



ARTICLE

Adaptive Object Tracking Discriminate Model for Multi-Camera Panorama Surveillance in Airport Apron

Dequan Guo¹, Qingshuai Yang², Yu-Dong Zhang³, Gexiang Zhang¹, Ming Zhu¹ and Jianying Yuan^{1,*}

¹School of Control Engineering, Chengdu University of Information Technology, Chengdu, 610225, China

²School of Computer Science, Sichuan University, Chengdu, 610041, China

³Department of Informatics, University of Leicester, Leicester, LE1 7RH, UK

*Corresponding Author: Jianying Yuan. Email: yuanjy@cuit.edu.cn

Received: 27 February 2021 Accepted: 28 June 2021

ABSTRACT

Autonomous intelligence plays a significant role in aviation security. Since most aviation accidents occur in the take-off and landing stage, accurate tracking of moving object in airport apron will be a vital approach to ensure the operation of the aircraft safely. In this study, an adaptive object tracking method based on a discriminant is proposed in multi-camera panorama surveillance of large-scale airport apron. Firstly, based on channels of color histogram, the pre-estimated object probability map is employed to reduce searching computation, and the optimization of the disturbance suppression options can make good resistance to similar areas around the object. Then the object score of probability map is obtained by the sliding window, and the candidate window with the highest probability map score is selected as the new object center. Thirdly, according to the new object location, the probability map is updated, the scale estimation function is adjusted to the size of real object. From qualitative and quantitative analysis, the comparison experiments are verified in representative video sequences, and our approach outperforms typical methods, such as distraction-aware online tracking, mean shift, variance ratio, and adaptive colour attributes.

KEYWORDS

Autonomous intelligence; discriminate model; probability map; scale adaptive tracking

1 Introduction

Autonomous intelligence has seen a wide range of applications in intelligent transportation. At the same time, aircraft transportation plays a significant role in the current rapid development of intelligent traffic. The airport apron security is becoming more and more important for increasing of air transport service. In general, it is difficult for a single camera to cover the large area of airport apron scene. The tarmac scene is relatively scattered, and a small field of view is not conducive to the observation of multiple objects in a large area. With the emergence and maturity of image mosaic technology, panoramic monitoring of large-scale scenes can be realized. Multi-camera panorama surveillance based on cameras are often used to monitor



important areas and objects in the airport apron. In this case, multiple cameras can monitor objects in different areas simultaneously. Object tracking is a hot topic in the field of intelligent surveillance [1]. The adaptive and automatic tracking needs higher robustness in unstructured and changing environments. This paper studies object tracking based on multi-perspective monitoring in airport apron, which will help to supervise the operation of aircraft in the take-off and landing stage.

For the airport apron object tracking, there are some difficult, for example, the object is with deformed appearance, lighting changes, appearance similarity, motion blur, occlusion change, out of sight, scale change, background similar or confusion [2–6]. In the framework of visual tracker, it is very important to extract object description features between various planes, which have a great impact on the accuracy and speed of tracking. Tracking algorithms based on machine learning and deep learning have also been applied. The convolutional layer is the main calculation, the capacity of convolutional neural network (CNN) needs to be controlled in related studies and optimization. When the object image is updated online, the object features will be fixed after the CNN offline training, thus avoiding the problem that the stochastic gradient descent and back propagation are almost impossible to carry out in real time. The research of high-performance visual tracker is not limited to testing in the database, but also needs to combine a variety of sensors to assist visual tracking, especially in the open environment.

Aiming to improve the plane tracking in airport apron, this paper presents a discriminant tracking method based on RGB color histogram. The method includes: Firstly, the probability map of the object is estimated in advance to reduce the computational effort of searching the object in the search ranges; Secondly, the object score of probability in candidate window is calculated by sliding windows in the current search area, and the candidate window with the highest score is selected as the new object position. Finally, the probability map is updated according to the new object position. The innovations of this study are listed as follows: 1). It optimizes the interference suppression term and with better resistance to similar areas around the target. 2). In view of the airport apron situation that the scale of the tracking plane may change in time, the scale estimation function is added into the algorithm, and the size can be adjusted to the object real size automatically.

The remainder of this paper is organized as follows: Section 2 briefly reviews the related literatures. The proposed object tracking algorithm will be discussed in Section 3. Some results on typical methods and real multi-camera panorama surveillance scenes are shown in Section 4. Conclusion and future work are given in Section 5.

2 Related Works

In recent years, many high-performance video tracking algorithms have emerged continuously, such as distraction-aware online tracking (DAT) [2], mean shift (MS) [3], variance ratio (VR) [4], adaptive colour attributes (ACA) [5]. Normalized cross-correlation (NCC) could perform simple pixel intensity matching. Recently, Briechele et al. [6] proposed a tracking method that did not need to update the object template. Snake model was a popular contour-based tracking algorithm [7]. The model generation method used the minimum reconstruction error to describe the object and search the most similar region of the model from the image [8–10]. Michael et al. [8] used the offline subspace model to represent the region of interest (ROI) of the object. The discriminant method adopted object detection to achieve tracking. In literature [11], the drift problem was dealt with by improving the online update discriminant feature algorithm and half online algorithm. Zdenek et al. [12] proposed a P-N learning algorithm based on the potential structure of positive

and negative samples to train the object tracking classifier. In [13], the feature variance ratio of maximized object to background was used to solve the performance degradation problem in the tracking process. Discriminant model was data association to some extent [14]. The method matched through color histogram without pixel space information, therefore, it could handle large object shape change. During tracking, the object was no longer updated. When the background and object color channels were similar, it was easy to cause the marked object region inaccurately, and the tracking object would be lost eventually.

