Sensors and Materials, Vol. 32, No. 12 (2020) 4517–4536
MYU Tokyo

S & M 2426

Method of Unsupervised Static Recognition and Dynamic Tracking for Vehicles

Yifei Cao, Jingguo Lv,* Yingqi Bai, and Anqi Wu

Beijing University of Civil Engineering and Architecture, Beijing 102616, China

(Received September 29, 2020; accepted December 2, 2020)

Keywords: vehicle tracking, mean-shift algorithm, Gaussian mixture model, edge detection, shadow elimination

Vehicle object tracking is a research hotspot in computer vision. To solve the problem of single object extraction caused by the shadow effect and occlusion between vehicles, this paper presents a vehicle object tracking algorithm suitable for both dynamic and stationary states. First, the improved Canny algorithm is used to obtain the information in a video sequence, and the dynamic region of the object is extracted using the difference between the mean of the video sequence and the object frame. Secondly, the Gaussian mixture model is used for video object segmentation to obtain the foreground image and the background image, and the static object is identified through the intersection operation of the object dynamic region and the foreground image combined with the edge information. Then, chroma information is introduced into a statistical nonparametric model to eliminate the shadow of the foreground image, and the mean-shift tracking algorithm is used for dynamic object tracking of the foreground image after eliminating the shadow. The experimental results show that the proposed tracking algorithm can identify and track vehicles effectively and quickly, providing new ideas for the future development of the sensor field.

1. Introduction

In recent years, with the continuous development of sensors, we can obtain clear traffic video information through cameras, enabling us to achieve vehicle object tracking and detection. With the continuous development of computer vision technology, vehicle object recognition and tracking have become research hotspots in the fields of computer vision and image processing. Vehicle object tracking is widely used in many fields, such as video intelligent monitoring, traffic monitoring, and automatic driving.

The unique sequence characteristics of video allow video object segmentation to play a decisive role in vehicle object tracking. When performing vehicle tracking in a complex environment, factors such as the variation of light, noise interference, and object occlusion will affect video object segmentation. Therefore, to provide a real-time and accurate foreground image for the tracking model, it is necessary to solve the problem of shadow interference caused

*Corresponding author: e-mail: lvjingguo@bucea.edu.cn
https://doi.org/10.18494/SAM.2020.3129

ISSN 0914-4935

by illumination changes and the difficulty in single object extraction caused by occlusion between objects. It is difficult for the traditional object-tracking methods such as region segmentation, optical flow, and interframe difference to solve the problem. Although the currently popular deep learning methods have strong advantages in the vehicle object tracking field, they need a large number of samples to train a network model in practical applications.

Video object segmentation is a basic task in vehicle object tracking, which is to assign the same label value to pixels belonging to the same object in consecutive video frames by analyzing the video sequence. According to the type of information used, video object segmentation can be divided into two categories: one is based on spatial information[1] and the other is based on temporal information.[2] However, in many cases, video foreground and background cannot be distinguished effectively only from spatial or temporal information. In this case, there are new approaches that combine the two methods.[3–5] For example, Wang *et al*. proposed a video saliency detection algorithm based on the spatiotemporal gradient.[3] Wei *et al*. proposed a video object segmentation model using multiframe and multiscale features through the feedback of back layer features.[4] Zeng *et al*. proposed a video object segmentation algorithm based on prior probability and metric learning.[5] These video object segmentation algorithms improve the segmentation accuracy by updating the reference space with features having high classification confidence. However, the reference space significantly reduces the speed of the algorithm because its size is gradually increased by introducing more classification confidence features.

Shadow elimination is a necessary step to distinguish dynamic objects and their shadows. Most of the existing algorithms use the inherent characteristics of shadows to identify shadow regions.[6–10] The use of a color feature eliminates shadows according to the characteristics of the decrease in the brightness value in the shadow area; the use of an edge feature eliminates shadows according to the large difference between the shadow and the background; the use of a texture feature eliminates shadows according to the slight difference in the characteristics of the texture feature before and after shadow area coverage. For this reason, Qu *et al*. analyzed the characteristics of shadow and provided a reference for shadow detection algorithms.[6] Kar and Deb,[7] Jiang *et al*.,[8] and Wang and Suter[9] respectively removed shadows in HSV color space, YUV color space, and RGI color space. Farou *et al*.[10] combined the best color features of different color spaces to determine the shadow area. Although the above methods improve the shadow elimination algorithm based on color features, there are still many problems when using only a single feature in complex scenes. Therefore, more features are needed to eliminate shadow.

In video object dynamic tracking, the mainstream video object dynamic tracking algorithms are divided into two categories: generative object tracking algorithms[11–13] and discriminant object tracking algorithms.[14–17] Generative object tracking algorithms establish the object model through online learning, then use the model to search the image area to find the smallest reconstruction error to complete the object positioning. This kind of tracking algorithm has advantages of high stability, no need for training, and low computational requirement. However, only the region most similar to the object model is considered in the tracking process, which leads to low tracking accuracy. The discriminant object tracking algorithms regard

object tracking as a binary classification problem. They extract both object and background information to train the classifier. Then the object is separated from the background of the image sequence by the classifier. Finally, the position of the object in the current frame is obtained. The discriminant object tracking algorithms include those based on correlation filtering algorithms and deep learning algorithms. Although discriminative object tracking algorithms have good tracking accuracy, they are based on deep learning, which requires a lot of data training. Owing to the weak generalization of this kind of algorithm and the difficulty of training data to cover all actual scenes, the accuracy of object-tracking algorithms based on deep learning is significantly reduced when there are no training scenes.

In conclusion, with the development of vehicle tracking research, the current tracking algorithms have made progress in video object segmentation and dynamic object tracking, but further improvement is required to deal with the shadow interference caused by illumination changes and to overcome the difficulty of single object extraction caused by vehicle occlusion. This paper proposes a method of unsupervised static recognition and dynamic tracking for vehicles. To solve the problem of unclear edge information in video object segmentation, an improved Canny algorithm is proposed. To solve the problem that the tracking object feature is similar to the shadow feature, chroma information is introduced into a statistical nonparametric quantization algorithm to achieve shadow elimination. In addition, we use the mean-shift algorithm to track the dynamic object in real time.

2. Related Work

2.1 Video object segmentation

Video object segmentation is the most basic step in vehicle tracking. At present, video object segmentation is mainly realized by establishing a mathematical model or neural network.[18] A method based on a neural network requires a large number of training data samples and a large amount of calculation. Since the video data studied in this paper is a vehicle-monitoring video with a relatively fixed background, the method of building a neural network is more complex and inefficient than that of establishing a mathematical model. Therefore, a Gaussian mixture model (GMM) is used for video object segmentation.

As a pixel-level video object segmentation model, the GMM can realize multimodal simulation in complex scenes by the linear combination of multiple Gaussian distributions and modeling each pixel.

