transferred to a cloud computing platform through a wireless sensor network and can be further analyzed and processed by policymakers to enhance the vehicle management level [25, 27, 51].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported in part by the National Natural Science Foundation of China (nos. 61304205 and 61502240), Natural Science Foundation of Jiangsu Province (BK20191401), and Innovation and Entrepreneurship Training Project of College Students (201910300050Z and 201910300222).

References

- [1] Y. Guo, J. Wang, S. Peeta, and P. Ch. Anastasopoulos, "Personal and societal impacts of motorcycle ban policy on motorcyclists' home-to-work morning commute in China," *Travel Behaviour and Society*, vol. 19, pp. 137–150, 2020.
- [2] Y. Guo and S. Peeta, "Impacts of personalized accessibility information on residential location choice and travel behavior," *Travel Behaviour and Society*, vol. 19, pp. 99–111, 2020.

- [3] J. Zhou, J. Sun, P. Cong et al., "Security-critical energy-aware task scheduling for heterogeneous real-time MPSoCs in IoT," *IEEE Transactions on Services Computing (TSC)*, 2019, In press.
- [4] J. Zhou, J. Sun, X. Zhou et al., "Resource management for improving soft-error and lifetime reliability of real-time MPSoCs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 12, pp. 2215–2228, 2019.
- [5] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, and W. Dou, "Be-Come: blockchain-enabled computation offloading for IoT in mobile edge computing," *IEEE Transactions on Industrial Informatics*, 2019.
- [6] X. Zhao, C. Su, and H. Chen, "Research on target detection technology based on video image," *Electronic Design Engineering*, vol. 21, no. 16, pp. 114–116, 2013.
- [7] Y. Ma, X. Song, X. Li, and J. Liu, "Research and implementation of real-time monitoring algorithm based on embedded traffic flow," *LCD & Display*, vol. 33, no. 9, pp. 787–792, 2018.
- [8] T. Gao, Z.-g. Liu, S.-h. Yue, J. Zhang, J.-q. Mei, and W.-c. Gao, "Robust background subtraction in traffic video sequence," *Journal of Central South University of Technology*, vol. 17, no. 1, pp. 187–195, 2010.
- [9] J. Xu, M. Fang, and H. Yang, *Motion Detection and Tracking in Computer Vision*, National Defense Industry Press, Beijing, China, 2012.
- [10] D. Forsyth, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2014.
- [11] J. Guerrero, S. Zeadally, and J. Contreras-Castillo, "Sensor technologies for intelligent transportation systems," *Sensors*, vol. 18, no. 4, p. 1212, 2018.
- [12] S. Zhu, Y. Guo, J. Chen, D. Li, and L. Cheng, "Integrating optimal heterogeneous sensor deployment and operation strategies for dynamic origin-destination demand estimation," *Sensors*, vol. 17, no. 8, p. 1767, 2017.
- [13] S. Zhu, H. Zheng, S. Peeta, Y. Guo, L. Cheng, and W. Sun, "Optimal heterogeneous sensor deployment strategy for dynamic origin-destination demand estimation," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2567, no. 1, pp. 18–27, 2016.
- [14] P. Ye and D. Wen, "Optimal traffic sensor location for origin-destination estimation using a compressed sensing framework," *Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1857–1866, 2017.
- [15] S. Oh, S. Russell, and S. Sastry, "Markov chain Monte Carlo data association for multi-target tracking," *IEEE Transactions on Automatic Control*, vol. 54, no. 3, pp. 481–497, 2009.
- [16] G. Yuan, Z. Chen, and J. Gong, "A moving target detection algorithm combining optical flow method and three-frame difference method," *Small Microcomputer System*, vol. 34, no. 3, pp. 668–671, 2013.
- [17] N. Ali and G. Hassan, "Kalman filter tracking," *International Journal of Computer Applications*, vol. 89, no. 9, pp. 15–18, 2014.
- [18] Y. Zhang, C. Yin, Q. Wu et al., "Location-aware deep collaborative filtering for service recommendation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019.
- [19] J. Sun, "A fast MEANSHIFT algorithm-based target tracking system," *Sensors*, vol. 12, no. 6, pp. 8218–8235, 2012.
- [20] Y. Zhang, K. Wang, Q. He et al., "Covering-based web service quality prediction via neighborhood-aware matrix factorization," *IEEE Transactions on Services Computing*, 2019.
- [21] JH. Lee, HJ. Jung, and J. Yoo, "A real-time face tracking algorithm using improved CamShift with depth information," *Journal of Electrical Engineering & Technology*, vol. 12, no. 5, pp. 2067–2078, 2017.
- [22] Y. Zhang, G. Cui, S. Deng et al., "Efficient query of quality correlation for service composition," *IEEE Transactions on Services Computing*, 2018.
- [23] XF. Kong, Q. Chen, GH. Gu, K. Ren, WX. Qian, and ZW. Liu, "Particle filter-based vehicle tracking via HOG features after image stabilisation in intelligent drive system," *IET Intelligent Transport Systems*, vol. 13, no. 6, pp. 942–949, 2018.
- [24] X. Xu, R. Mo, F. Dai, W. Lin, S. Wan, and W. Dou, "Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloud," *IEEE Transactions on Industrial Informatics*, 2019.
- [25] X. Xu, Q. Cai, G. Zhang et al., "An incentive mechanism for crowdsourcing markets with social welfare maximization in cloud-edge computing," in *Concurrency and Computation: Practice and Experience*, 2018.
- [26] X. Xu, Y. Li, T. Huang et al., "An energy-aware computation offloading method for smart edge computing in wireless metropolitan area networks," *Journal of Network and Computer Applications*, vol. 133, pp. 75–85, 2019.
- [27] X. Xu, X. Zhang, M. Khan, W. Dou, S. Xue, and S. Yu, "A balanced virtual machine scheduling method for energy-performance trade-offs in cyber-physical cloud systems," *Future Generation Computer Systems*, 2017.
- [28] L. Qi, X. Zhang, W. Dou, and Q. Ni, "A distributed locality-sensitive hashing-based approach for cloud service recommendation from multi-source data," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2616–2624, 2017.
- [29] L. Qi, X. Zhang, W. Dou, C. Hu, C. Yang, and J. Chen, "A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment," *Future Generation Computer Systems*, vol. 88, pp. 636–643, 2018.
- [30] L. Qi, X. Zhang, S. Li, S. Wan, Y. Wen, and W. Gong, "Spatial-temporal data-driven service recommendation with privacy-preservation," *Information Sciences*, vol. 515, pp. 91–102, 2020.
- [31] J. Zhou, X. S. Hu, Y. Ma, J. Sun, T. Wei, and S. Hu, "Improving availability of multicore real-time systems suffering both permanent and transient faults," *IEEE Transactions on Computers*, vol. 68, no. 12, pp. 1785–1801, 2019.
- [32] H. Han, Y. Ding, and K. Hao, "A new immune particle filter algorithm for tracking a moving target," in *Proceedings of the 2010 Sixth International Conference on Natural Computation*, pp. 3248–3252, Yantai, China, August 2010.
- [33] W. Liu, X. Dai, and Z. Zhu, "Maneuvering target tracking method based on importance resampling particle filter," *Journal of Jiangsu University of Science and Technology, Natural Science Edition*, vol. 21, no. 1, pp. 37–41, 2007.
- [34] R. M. Yang, Q. Liu, M. Li, J. Wang, and Y. Li, "A survey of matrix completion methods for recommendation systems," *Big Data Mining and Analytics*, vol. 1, no. 4, pp. 308–323, 2018.
- [35] H. Liu, H. Kou, C. Yan, and L. Qi, "Link prediction in paper citation network to construct paper correlation graph," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, 2019.
- [36] S. Kumar and M. Singh, "Big data analytics for healthcare industry: impact, applications, and tools," *Big Data Mining and Analytics*, vol. 2, no. 1, pp. 48–57, 2019.