To solve the problems of deformation, illumination change and rotation, two parallel correlation filters were proposed in [15] to track the appearance change and movement of the object respectively, and good tracking performance was obtained. In order to solve the regression problem of discriminant learning-related filters, the convolution regression framework was adopted to optimize the single-channel output convolution layer, and the new objective function was helpful to eliminate errors [16]. Combining the discriminant correlation filter with the vector convolutional network, the coarse-to-fine search strategy was adopted to solve the drift problem [17]. Fan et al. [18] used dynamic and discriminative information to improve the effect of recognition, it was helpful to represent variation in facial appearance and spatial domain. Li et al. [19] applied sub-block based image registration to remove the global motion in video frame. Gundogdu et al. [20] established a flexible network model suitable for special design based on the loss function of the network, and proposed a novel and efficient back propagation algorithm. In order to improve the generalization degree of the model, the appearance adaptive tracker was aimed at the object region by learning adaptability in the antagonistic network [21]. In [22], a new scale adaptive tracking method was proposed, which could translate and estimate the scale by learning independent discriminant correlation filter.

3 The Proposed Method

3.1 The Proposed Overall Framework

The important techniques in our proposed method include: object probability map estimation, object location update, object scale update, and probability map update. Our work is presented in red dotted line rectangles in Fig. 1. Based on the general tracking framework, we will get a new probability map of the object in the current frame, the disturbance calculation is helpful to suppress the disturbance information.

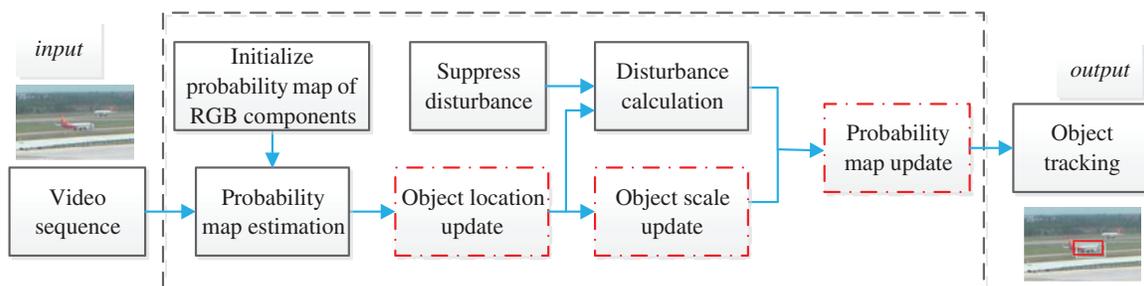


Figure 1: The overall framework of the proposed method

This study proposes a color multi-channel (red, green, blue, RGB) and discriminant method to track object in large airport apron. Based on color feature and intensity channel, combining the advantages of discriminant method and RGB histogram, which can effectively improve the

accuracy and adaptability of tracking. Color histogram can describe object features from more channels. In the object histogram, the image features can represent the internal connecting parts of the object. In the situation, the object is seen as a whole, with a certain continuity of color that can be distinguished from the background. Some colors of the object are similar to the background. When the object moves, the similar parts of the object move together, but the background parts do not move. That is, the multi-channel color of the background does not match the color of the object. When the object moves, its size and appearance will change at the same time, therefore, the method in this study is still with good adaptability. In addition, when the object is within the search range, the precomputed probability map and integral histogram facilitate real-time processing. The color histogram is used to estimate the object probability map, which can reduce the amount of searching computation. The region with the highest score in the sliding window is selected as the new position of the object. Different from other tracking methods, based on color histogram, the new designed interference suppression term can effectively reduce the influence of the similar area around the object. Therefore, the object can be tracked adaptively.

3.2 Pre-Estimation of the Object Probability Map

The discriminative-based tracking method regards object tracking to be the binary classification problem between object and background. The method uses one frame to sample the object position to distinguish the local object region in the background of the current frame. The accuracy and stability of tracking depend on the separability of object and background. A good classifier is of great significance for discriminating tracking algorithm. A widely used Bayesian classifier is used in this study. The blue identification box is the tracking object, which represents the object, here, marked as “O”. The red rectangle is the outer rectangle of the target, denoted by R. The green box marks the area around the target, containing part of the around, here, denoted as A. The yellow box indicates the disturbance area, and the distance from the object, here is marked as “D”. These marked boxes are shown in Fig. 2.

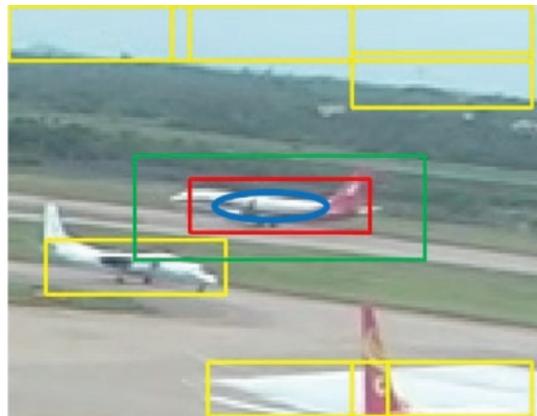


Figure 2: Region marked: Blue—Object, Red—Rectangle, Green—Around, Yellow—Disturbance