The calculation process of the GMM is described as follows. Firstly, K Gaussian models are established for each pixel point in RGB color space to represent the features of each pixel point in the image. Features that can be described include color, depth, gradient, and brightness. The value of pixel i in RGB space at time t is denoted as $x_{i,t}$, and the probability that it belongs to the background is obtained by the following equation:

$$P(\Theta_B) = \sum_{i=1}^{K} \frac{\omega_{i,t}}{\sqrt{2\pi}\sigma_{i,t}} e^{-\frac{1}{2}\left(\frac{x_{i,t}-\mu_{i,t}}{\sigma_{i,t}}\right)^2}, \quad (1)$$

where Θ_B represents the background. K is the number of Gaussian components representing pixel features and its value is generally 3–5. $\omega_{i,t}$ is the ith Gaussian model weight at time t. $\mu_{i,t}$ and $\sigma_{i,t}^2$ are the mean and variance of Gaussian model i at time t, respectively.

The K Gaussian distributions used to describe the color distribution of each point are sorted in order of decreasing ω/σ. The threshold $T \in [0,1]$ is chosen to determine the number of Gaussian components that can be retained. Only the first b Gaussian distributions above this threshold are considered as the background distribution B, and the others are considered as the foreground distribution F. The background distribution B is expressed by the following equation:

$$B = \arg\min_b \left(\sum_{i=1}^{b} \omega_{i,t} > T \right). \tag{2}$$

Each pixel in the current image is matched with all the Gaussian components one by one as follows:

$$\left| x_{i,t} - \mu_{i,t-1} \right| \le 2.5\sigma_{i,t-1}. \tag{3}$$

If the difference between a pixel and a Gaussian component in B satisfies the above formula, it indicates that the pixel is a background point; if the difference between the pixel and a Gaussian component in F satisfies the above formula, it indicates that the pixel is a foreground point; if the pixel does not match any Gaussian component in the GMM, a new Gaussian distribution is defined to replace the Gaussian component with the lowest $\omega_{i,t}$. The mean of the new Gaussian distribution is initialized to the current pixel value and its standard deviation and weight are set to 30 and $1/K$, respectively. The purpose of setting a large standard deviation is to include as many pixels as possible in the model so as to obtain the most likely model. The reason for setting the weight to $1/K$ is that the most likely Gaussian model is obtained by setting a low weight to update its parameters.

When the matching between pixels and the Gaussian model is completed, all the Gaussian components are updated:

$$\begin{aligned} \omega_{i,t} &= (1-\alpha)\omega_{i,t-1} + \alpha M_{i,t}, \\ \mu_{i,t} &= (1-\rho)\mu_{i,t-1} + \rho X_{i,t}, \\ \sigma_{i,t}^2 &= (1-\rho)\sigma_{i,t-1}^2 + \rho\left(X_{i,t} - \mu_{i,t}\right)^2, \end{aligned} \tag{4}$$

where the renewal rate parameters α and ρ satisfy $0 \le \alpha \le 1$ and $\rho = \alpha/\omega_{i,t}$, respectively. $M_{i,t}$ represents the matching result between the pixels and the Gaussian model in time t. If the match is successful, then $M_{i,t} = 1$; otherwise, $M_{i,t} = 0$.

2.2 Shadow elimination

Owing to the influence of illumination in a real environment, a shadow will inevitably appear in the tracking object and background, which will reduce the accuracy of vehicle object tracking. To reduce the influence of the shadow on tracking results, the statistical nonparametric quantization (SNP) algorithm is used as a good shadow elimination algorithm.

The SNP theory assumes that the expected brightness of pixel i in a video frame is $E_i = [E_R(i), E_G(i), E_B(i)]$, the brightness of pixel i in the current frame is $I_i = [I_R(i), I_G(i), I_B(i)]$, CD_i is the chromaticity distortion of pixel i, and $\varphi(\alpha_i)$ is the luminance distortion of pixel i.

$\varphi(\alpha_i)$ is a scalar value that brings the observed color close to the expected chromaticity. It is given by $\varphi(\alpha_i) = \min (I_i - \alpha_i E_i)^2$, where α_i represents the brightness of the pixel with respect to the expected value. $\alpha_i = 1$ indicates that the pixel is regarded as background, $\alpha_i < 1$ indicates that the pixel is regarded as shadow, and $\alpha_i > 1$ indicates that the pixel is regarded as dynamic foreground. Chromaticity distortion is defined as the orthogonal distance between the observed color and the expected chromaticity line. The chromaticity distortion of pixel i is given by $CD_i = \|I_i - \alpha_i E_i\|$. Figure 1 illustrates the chromaticity distortion in RGB space.

By calculating the channel average value of pixel i in the first N frames, the mean vector $\overline{E_i}$ is obtained. $\overline{E_i}$ is used as an estimate of the luminance expected value E_i. It is represented by $\overline{E_i} = [\mu_R(i), \mu_G(i), \mu_B(i)]$, where $\mu_R(i)$, $\mu_G(i)$, and $\mu_B(i)$ represent the arithmetic means of the R, G, and B components of pixel i, respectively. By substituting $\overline{E_i}$ into $\varphi(\alpha_i)$ and CD_i, the initial values of luminance distortion and chromaticity distortion $\varphi(\alpha_0)$ and CD_0, respectively, are obtained:

$$\varphi(\alpha_0) = \min\left[\left(I_R(i) - \alpha_i \mu_R(i)\right)^2 + \left(I_G(i) - \alpha_i \mu_G(i)\right)^2 + \left(I_B(i) - \alpha_i \mu_B(i)\right)^2\right], \tag{5}$$

$$CD_0 = \sqrt{\left(I_R(i) - \alpha_i \mu_R(i)\right)^2 + \left(I_G(i) - \alpha_i \mu_G(i)\right)^2 + \left(I_B(i) - \alpha_i \mu_B(i)\right)^2}. \tag{6}$$

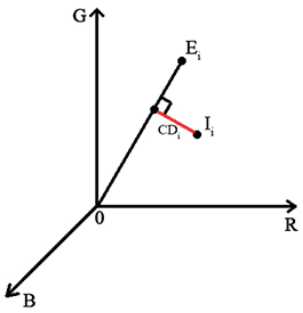


Fig. 1. (Color online) Chromaticity distortion in RGB space.