- [37] X. Xu, Q. Liu, Y. Luo et al., "A computation offloading method over big data for IoT-enabled cloud-edge computing," *Future Generation Computer Systems*, vol. 95, pp. 522–533, 2019.
- [38] S. Zhang, W. Zhang, and H. Ding, "Target detection method for fused optical velocity and background modeling," *Journal of Image and Graphics*, vol. 16, no. 2, pp. 236–243, 2011.
- [39] W. Gong, L. Qi, and Y. Xu, "Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 3075849, 8 pages, 2018.
- [40] L. Xia, L. Chen, J. Fu, and J. Wu, "Symmetries and variation calculation of discrete Hamiltonian systems," *Chinese Physics B*, vol. 23, no. 7, 2014.
- [41] J. Lin, S. Yan, and Y. Liu, "Application of meshless nine-point difference method in solving marine pollution," *Journal of Dalian Maritime University*, vol. 30, no. 1, pp. 78–80, 2004.
- [42] W. Gao, H. Dong, and L. Lan, "Research on moving target shadow detection algorithm in adaptive background," *Modern Electronic Technology*, vol. 31, no. 6, pp. 59–61, 2008.
- [43] Y. Jiang, "Comparative study of three commonly used color models," *Journal of Hunan University of Science and Technology*, vol. 28, no. 4, pp. 37–38, 2007.
- [44] X. Gao, J. Li, and C. Tian, *Modern Image Analysis*, Xi'an University of Electronic Science and Technology Press, Xi'an, China, 2011.
- [45] Z. Yin, K. Mo, and Y. Xiong, "Defective point data reconstruction based on improved process of morphological operations," *Science China Technological Sciences*, vol. 54, no. 12, pp. 3166–3179, 2011.
- [46] R. Priyadharsini, A. Beulah, and T. S. Sharmila, "Optic disc and cup segmentation in fundus retinal images using feature detection and morphological techniques," *Current Science*, vol. 115, no. 4, pp. 748–752, 2008.
- [47] J. Li and J. Wang, *Image Detection and Target Tracking Technology*, Beijing Institute of Technology Press, Beijing, China, 2014.
- [48] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proceedings of the Computer Vision—ECCV 2002*, pp. 661–675, Copenhagen, Denmark, May 2002.
- [49] X. Wang, N. Chen, and A. Shi, "An improved similarity measure in particle filters for robust object tracking," in *Proceedings of the International Congress on Image and Signal Processing*, pp. 46–50, Hangzhou, China, December 2013.
- [50] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.
- [51] Y. Guo, J. Wang, S. Peeta, and P. C. Anastasopoulos, "Impacts of internal migration, household registration system, and family planning policy on travel mode choice in China," *Travel Behaviour and Society*, vol. 13, pp. 128–143, 2018.