To identify object pixels $x \in O$ from around background pixels, based on Bayes classifier on the image I , the colour histogram is used. Here, $H_{\Omega}^I(b)$ define the b -th bin of the non-normalized histogram H , in the local region $\Omega \in I$. Furthermore, b_x is the bin b of the color components of $I(x)$. In Fig. 2, “O” and “R” denote the object bounding box and rectangle around the object,

through Bayes rule, the object likelihood at location x is obtained:

$$P(x \in O | O, R, b_x) \approx \frac{P(b_x | x \in O)P(x \in O)}{\sum_{\Omega \in \{O, R\}} P(b_x | x \in \Omega)P(x \in \Omega)} \quad (1)$$

Additionally, we extend the representation, the probability of the pixel x on the frame I , which belongs to the object O can be calculated by:

$$P(x \in O | O, R, A, rgb_x) = \frac{P(rgb_x | x \in O)P(x \in O)}{\sum_{U \in \{O, R, A\}} P(rgb_x | x \in U)P(x \in U)} \approx \frac{\sum_{T \in \{O, R\}} P(rgb_x | x \in T)P(x \in T)}{\sum_{U \in \{O, R, A\}} P(rgb_x | x \in U)P(x \in U)} \quad (2)$$

where rgb_x represents the RGB pixel value at x , then the RGB histogram is estimated by the likelihood:

$$\begin{cases} P(rgb_x | x \in O) \approx H_O^I(rgb_x)/|O| \\ P(rgb_x | x \in R) \approx H_R^I(rgb_x)/|R| \\ P(rgb_x | x \in A) \approx H_A^I(rgb_x)/|A| \end{cases} \quad (3)$$

where $H_U^I(rgb_x)$ represents number of RGB colour, with x in U , $|\bullet|$ is the cardinality of the set, the prior probability $P(x \in O) \approx |R|/(|O| + |R| + |A|)$, Put Eq. (3) in Eq. (2), then obtain:

$$P(x \in O | O, R, A, rgb_x) = \begin{cases} \frac{H_O^I(rgb_x)}{H_O^I(rgb_x) + H_R^I(rgb_x) + H_A^I(rgb_x)}, & I(x) \in I(O \cup R \cup A) \\ 0.5, & others \end{cases} \quad (4)$$

For the un-appeared RGB colour vector, the probability that the object region will appear in the next frame with 50 percent, without loss of generality, the value is set to be 0.5. When a similar area appears around the object, it is possible to misjudge the similar area as part of the object or even the object. To solve this problem, the similar area around the object is assumed the current similar area to be Eq. (2). The object probability based on the similar region is:

$$P(x \in O | O, R, D, rgb_x) = \begin{cases} \frac{H_O^I(rgb_x)}{H_O^I(rgb_x) + H_R^I(rgb_x) + H_D^I(rgb_x)}, & I(x) \in I(O \cup R \cup D) \\ 0.5, & others \end{cases} \quad (5)$$

Then obtain $P(x \in O | O, R, A, rgb_x)$, and the final object possibility map is listed as follows:

$$P(x \in O | rgb_x) = \begin{cases} \max(PA_x, PD_x), & PA_x > \delta_A \ \& \ PD_x > \delta_D \\ PD_x, & PA_x > \delta_A \ \& \ PD_x < \delta_D \\ (PA_x + PD_x)/2, & others \end{cases} \quad (6)$$

where $PA_x = P(x \in O | O, R, A, rgb_x)$, $PD_x = P(x \in O | O, R, D, rgb_x)$. δ_A denotes the probability score in the background area around the target, δ_D is the current probability score in a similar area. In Fig. 3a, only the object (the region enclosed by red rectangle) is considered, and the probability map can be calculated via Eq. (4), which is shown in the right column of Fig. 3a. The probability of the disturbance item (the area enclosed by yellow rectangular) is higher, which may influence the tracking results. The disturbance item is added to inhibit probability map induced by Eq. (6), which is shown in the right column of Fig. 3b.

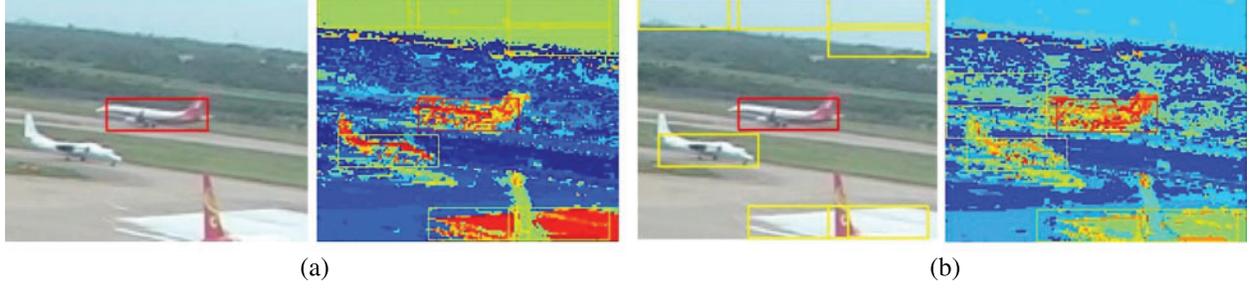


Figure 3: Object and background possibility map (a) The probability map is not imported (b) The probability map is imported

After the disturbance suppression is added, the value of the probability map disturbance item is suppressed obviously, and the disturbance with the real object is also reduced in Fig. 3. Due to the influence of the object in continuous movement and external conditions (light, fog, haze, etc.), the appearance of the object may change constantly, therefore, it is necessary to update the probability map constantly. Eq. (6) can calculate the probability $P(x \in O | rgb_x)$ of the current frame with the weighted sum $P_{t-1}(x \in O | rgb_x)$ of the previous frame,

$$P_t(x \in O | rgb_x) = \alpha P(x \in O | rgb_x) + (1 - \alpha) P_{t-1}(x \in O | rgb_x) \quad (7)$$

The probability map Eq. (7), which takes the attribute value of the N frame before considering the object. Henceforth this algorithm is with a strong robustness to adverse situation that the local object is temporarily blocked during the process of tracking the object.