Owing to the different sensitivities of cameras to different color channels, the variation of the values of the three channels is not consistent. To solve this problem, the standard deviation $S_i = [\sigma_R(i), \sigma_G(i), \sigma_B(i)]$ of each channel is used as the weight to correct the sensitivity of Eqs. (5) and (6), and we obtain

$$\varphi(\alpha_i) = \min\left[\left(\frac{I_R(i)-\alpha_i\mu_R(i)}{\sigma_R(i)}\right)^2 + \left(\frac{I_G(i)-\alpha_i\mu_G(i)}{\sigma_G(i)}\right)^2 + \left(\frac{I_B(i)-\alpha_i\mu_B(i)}{\sigma_B(i)}\right)^2\right], \tag{7}$$

$$CD_i = \sqrt{\left(\frac{I_R(i)-\alpha_i\mu_R(i)}{\sigma_R(i)}\right)^2 + \left(\frac{I_G(i)-\alpha_i\mu_G(i)}{\sigma_G(i)}\right)^2 + \left(\frac{I_B(i)-\alpha_i\mu_B(i)}{\sigma_B(i)}\right)^2}. \tag{8}$$

If $\varphi(\alpha_i) < 0$, the pixels are shadow points; if $\varphi(\alpha_i) > 0$, the dynamic objects are distinguished by CD_i.[19]

2.3 Dynamic object tracking

To locate all the objects in the video image in real time, it is necessary to track the dynamic vehicle on the basis of static vehicle object recognition. In this paper, the mean-shift tracking algorithm is used to achieve the real-time tracking of dynamic vehicles. This is an adaptive gradient algorithm based on kernel density estimation. It uses a probability density function and kernel function to describe the research target.

In the mean-shift algorithm, n sample points are assumed to exist in d-dimensional space R^d: x_i ($i = 1, ..., n$). For a point in this space, the mean-shift vector M_h can be expressed as

$$M_h = \frac{1}{k}\sum_{x_i \in S_h}(x_i - x), \tag{9}$$

where k is the number of sample points x_i in S_h, a high-dimensional spherical region with radius h. The sample points in S_h are distributed along the direction of increasing gradient value of the probability density. The sample point y is a member of the following set:

$$S_h = \{y: (y-x)^T(y-x) \le h^2\}. \tag{10}$$

3. Method

3.1 Proposed framework

In this paper, the object recognition and tracking in a video sequence are divided into three parts: static object recognition, dynamic object recognition, and dynamic object tracking. Figure 2 shows the flowchart of the algorithm in this paper.

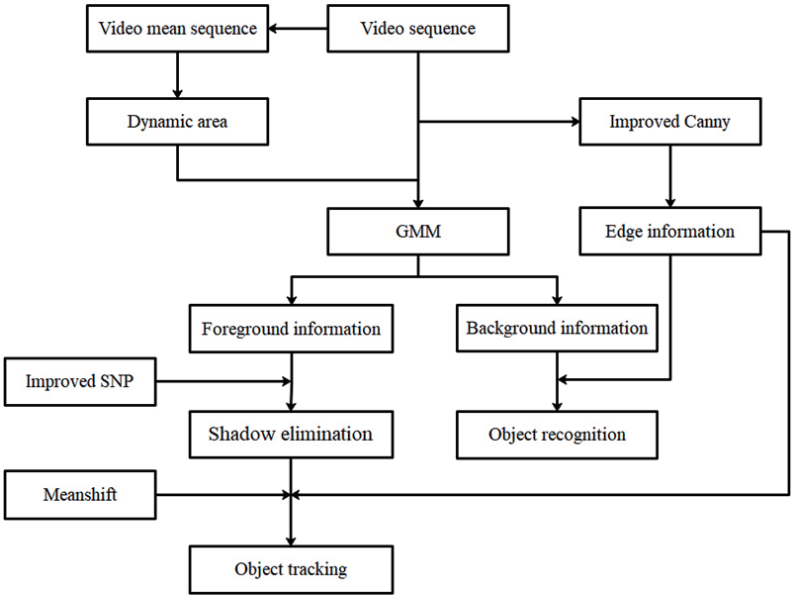


Fig. 2. Algorithm flowchart.

3.2 Static object recognition

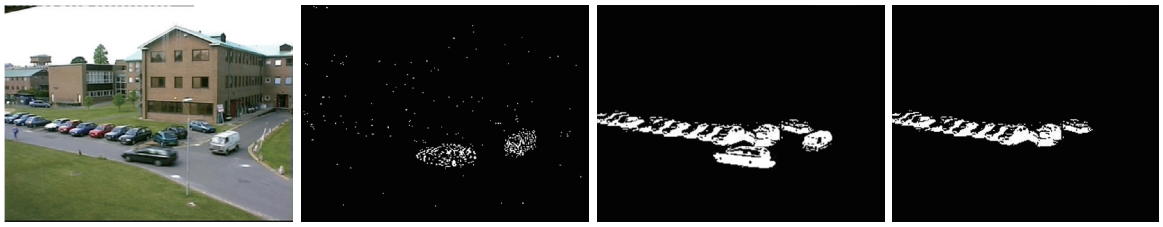
The vehicle in a video sequence may be stationary for a long time. Therefore, identifying the static object stably in the video is an important task in this algorithm. Considering that the GMM can extract the object of interest from the current frame, how to distinguish the dynamic object from the static object and how to locate the static object are the most important problems in this part.

The recognition of a stationary object is shown in Fig. 3. We use the dynamic area D and the foreground information F to perform the intersection calculation, then a rough position of the dynamic object in the foreground information is obtained, and the rest of the foreground information is considered as the static object S, which is expressed by the following equation:

$$S = F - (D \cap F). \tag{11}$$

The static object is located by an edge detection algorithm. To achieve this, the Canny algorithm is the best algorithm. However, the high and low thresholds of the Canny algorithm are fixed, meaning that it is not suitable for edge detection in dynamic scenes. Therefore, this paper proposes an improved Canny algorithm based on the idea of Zhang *et al.*[20] to ensure that high-quality edge information can still be obtained in changing scenes.

We use the maximum interclass variance algorithm (Otsu) to adaptively select the high threshold and low threshold when edge connection occurs. The Otsu algorithm first calculates the gray level range of the image, and then divides the pixels into two categories, called C_0 and C_1, by setting a certain gray value T. The optimal threshold is calculated as the maximum variance between classes. The high and low thresholds are denoted by T_h and T_l, respectively,



(a) (b) (c) (d)

Fig. 3. (Color online) Recognition of stationary object: (a) video mean sequence, (b) dynamic area, (c) foreground area obtained by GMM, and (d) stationary object recognition.

$$T_h = \arg\max_{1\le t\le L}\{\sigma(t)\} = \arg\max_{1\le t\le L}\{P_0(t)u_0^2(t) + P_1(t)u_1^2(t)\}, \tag{12}$$

where $\sigma(t)$ is the interclass variance of the image, L is the range of the gray level, $P_0(t)$ and $P_1(t)$ denote the probabilities of occurrence of C_0 and C_1, and $u_0(t)$ and $u_1(t)$ denote the average gray levels of C_0 and C_1, respectively. The low threshold is given by $T_l = T_h / 2$.