3.3 Location Updating

While the object is moving continuously, the place O_t of the object at t will be not at the next frame $t-1$ of position O_{t-1} . Define a search area and a sliding window, they start from the left upper corner to right, and from top to bottom to calculate the current sliding window. Then a new position of the object score is obtained, the work can be shown in Fig. 4. The size of the search area depends on the object size in the previous frame. The horizontal step and the vertical step are determined by the overlap threshold, and the sliding window is 3/4 of the object size in the previous frame.

Define the current sliding window score of the calculation formula as follows:

$$S(O_{t,i}) = \left(\sum_{x \in \{O_{t,i}, R_{t,i}\}} P_{t-1}(x \in O | rgb_x) \right) \left(\sum_{x \in \{O_{t,i}, R_{t,i}\}} \|x - c_{t-1}\|^2 \right) \quad (8)$$

where, c_{t-1} is the center coordination of $t-1$ in the object.

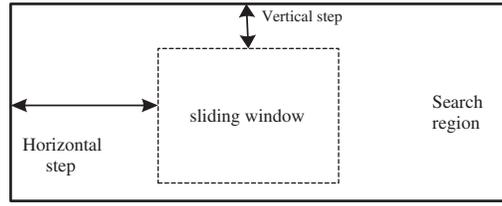


Figure 4: Searching window schematic

When the real object is in a similar area around the object, it interferes with the object tracking. Here, a disturbance term is employed to calculate the probability in the previous part. With the object changing constantly, the disturbance term will change accordingly, which makes the current disturbance area reset. Based on O_t^* , a new disturbance region is defined:

$$S(O_{t,i}) \geq \lambda_S S(O_t^*) \quad (9)$$

where λ_S is the adjust parameter. The highest scoring sliding window is the new object position:

$$O_t^* = \max_{O_{t,i}} (S(O_{t,i})) \quad (10)$$

3.4 Scale Updating

The object size may be changed while the object is moving, the size is estimated in the current frame. Here, a scale update strategy is designed. First, locate the object in the new frame, and then estimate the size. Based on $P(x \in O | rgb_x)$, ROI is segmented by a threshold. The estimation is influenced by the complexity of the object background and mildly rapid changes. Therefore, dynamic variation of the threshold is needed. The probability is applied to calculate the cumulative histogram of region O and D , respectively:

$$C_O^P(b) = \sum_{i=1}^{rgb} H_O^P(i) / |O| \quad (11)$$

$$C_D^P(b) = \sum_{i=1}^{rgb} H_D^P(i) / |D| \quad (12)$$

Based on Eqs. (11) and (12), threshold T can be calculated in Eq. (13), and defined as follows:

$$T = \operatorname{argmin}(\operatorname{abs}(C_O^P(b) - (1 - C_D^P(b))) + (1 - \sum_i (C_O^P(i+1) - C_O^P(i))) + (C_O^P(b) < 1 - C_D^P(b))) \quad (13)$$

where, T is a vector, and the smallest element of T is the threshold. To adjust the current object area O_t^* , define a range object area about $3/4$ of the size, and then calculate the sum S_p of the probability values of each row and column from four directions. If $S_p > \lambda T$, the current row or column is considered as part of the object area, otherwise, it is determined to be the background. In the algorithm, p is the present image, N is the number of sequence.

The main process of the proposed method is described below:

Algorithm 1: The proposed tracking method

Input: Image sequence**Output:** Object tracking image sequence1: **initialize** location **while** $p \leq N$ **do**2: **Represent the discriminant model:** get color range for pixel, and get probability map $P(x \in O | O, R, A, rgb_x)$ by Eq. (2); obtain object region and distractor-region $P(x \in O | rgb_x)$ by Eq. (6);3: **Set new object location:** calculate current sliding window score $S(O_{t,i})$ by Eq. (8)

adopt iterative non-maximum suppression strategy by Eq. (9)

 obtain new object position $O_t^* = \max_{O_{t,i}}(S(O_{t,i}))$ by Eq. (10)4: **Estimate scale:** calculate the cumulative histogram $C_O^P(b)$ by Eq. (11), and $C_D^P(b)$ by Eq. (12) obtain new adaptive segmentation threshold T by Eq. (13)

calculate the scale size and update the size of the object.

5: **end while**6: **until** end of sequences.

4 Experimental Results and Analysis

The experiments are performed on desktop computer with Intel i7-7700 CPU (2.80 G) and 16 G memory. The software environment is 64-bit Windows operating system, Visual Studio 2008 integrated development environment, OpenCV 2.4.6 library. The initial size of object “O” is set manually. The boundary/size of R is about 5/4 of O, meanwhile, the boundary/size of A is about 5/4 of R. The region of D is selected by random in the image. The template for T is also used the initial object by manually or system setting. Then the T of the next frame is calculated by the previous frame in the tracking process. The whole tracking process is real-time, it can be up to 25 frames per second for 4000×1080 image through parallel computing 1080 GPU machine. To validate the effectiveness of the proposed method, experiments are performed from our real aircraft in the airport apron, at the same time, compared with the datasets and benchmark [23]: http://cvlab.hanyang.ac.kr/tracker_benchmark/benchmark_v10.html or VOT Dataset is <https://votchallenge.net/vot2015/dataset.html>. The dataset comprises 60 short sequences showing various objects in representative set of challenging backgrounds. The attributes of aircraft tracking performance are variation of illumination and occlusion, deformation and scale variation mainly. These attributes of the aircraft are often appeared in airport apron.