The Canny algorithm calculates the gradient amplitude and direction of the image using the first-order partial derivative 2 × 2 finite difference, so it can only calculate the gradient amplitude in the vertical, horizontal, and diagonal directions. Owing to the large interval between different directions, the Canny algorithm is fuzzy for the edge perception of nonspecific angles. Therefore, in this paper, all gradient directions are divided into eight parts to enhance the perceptual strength of the Canny algorithm for random angle edges. At the same time, a 3 × 3 edge detection template is used to calculate the gradient amplitude and direction, so as to reduce noise interference. The template for the gradient calculation in eight directions is shown in Fig. 4.

Taking the 0° direction as an example, the mathematical expression for the partial derivative $G_1(x, y)$ in this direction is as follows:

$$\begin{aligned} G_1(x,y) = [&g(x-1,y+1) + 3g(x,y+1) + 3g(x+1,y+1) \\ &-g(x-1,y-1) - 3g(x,y-1) - 3g(x,y-1)] \end{aligned}. \tag{13}$$

By calculating the directional partial derivatives in eight directions, we can calculate the gradient direction $\alpha(x, y)$ and gradient amplitude $M(x, y)$ at the pixel (x, y) as follows:

$$\alpha(x,y) = \tan^{-1}\left(\frac{G_y(x,y)}{G_x(x,y)}\right), \tag{14}$$

$$M(x,y) = \sqrt{\sum_{i=1}^{8} G_i^2(x,y)}. \tag{15}$$

-1	-3	-1
0	0	0
1	3	3

(a)

-2	-2	0
-1	0	1
0	2	2

(b)

-3	-2	0
-2	0	2
0	2	3

(c)

-2	-1	0
-2	0	1
0	2	2

(d)

-1	0	1
-3	0	3
-1	0	1

(e)

0	1	2
-1	0	2
-2	-2	0

(f)

0	2	3
-2	0	2
-3	-2	0

(g)

0	2	2
-2	0	1
-2	-1	0

(h)

Fig. 4. Detection templates in eight directions: (a) 0° direction, (b) 22.5° direction, (c) 45° direction, (d) 67.5° direction, (e) 90° direction, (f) 112.5° direction, (g) 135° direction, and (h) 167.5° direction.

(a)

(b)

(c)

Fig. 5. (Color online) Results of edge detection by improved Canny algorithm: (a) original image, (b) result of edge detection by Canny algorithm, and (c) result of edge detection by improved Canny algorithm.

Through finding the gradient direction $\alpha(x, y)$ and substituting it into the eight directions we designed, we can achieve more accurate edge detection. Figure 5 shows the edge detection results of the improved Canny algorithm for a certain frame in a video compared with those of the traditional Canny algorithm. We can see that the edge contours acquired by the improved Canny algorithm are clearer; there is less noise in the image, and the edges are well connected.

3.3 Dynamic object recognition

The dynamic object area contains not only the dynamic object but also shadow caused by illumination. To distinguish the dynamic object and shadow in the foreground image and reduce the error of subsequent processing, it is necessary to eliminate the shadow part in the foreground region. In this paper, chroma information is introduced into the SNP algorithm to eliminate shadow. As shown in Sect. 2.2, the SNP algorithm distinguishes the object from the shadow using only the value of luminance distortion. This algorithm achieves good results for removing shadow in a simple video environment but will not be suitable in a

complex environment. Inspired by Wang and Suter,[21] the brightness threshold is set using the traditional SNP shadow elimination algorithm. When the scene brightness is higher than the threshold value, the normalized color feature space is used. In contrast, the original color feature space is used for shadow elimination. By using a different color space for shadow elimination when the brightness information is different, false detection caused by a change in brightness is effectively reduced. The color feature of pixel x is set as follows:

$$x = \begin{cases} (r, g, b) \text{ if } I \geq L \\ (R, G, B) \text{ if } I < \mathrm{L} \end{cases}, \tag{16}$$

where (r, g, b) is the color eigenvalue obtained by normalization, (R, G, B) is the real color eigenvalue, and L is the luminance threshold.

Considering the low brightness of the shadow part and the limitations of the light sensor hardware, the noise distribution in the shadow part is relatively dense, which leads to a change in the chrominance of the shadow area. Therefore, we propose a method of shadow elimination by using luminance information and chroma information at the same time.

The sensitivity-corrected luminance distortion $\varphi(\alpha_i)$ and chromaticity distortion CD_i are obtained (see Sect. 2.2), then the root mean square (RMS) values of $\varphi(\alpha_i)$ and CD_i, $RMS[\varphi(\alpha_i)]$ and $RMS(CD_i)$, respectively, are calculated. Then the following values are obtained.

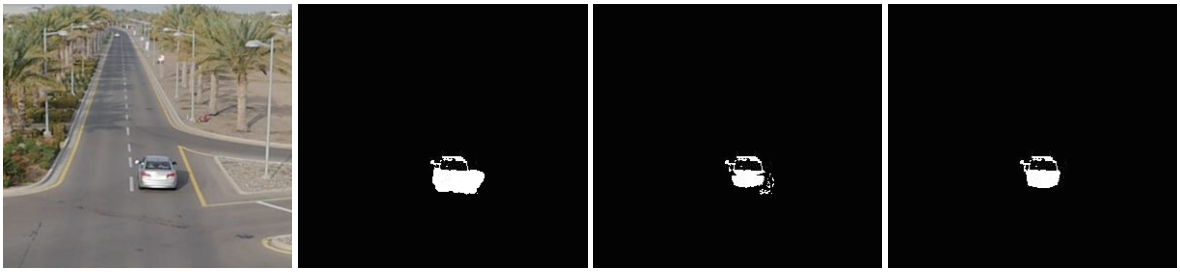
$$\hat{\varphi}(\alpha_i) = \frac{\alpha_i - 1}{RMS[\varphi(\alpha_i)]}, \tag{17}$$

$$\hat{CD}_i = \frac{CD_i}{RMS(CD_i)}. \tag{18}$$

The dynamic object and shadow are determined by setting the following threshold:

$$X(i) = \begin{cases} \hat{\phi}(\alpha_i) < 0 \text{ and } \hat{CD}_i > 50 & i \in B, \\ \hat{\phi}(\alpha_i) < 0 \text{ and } \hat{CD}_i < 50 & i \in S, \\ \hat{\phi}(\alpha_i) > 0 \text{ and } \hat{CD}_i > 50 & i \in D, \\ \text{else} & i \in F, \end{cases} \tag{19}$$

where B represents the background, S represents the shadow, D represents different objects in the foreground, and F represents the same object in the foreground. Figure 6 shows the elimination effect of the improved SNP algorithm and the SNP algorithm for a certain frame shadow in the video. We can see that when introducing chroma information to assist shadow elimination, the brake lamp is not recognized as shadow in Fig. 6(d). However, the chroma



(a) (b) (c) (d)

Fig. 6. (Color online) Results of elimination by improved SNP algorithm: (a) original image, (b) regions extracted by GMM, (c) result of shadow elimination by SNP algorithm, and (d) result of shadow elimination by improved SNP algorithm.

information of the tire is similar to that of the shadow, which leads to the misjudgment of the tire in Fig. 6(d). Owing to the deviation of the brightness information in the shadow area, the edge of the shadow is mistakenly detected in Fig. 6(c), while better shadow suppression is obtained in Fig. 6(d).

3.4 Dynamic object tracking

Different motion states of an object will change its shape, and the mean-shift algorithm cannot reflect a change in object shape during tracking. In this paper, the mean-shift tracking algorithm based on edge information is used to track the dynamic object.

First, we select the object model. The edges of foreground objects D and F obtained in Sect. 3.3 are detected by using the improved Canny algorithm, then the object model is determined by a morphological closed operation, and the object model is represented by a rectangular box. The Gaussian kernel function is used to calculate the object model, and the bandwidth h of the Gaussian kernel function is the size of the object area. All the pixel values in the video frame are divided into n intervals, and each interval corresponds to an eigenvalue u according to the size of the range. The probability density of each object model feature is calculated using the following equation:

$$\hat{q}_u = C\sum_{i=1}^{n} k\left(\left\|\frac{x_0 - x_i}{h}\right\|^2\right)\delta[b(x_i) - u], \tag{20}$$

where C is the normalization constant of the object model and $k(x)$ is the Gaussian kernel function used for pixel weighting. The Kronecker delta function $\delta[b(x_i) - u]$ is used to determine whether the pixel value x_i in the object model is equal to the characteristic value u.

Second, we select the candidate model. The object frame region may be included in the subsequent frames, and the center coordinate of the region is the center coordinate y of the kernel function. The feature probability distribution of the candidate model is similar to that of the object model, which is expressed by the following equation:

$$\hat{p}_u(y) = C_h \sum_{i=1}^{n} k\left(\left\| \frac{y - x_i}{h} \right\|^2 \right) \sigma[b(x_i) - u]. \tag{21}$$

Third, we compare the similarity of functions. The Bhattacharyya coefficient is selected as the similarity function, which is expressed by the following equation:

$$\rho[\hat{p}(y), \hat{q}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(y)\hat{q}_u}. \tag{22}$$

The similarity between the candidate model and the object model is judged by the Bhattacharyya coefficient. The mean-shift vector m of the object model is obtained as follows:

$$m = \frac{\sum_{i=1}^{n} x_i g\left(\left\| \frac{x - x_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n} g\left(\left\| \frac{x - x_i}{h} \right\|^2 \right)} - x, \tag{23}$$

where $g(x) = -k'(x)$.

Finally, we determine the object area. The center of the object box in the previous frame is taken as the center of the search window. The mean-shift vector is iterated continuously to find the candidate area that maximizes the similarity function. We update the object box and repeat until the video and the tracking end.

4. Experimental Results and Analyses

4.1 Experimental environment and dataset

The experimental hardware environment was an Intel Xeon E5-1603 v4 CPU with a clock speed of 2.80 GHz and an NVIDIA NVS 315 GPU. The algorithm was developed in Python and the tracking visualization was based on OpenCV. We selected challenging image sequences from the VOT2013, OTB2015, and VOT2017 datasets, which included illumination changes, rapid motion, object rotation, and occlusion. The video images had a height of 280 pixels, a width of 320 pixels, and a frame rate of 25 frames per second (FPS).

4.2 Qualitative analysis

We tested and evaluated our proposed vehicle tracking algorithm. Figure 7 shows the vehicle motion scene and the tracking results in the experiment. The first line of images is the first frame of each video sequence, and the remaining lines are the tracking results of the algorithm and the actual position of the object vehicle.

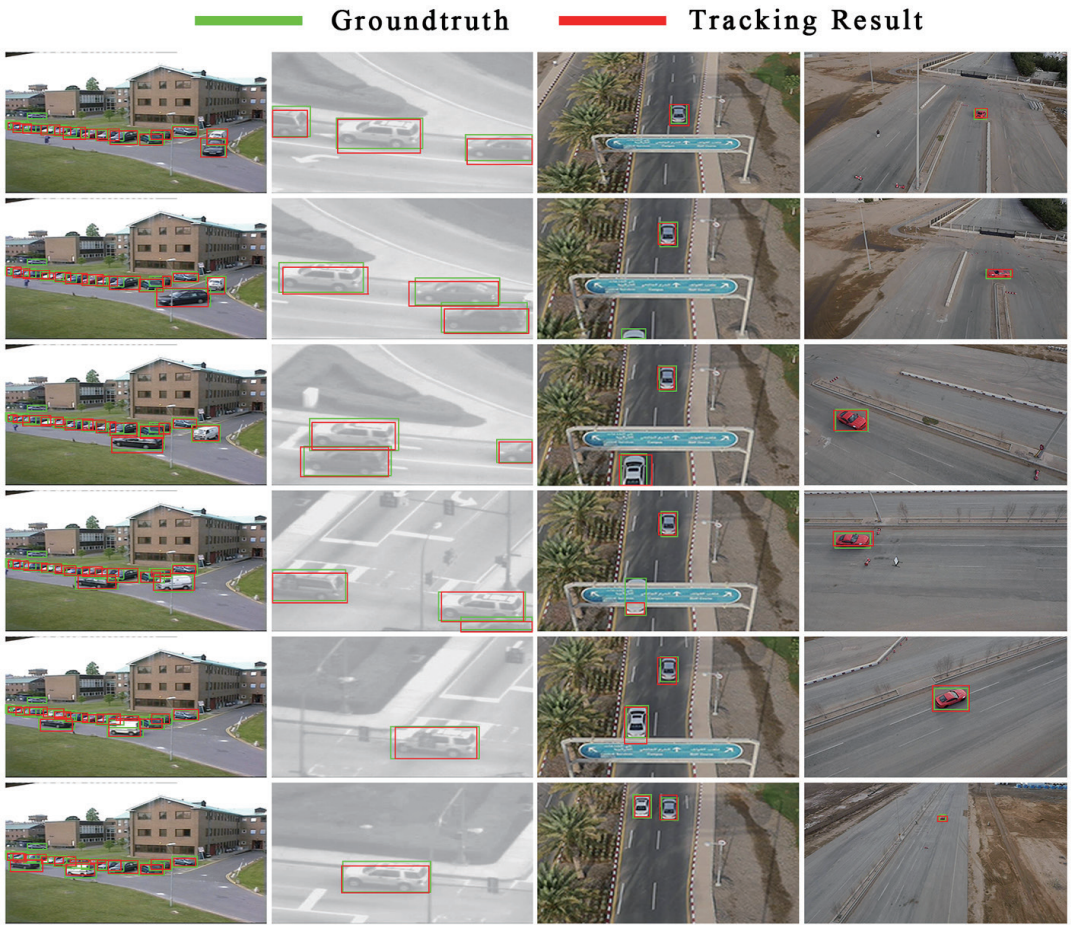


Fig. 7. (Color online) Vehicle tracking results.