Due to the particularity of the airport apron, cameras are not allowed to be set up in the middle, therefore, The cameras are fixed at points that do not affect the normal operation of the aircraft. After the aircraft enters the airport apron, it is inevitable to be blocked by the corridor bridge and the lighthouse pole, resulting in illumination and occlusion. At the same time, when the cameras are shooting moving aircraft from the fixed points, the angles are constantly changing, the variation of scale and deformation of the aircraft is essential. Therefore, the real airport aircraft experiments are mainly tested from these two aspects. Then, in order to compare with other methods, experiments are carried out on illumination, occlusion, scale and deformation. The qualitative and quantitative analysis are used to prove the effectiveness of the proposed method.

4.1 Adaptive Deformation and Scale Variation Tests

To verify the adaptability of our algorithm with the object size, the object is tracked in a long-term image sequence. Fig. 5 shows the process of a civil aircraft moving from an airport apron runway to a taxiway through the terminal building, in which the object size changes greatly in the sequence. Frame #0001 is the object's initial position, and then the object moves from right to left, near to far, and the tracking rectangle gets smaller and smaller, in frame #0550. After a certain time, in frame #1719, the plane object moves from the lower left corner to the taxiway, and the shape and size of the object decrease. Currently, the object on the taxiway begins to move, and the tracking rectangle changes accordingly. In Fig. 5, the object goes through frame #1719, #2580, #4099, and #4837, the marked rectangle grows as the object is near and far.

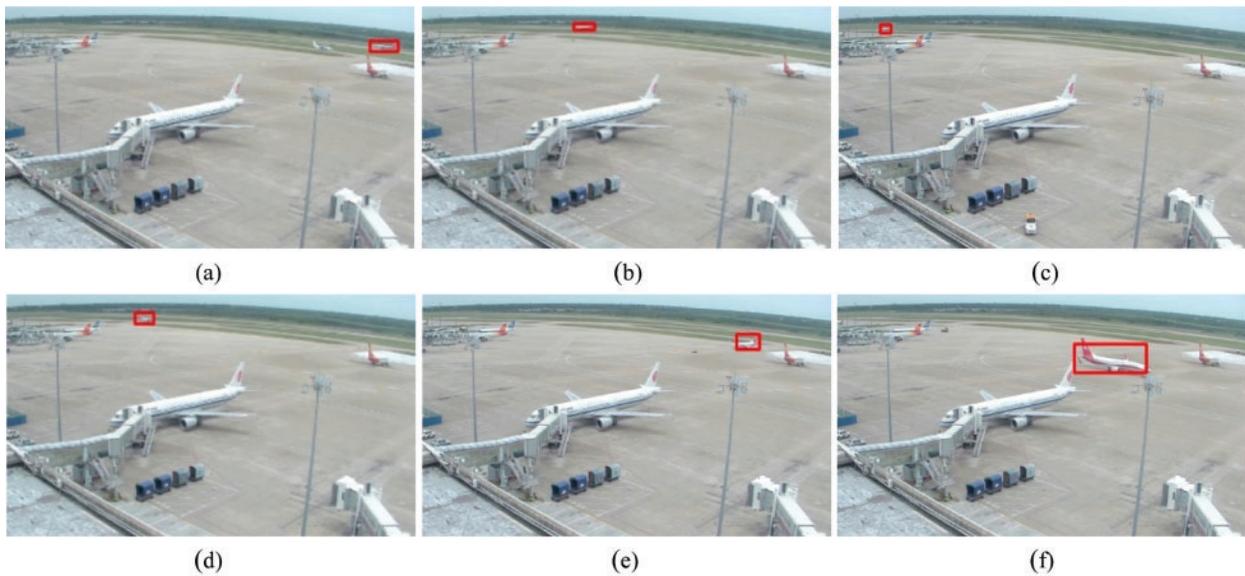


Figure 5: Adaptive scale size process in a certain civil aircraft (a) #0001 (b) #0550 (c) #1719 (d) #2580 (e) #4099 (f) #4837

Fig. 5 demonstrates that the proposed tracking method can accomplish the estimation of the object size exactly during object size variation. Our algorithm maintains good tracking performance, when the object plane goes through the middle of the light bracket, and other similar shape of the aircraft. The object can be detected until the human eye can barely see the target. The good performance of proposed tracking is verified in the actual scene detection.

4.2 Robustness Variation of Illumination and Occlusion Tests

To test the robustness of the proposed method, the splicing parameters are specially adjusted to appear some splicing joints in the panoramic image. There are some hinders, such as deformation, fracture, or partial loss of the object at the splicing joints, and the illumination changes greatly. Fig. 6 shows the tracking results in panoramic videos, which are combined with the four-way cameras. Due to the different orientations of the spliced camera, there is inevitably different illumination in the panoramic videos. The variation of illumination and occlusion is in Fig. 6.

There are obvious slight changes at the splicing joint in the right side of the panorama in Fig. 6. From the results of frame #0010, #0038 and #0070, it can be found that the proposed

method achieves stable tracking of the splicing joint with obvious light changes across the target. There is obvious fracture in the left side of panorama image. From frame #0163 to #0238, the object is moving through the physical seam, it appears to be apparent fracture and part of the loss. And then our algorithm still tracks the object successfully. In all, the proposed method is strongly resisted to deformation or fracture.

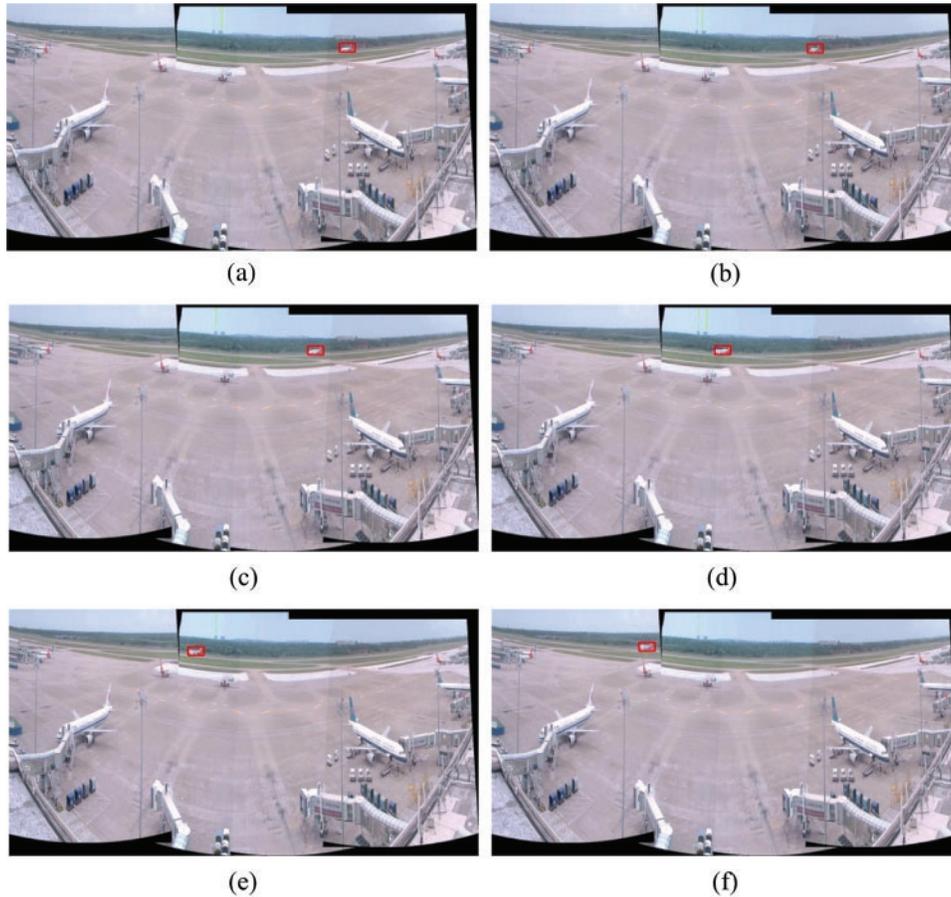


Figure 6: Tracking results under the circumstance of panoramic situation (a) #0010 (b) #0038 (c) #0070 (d) #0163 (e) #0211 (f) #0238

4.3 Qualitative Comparison Analysis

To test our method and others fairly and comprehensively, a small object is selected to test. In Fig. 7, an airport apron transport vehicle goes through the parking space, and with the period of blocking by the plane. It shows the tracking results in the Seq3 image sequence (511 frames), #017, #095, #0191, #0293, #0371, #0429. We obtain the comparison of the tracking results of ACA, DAT, VR, MS, and our method in sequence Seq3. From the two-group airport apron image test sequences, the other four tracking methods are with some extent degrees of deviation or affected by similar area in tracking, even the object is lost in VR and MS. In the whole, the proposed approach outperforms other methods.

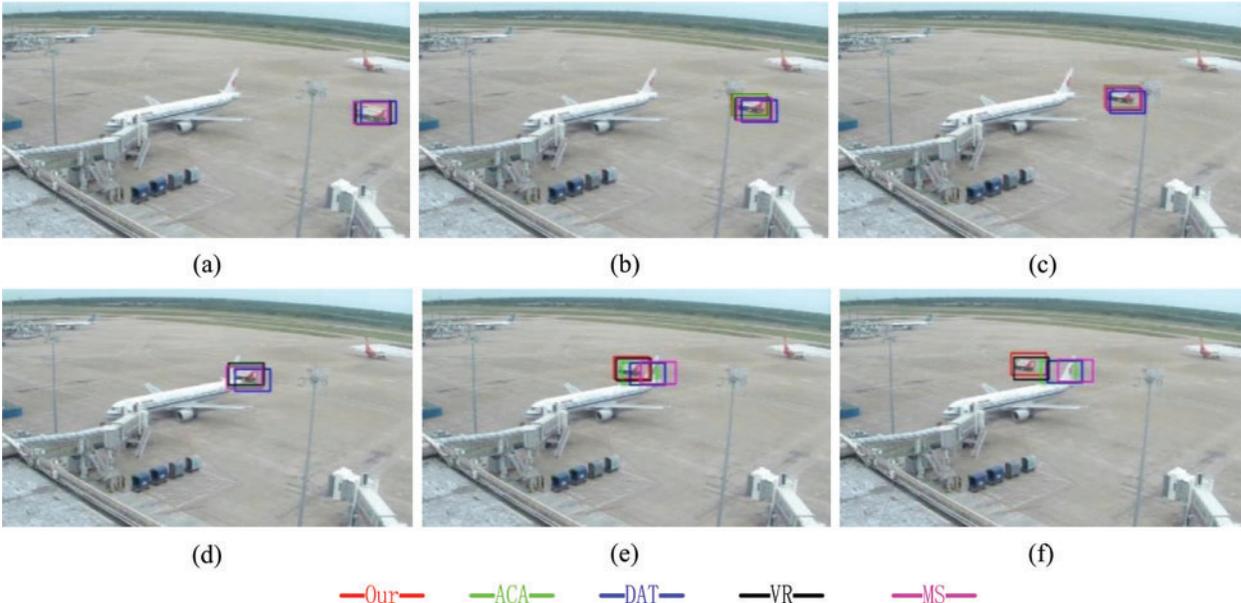


Figure 7: Comparison of the tracking results of ACA, DAT, VR, MS, and our method in near sequence (a) #0017 (b) #0095 (c) #0191 (d) #0293 (e) #0371 (f) #0429