The first scene [Fig. 7(a)] shows the tracking of slow-moving vehicles from the upper right view. The tracking objects are black and white vehicles with high definition. The object vehicle runs slowly on the road, and the two vehicles intersect during driving. As can be seen from Fig. 7(a), this algorithm can stably track slow-moving vehicles and distinguish vehicles of different colors and models. When the black vehicle blocks the white vehicle, the tracking of the black vehicle has no offset, and the white vehicle can be continuously tracked after partial occlusion. Meanwhile, vehicles stopped on the road can be detected. However, the silver vehicle in the upper left corner, whose color is similar to the background, is not detected.

The second scene [Fig. 7(b)] shows the tracking of vehicles moving in and out of the field of vision, also from the upper right view, where the tracking objects are black and white vehicles with ordinary definition. At first, the white vehicle is waiting for the traffic lights. During the waiting process, the black vehicle enters the camera area, and the camera follows the black vehicle. After the green light, the white vehicle moves forward along the road, and the black vehicle turns left to leave the camera area, and the camera follows the white vehicle. We can see that this has little effect on the tracking accuracy of the dynamic camera, i.e., when the object moves in or out of the field of view, the object can be tracked well.

Figure 7(c) shows the tracking situation when the vehicle is occluded. This scene shows the view from above the vehicle. The tracking object is a white vehicle with high definition that is running with a high speed on the road. During the tracking, another white car enters the field of vision, and then the vehicle is blocked by a road sign. As can be seen from Fig. 7(c), this algorithm can track fast-moving vehicles since the tracking of the object vehicle has no offset. The second white vehicle is not tracked when it first enters the field, but it is successfully tracked after completely entering the field. When the road sign almost completely blocks the vehicle and the tracking position deviates from the actual position of the vehicle, the vehicle cannot be detected or tracked until the entire vehicle reappears in the field of view.

Figure 7(d) shows the tracking of vehicle rotation and scale changes with the view from above the vehicle. The tracking object is a red vehicle with high definition. After turning around on the road, the object vehicle quickly approaches then moves away from the camera. It can be seen from Fig. 7(d) that the rotation and scale changes of the object vehicle have little influence on its tracking. However, once the vehicle is far away, the tracking frame is slightly offset from the object vehicle position.

4.3 Quantitative analysis

4.3.1 Evaluation of improved Canny algorithm

To verify the effectiveness of the improved Canny algorithm proposed in this paper, it is compared with the Canny algorithm on the same video. The accuracy and stability of the algorithm are evaluated by the spatial accuracy (SA) and temporal coherency (TC).[22] The SA reflects the shape similarity between the segmentation result of each frame and the reference segmentation template, and the TC reflects the similarity of the spatial accuracy between two adjacent frames in the video sequence. Figure 8(a) shows that the SA of the proposed method is about 10% higher than that of the Canny algorithm, indicating that its segmentation accuracy is higher. Figure 8(b) shows that the TC of the proposed method is between 0.87 and 1, and the fluctuation range is small. The TC for the Canny algorithm is between 0.8 and 1, and the fluctuation range is large. This shows that the stability of the improved Canny algorithm is

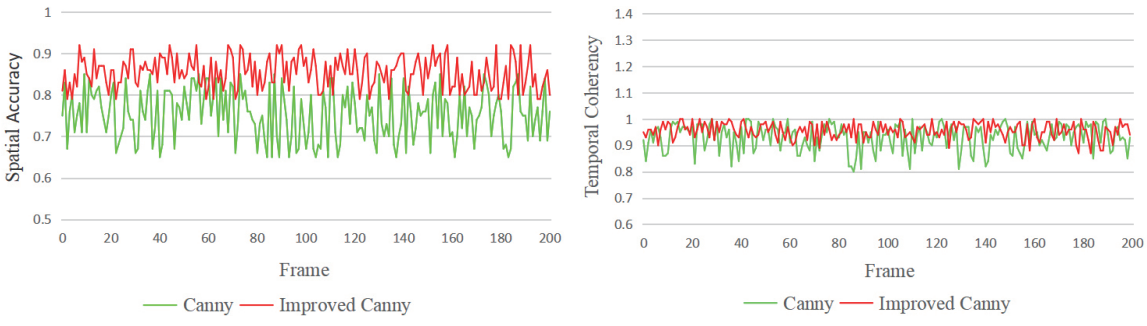


Fig. 8. (Color online) Performance comparison of algorithms for the same video sequence: (a) spatial accuracy and (b) temporal coherency.

effectively increased by automatically setting thresholds for different frames. Table 1 shows the SA and TC of the improved Canny algorithm with different numbers of directions. We find that when the number of directions is eight, good results are achieved for the SA and TC.

4.3.2 Evaluation of improved SNP algorithm

To verify the performance of the shadow elimination algorithm proposed in this paper, we compare the improved SNP algorithm and the SNP algorithm for the video sequences in Figs. 7(a)–7(d) in Sect. 4.2. The accuracy of the algorithms is evaluated by the shadow detection rate η and the shadow discrimination rate ξ.[23] By calculating η, the recall rate of the shadow elimination algorithm for shadow areas can be evaluated. By calculating ξ, the precision of the shadow elimination algorithm for shadow areas can be evaluated. As shown in Table 2, the average values of η and ξ of the improved SNP algorithm are 2.5 and 7.3% higher than those of the SNP algorithm. Therefore, the shadow elimination effect of the improved SNP algorithm is better than that of the SNP algorithm.

4.3.3 Comparison with other methods

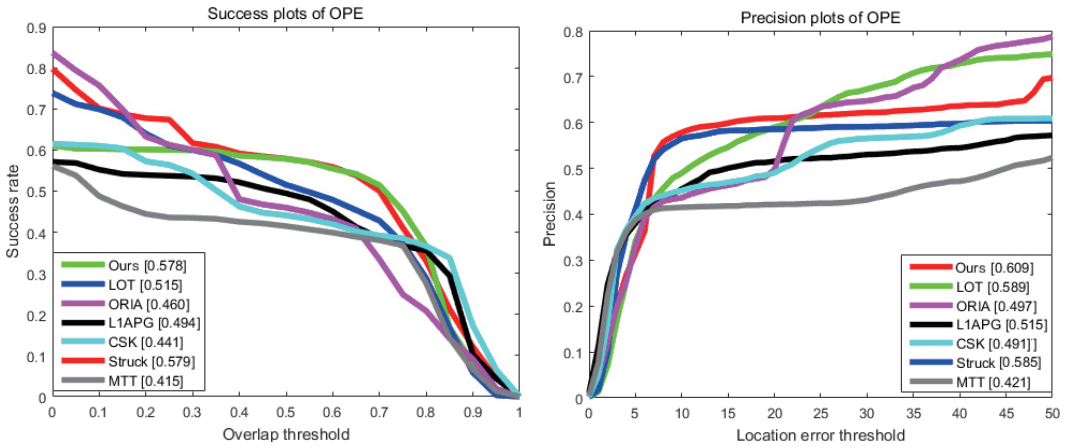
To better evaluate the effect of our algorithm for vehicle tracking, we not only use the precision and success rate to analyze our method and other methods, but also use the quantitative index FPS as the speed evaluation index. Figure 9 shows the precision plots and success rate plots[24] of the experimental results of our method and six other trackers: LOT,

Table 1
Comparison of algorithm performance for different numbers of directions.