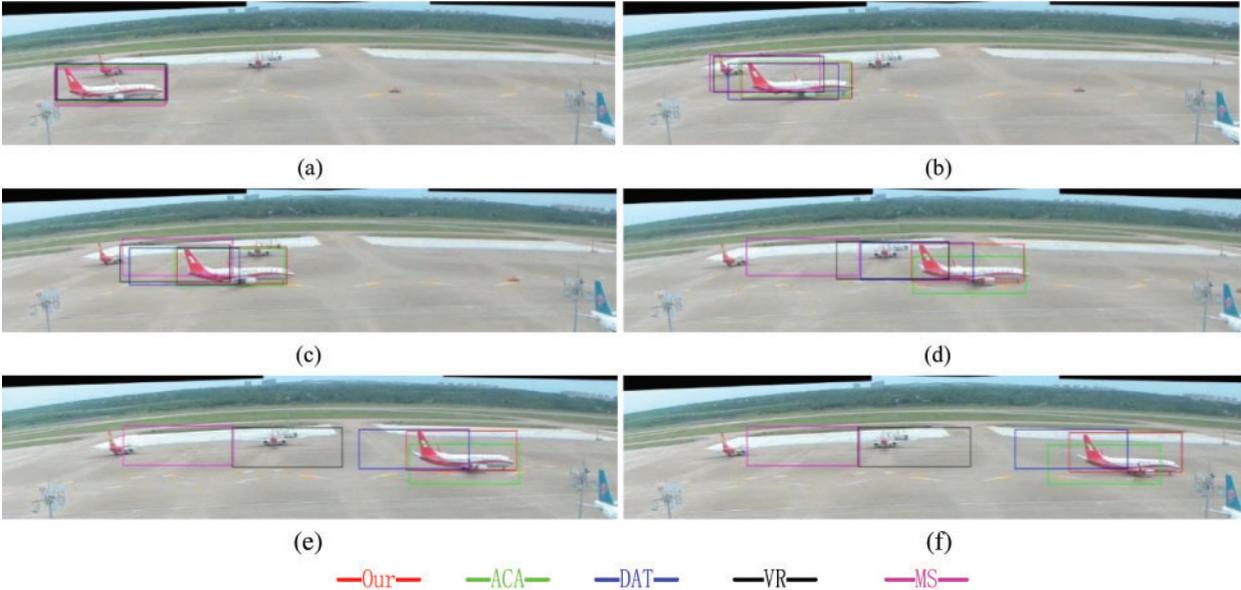


Figure 8: Comparison of tracking results of ACA, DAT, VR, MS, and our method in far sequence (a) #0171 (b) #0280 (c) #0368 (d) #0523 (e) #0678 (f) #0752

In Fig. 8, the tracked object is a passenger jet in the panoramic image. Fig. 8 shows the tracking results of the Seq2 image sequence (856 frames), ##0171, ##0280, #0368, #0523, #0678 and #0752. The goal plane moves from left to right, and goes through other planes, in the middle process, the goal plane is partly blocked by other planes in the parking apron, the proposed

method can mark the object plane much better than other algorithms. Especially, at the end of frame #0752, the rectangle center is almost the plane object center.

4.4 Quantitative Comparison Evaluation

1) Single attribute accuracy tests

In order to compare the performance of these five tracking algorithms objectively, the real position of the tracking object in the three test sequences are marked. The tracking accuracy and success rate [23] are used to evaluate the performance of the five algorithms. Accuracy represents the percentage of the estimated position within each threshold distance from the ground truth center. Accuracy is the Euclidean distance between the center positions by pixels. The evaluation indexes in [24] is tested. Distance accuracy refers to the pixel distance between the truth value and the center of the tracking result, which means that the center positioning error probability is less than a certain value. The overlap rate is tracked by testing whether it is greater than a certain value [25,26]. Define tracking score:

$$score = \frac{area(B_t \cap G_t)}{area(B_t \cup G_t)} \quad (14)$$

where B_t is the tracking rectangle, G_t defines the object marked region box, \cap represents overlap (intersection) of two regions, \cup represents union set of the two regions, $area(\cdot)$ is the searching area. The score of tracking is from 0 to 1. Tab. 1 shows the comparison results of accuracy and success rate in the five tracking methods from image sequence 1, 2, 3 and 4. In the experiment, it is found that when the accuracy is less than 6 pixels, the positioning accuracy in the proposed method is basically the same as that of the ACA method, and slightly higher than that of the other three tracking methods. When the accuracy is less than 15 pixels, the tracking accuracy of the proposed method is higher than that of the other four methods. Here, S₁, S₂, S₃, S₄ are the abbreviation of image sequence 1, 2, 3, and 4.