Number of directions	Average SA	Average TC
4	77.43	0.973
5	79.56	0.957
6	82.31	0.963
8	84.06	0.979
9	84.07	0.953
10	83.62	0.978
12	83.24	0.959

Table 2
Analysis of shadow detection rate and shadow discrimination rate.

Video sequence	Evaluation standard	SNP (%)	Improved SNP (%)
a	η	72.8	72.6
	ξ	88.9	89.3
b	η	80.5	83.2
	ξ	74.2	87.6
c	η	82.9	81.1
	ξ	77.3	88.7
d	η	84.0	92.0
	ξ	92.3	92.4



(a) (b)

Fig. 9. (Color online) Precision and success rate plots of the different trackers on the OTB vehicle dataset: (a) precision plots and (b) success rate plots.

Table 3
Speed comparison of different trackers.

	Ours	LOT	ORIA	L1APG	CSK	Struck	MTT
Speed (FPS)	26.33	3.12	14.62	3.64	40.57	20.26	3.86

ORIA, L1APG, CSK, Struck, and MTT. In the precision plot, our algorithm ranks first; in the success rate plot, our algorithm ranks second. Table 3 shows the time cost of our method and the other methods for processing the same video. It can be seen that our algorithm can basically achieve real-time tracking.

5. Discussion

By introducing edge information into the object tracking algorithm, we realized the tracking of vehicles without providing prior information, and we carried out comparison experiments with many other tracking algorithms. The results show that our method has the advantage of a high speed without training. In this part, we will discuss several factors that influence the tracking results of this algorithm, including the selection and improvement of the edge detection algorithm, the selection and improvement of the shadow elimination algorithm, and the limitations of the algorithms.

5.1 Selection and improvement of edge detection algorithm

Currently, edge detection operators include the Roberts, Sobel, Prewitt, Laplacian, LOG, and Canny algorithms. Among them, the Roberts, Sobel, and Prewitt algorithms, as edge detection algorithms based on the first-order differential, have particular importance. The Roberts algorithm was the earliest algorithm devised for image edge detection. It has a good calibration effect for vertical edges but a poor detection effect for other angles, and its ability to suppress

noise is weak. The Sobel algorithm has the advantages of a low computational cost and high speed. However, its accuracy is low and it is only effective for horizontal and vertical edge detection. The Prewitt algorithm has better edge detection than the Roberts algorithm in both the horizontal and vertical directions, but an unreasonable setting of the gray threshold results in poor noise suppression. The Laplacian and LOG algorithms are second-order differential algorithms. The Laplacian algorithm can detect an edge of any direction, but the influence of noise is the most serious among the algorithms. The LOG algorithm is relatively accurate in detecting edge positions, but false edges caused by noise are easily detected. In general, although traditional first- and second-order differential algorithms have high detection accuracy in the horizontal and vertical directions, they have low detection accuracy in other directions. These algorithms are not suitable for targets with multidirectional edges such as vehicles. Compared with the first- and second-order differential algorithms for edge detection, the Canny algorithm has more theoretical and practical importance because of its better robustness and complete edge detection mechanism. Therefore, the Canny algorithm was improved in this study. Considering that the double threshold setting in the traditional Canny edge detection algorithm cannot achieve adaptive adjustment under changing scenarios and its anti-noise ability is low, the double threshold is adaptively adjusted by the Otsu method in this study. This ensures that high-quality edge information is obtained in a changing scene. At the same time, we have extended the detection template of the Canny algorithm in different directions. By increasing the number of sensing directions, the perceptual ability of the algorithm for random angle edges is enhanced.

5.2 Selection and improvement of SNP algorithm

Shadow elimination methods can be divided into two categories: deterministic and statistical methods. A deterministic method determines whether a pixel is a shadow through the known background and object characteristics in the video. According to whether it is necessary to establish a model in the decision process of the deterministic method, deterministic methods can be divided into the deterministic nonmodel-based method and deterministic model-based method. In the modeling process, the deterministic model-based method relies on parameters such as the lighting conditions and number of objects in the scene, so it has high computational complexity in a complex environment. Moreover, in a complex environment, this kind of method cannot achieve shadow elimination in multiple scenes through a unified model, so there has been relatively little research on this method. The computational cost of the deterministic nonmodel-based method is relatively low. However, owing to the limitations of the model, when the image content changes rapidly, the accuracy of shadow elimination is significantly reduced. The statistical methods have a certain degree of robustness against interference. They classify pixel points by the probability values of pixels, so the parameters of the statistical function used are essential. The statistical methods can be further divided into the statistical parametric method and statistical nonparametric method according to the method used to obtain the parameters. Owing to the large amount of calculation and low adaptability to changing scenes, the statistical parameter method is more suitable for shadow elimination in indoor scenes.

The target of this paper is vehicles, which are usually in outdoor scenes, so the statistical parameter method is not applicable. To sum up, by referring to the applicability and real-time performance of various methods, we have improved the statistical nonparametric model by introducing chroma information into pixel category determination. Our experiments show that the introduction of chroma information can effectively reduce false detection. At the same time, the chroma information can also be used to quadratically discriminate the foreground region extracted by the GMM, which improves the accuracy of the dynamic region and the tracking performance of the algorithm.

5.3 Limitations

Although our experiments show that our algorithm is effective, there are still some problems. For example, although the algorithm can solve the problem of partial occlusion of the object vehicle to a certain extent, tracking failure will still be caused by excessive occlusion. In Fig. 7(c), when the tracking object was occluded by a large area for a long time, the proposed algorithm exhibited tracking drift; since the algorithm locates the vehicle by edge detection, it is easy for vehicle detection failure to occur when the object is similar to the background because only edge information is used; there is a trade-off between the shadow elimination rate and target integrity. When the shadow removal rate is high, the integrity of the moving object is difficult to ensure. However, when the moving object is complete, the shadow removal rate will be reduced. However, tracking incomplete objects will make the object frame unable to completely cover all objects, which reduces the location precision of the object. However, in this paper, we mainly discuss the vehicle tracking problem without large changes in the background, a complex background, or multiple view changes. We will consider these issues in future research.

6. Conclusions

The development of sensors provides a means of vehicle detection and recognition, which are research hotspots in computer vision. In recent years, vehicle tracking algorithms have made great progress in real-time tracking and accuracy. To solve the problems of mutual occlusion and shadow interference in vehicle tracking, we presented an unsupervised vehicle object tracking algorithm suitable for both dynamic and stationary states. This algorithm shows good performance on public datasets. Experiments showed that:

(1) The proposed multidirectional Canny edge detection algorithm can extract edge information more effectively and reduce the incidence of false detection between different objects with a similar appearance.
(2) By introducing chroma information into the SNP model, shadow areas can be removed more accurately, increasing the accuracy of the object area. Thus, the tracking accuracy is improved.
(3) Compared with other methods, our algorithm, which basically meets the requirements of real-time tracking, can achieve better tracking results and has a higher speed.

(4) Compared with algorithms based on deep learning, our algorithm ensures better tracking accuracy. Moreover, it does not need a large number of samples to train the model or a lot of computing resources.

To obtain better vehicle tracking results, we will try to improve the recognition and tracking ability of the algorithm for large-area occlusion targets by adding trajectory prediction in the future. At the same time, we will do more in-depth research on the vehicle tracking problem when the background changes relatively rapidly, and try to enhance the tracking ability of the algorithm for vehicle objects by introducing optical flow information. We will also continue to optimize the algorithm structure and improve the efficiency of the algorithm.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant no. 41871367), the Ministry of Science and Technology of the People's Republic of China (grant no. 2018YFE0206100), and the BUCEA Postgraduate Innovation Project (PG2020086).

References

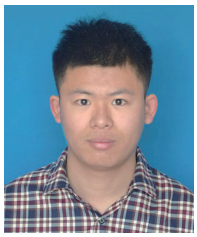
1 W. Wang, H. Song, S. Zhao, J. Shen, S. Zhao, S. C. H. Hoi, and H. Ling: Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (IEEE, 2019) 3059–3069.
2 L. Bao, X. Zhang, Y. Zheng, and Y. Li: Multimedia Tools Appl. **75** (2016) 7761. https://doi.org/10.1007/s11042-015-2692-4
3 W. Wang, J. Shen, and L. Shao: IEEE Trans. Image Process. **24** (2015) 4185. https://doi.org/10.1109/TIP.2015.2460013
4 J. Wei, S. Wang, and Q. Huang: Proc. the 34th AAAI Conf. Artificial Intelligence (AAAI, 2019) 12321–12328.
5 X. Zeng, R. Liao, L. Gu, Y. Xiong, S. Fidler, and R. Urtasun: Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (IEEE, 2019) 3928–3937.
6 L. Qu, J. Tian, H. Fan, W. Li, and Y. Tang: IET Comput. Vision **12** (2018) 95. https://doi.org/10.1049/iet-cvi.2017.0159
7 A. Kar and K. Deb: Proc. 2nd Int. Conf. Electrical Engineering and Information and Communication Technology (iCEEiCT, 2015) 21–23.
8 K. Jiang, A. Li, Y. Su, Z. Cui, and T. Wang: IET Comput. Vision **7** (2013) 115. https://doi.org/10.1049/iet-cvi.2012.0106
9 H. Wang and D. Suter: Pattern Recognit. **40** (2007) 1091. https://doi.org/10.1016/j.patcog.2006.05.024
10 B. Farou, H. Rouabhia, H. Seridi, and H. Akdag: Comput. Inf. **36** (2017) 837. https://doi.org/10.4149/cai_2017_4_837
11 R. E. Kalman: J. Fluids Eng. **82** (1960) 35. https://doi.org/10.1115/1.3662552
12 N. J. Gordon, D. J. Salmond, and A. F. M. Smith: IEEE Proc. Part F: Radar and Signal Processing. (IEEE, 1993) 107–113.
13 K. Fukunaga and L. D. Hostetler: IEEE Trans. Inf. Theory **21** (1975) 32. https://doi.org/10.1109/TIT.1975.1055330
14 M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg: Proc. European Conf. Computer Vision (ECCV, 2016) 1–9.
15 J. F. Henriques, C. Rui, M. Pedro, and B. Jorge: IEEE Trans. Pattern Anal. Mach. Intell. **37** (2015) 583. https://doi.org/10.1016/j.bmcl.2013.12.116
16 H. Nam and B. Han: Proc. IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition (IEEE, 2016) 4293–4302.
17 Y. Qi, S. Zhang, W. Zhang, L. Su, Q. Huang, and M. H. Yang: Proc. 33rd AAAI Conf. Artificial Intelligence (AAAI, 2019) 8835–8842.
18 P. Tokmakov, K. Alahari, and C. Schmid: Proc. IEEE Int. Conf. Computer Vision (IEEE, 2017) 4491–4500.
19 K. Onoguchi: Proc. 14th IEEE Int. Conf. Pattern Recognition (IEEE, 2002) 583–587.

20 H. Geng, M. Luo, and F. Hu: Proc. IEEE Int. Conf. Intelligent Human-Machine Systems and Cybernetics (IEEE, 2013) 527–530.
21 H. Wang and D. Suter: Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (IEEE, 2005) 1017–1020.
22 Z. Y. Zhang: Principle and Application of Video Object Segmentation and Extraction, G. B. Yang and Z. Liu Eds. (Science Press, Beijing, 2009) 1st ed., Chap. 3.
23 A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara: IEEE Trans. Pattern Anal. Mach. Intell. **25** (2003) 918. https://doi.org/10.1109/TPAMI.2003.1206520
24 Y. Wu, J. Lim, and M.H. Yang: IEEE Trans. Pattern Anal. Mach. Intell. **37** (2015) 1834. https://doi.org/10.1109/TPAMI.2014.2388226

About the Authors

Yifei Cao received his B.E. degree in surveying and mapping engineering from Beijing University of Civil Engineering and Architecture, Beijing, China, in 2019, where he is currently pursuing his master's degree with the Department of Remote Sensing. His current research interests include object detection and object tracking in remote sensing. (caoyifei@stu.bucea.edu.cn)

Jingguo Lv received his Ph.D. degree in charting and geographic information science from Beijing Normal University, in 2009. Since 2009, he has been teaching with Beijing University of Civil Engineering and Architecture, where he is an associate professor. His research interests include remote sensing information extraction, digital image processing, and visual tracking. He has published more than 40 related articles, published four academic monographs, and authorized seven invention patents. He has been awarded nine software copyrights and several technological awards. (lvjingguo@bucea.edu.cn)



Yingqi Bai received his B.E. degree in geographic information science from Beijing University of Civil Engineering and Architecture, Beijing, China, in 2018, where he is currently pursuing his master's degree with the Department of Remote Sensing. His current research interests include object tracking in remote sensing. (baiyingqi@stu.bucea.edu.cn)

Anqi Wu has been studying for a bachelor's degree at Beijing University of Civil Engineering and Architecture since 2017. Her major is photogrammetry and remote sensing. (wuanqi@stu.bucea.edu.cn)