Table 1: Comparison tests of five methods in image sequence 1, 2, 3 and 4

Method	Accuracy				Success rate			
	S1	S2	S3	S4	S1	S2	S3	S4
ACA [5]	0.671	0.666	0.666	0.713	0.565	0.563	0.560	0.603
DAT [2]	0.583	0.584	0.546	0.609	0.463	0.457	0.560	0.485
VR [4]	0.406	0.496	0.489	0.486	0.311	0.384	0.371	0.370
MS [3]	0.270	0.323	0.347	0.355	0.180	0.221	0.242	0.2964
Our	0.827	0.799	0.820	0.822	0.647	0.617	0.834	0.648

2) Overall attributes accuracy tests

In order to test the actual performance of the proposed tracking algorithm, the proposed tracking algorithm is compared with the current tracking algorithm (MS, ACA, VR, DAT) on multiple data sets in Fig. 9. The test sequences data sets include comprehensive image sequences of airport apron, railway station, stadium, road, park, indoor scene and so on. Objects are various, including planes, shuttles, cars and pedestrians, soccer balls, and more. From the results, our method maintains better performance than other methods. In the precision comparison test, the horizontal axis is the positioning error value, and the vertical axis is the accuracy [27–29].

In the success rate test, the horizontal axis represents the overlap threshold and the vertical axis represents OR. The area under curve (AUC) is used to measure the tracking accuracy and success rate. The temporal robustness assessment is generated from the mean of all tests in the follow-up results, and the other score is the spatial robustness assessment. Overall, the method in this paper maintains good performance.

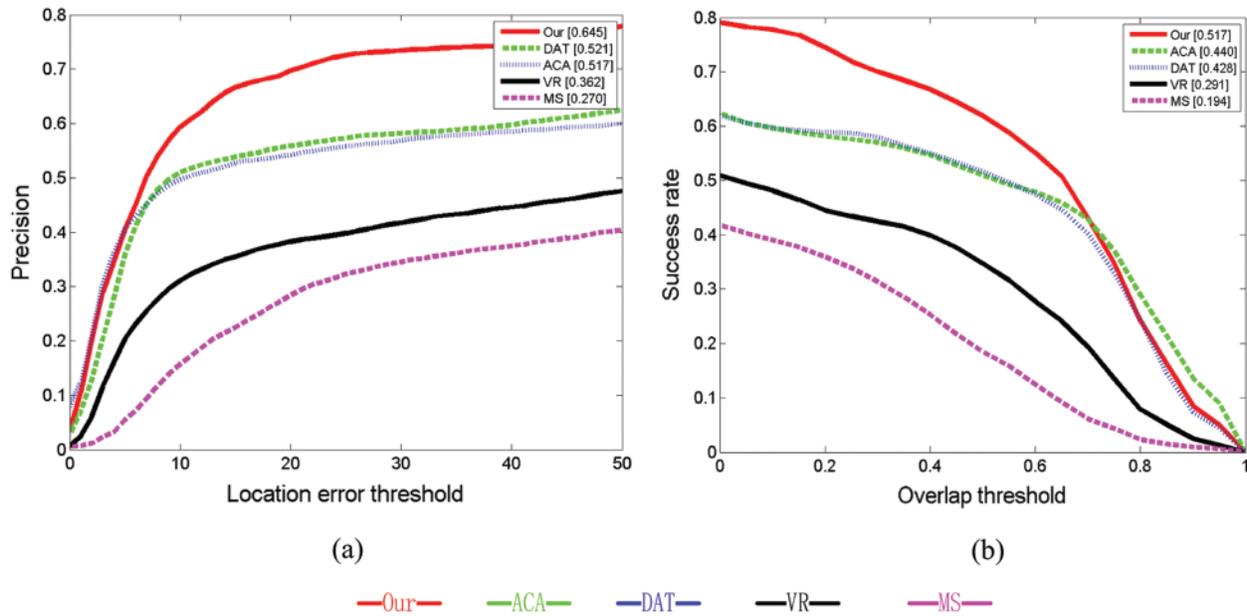


Figure 9: The tracking performance of ACA, DAT, VR, MS, and ours in comprehensive image sequences (a) precision plot (b) success plots ratio

5 Conclusion and Future Work

This study improves the object tracking problem of multi-camera mosaic algorithm and analyzes the limitations of the existing algorithm in airport apron object tracking. An improved discriminant object tracking method based on color histogram is proposed for the special environment of airport apron. Firstly, estimation of the object probability graph in advance helps to reduce the search calculation. Secondly, the sliding window of the current search area is calculated as the object score, and the candidate window with the highest score is selected as the new object position. In addition, the probability map is updated according to the new object position. In this study, the disturbance term is optimized to control the similar region around the target. For large objects in airport apron scene, the scale estimation function can be added to the algorithm when the object scale changes greatly. Finally, to verify the effectiveness and stability of the method, the qualitative performance and quantitative comparison experiments were carried out on several test image sequences. The experimental results show that the proposed method is superior to other general methods. In the future research, the object information in the field of view of multiple cameras will be employed, and the complementarity of feature information from multiple perspectives [30] and intelligent algorithm [31,32] will be fully considered, to provide prior content for the object in the panorama and improve the tracking performance.

Funding Statement: This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61806028, 61672437 and 61702428, Sichuan Science and Technology Program under Grant Nos. 2018GZ0245, 21ZDYF2484, 18ZDYF3269, 2021YFN0104, 2021YFN0104, 21GJHZ0061, 21ZDYF3629, 2021YFG0295, 2021YFG0133, 21ZDYF2907, 21ZDYF0418, 21YYJC1827, 21ZDYF3537, 21ZDYF3598, 2019YJ0356, and the Chinese Scholarship Council under Grant Nos. 202008510036, 201908515022.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Wu, Y., Liu, C., Lan, S., Yang, M. (2015). Real-time 3D road scene based on virtual-real fusion method. *IEEE Sensors Journal*, 15(2), 750–756. DOI 10.1109/JSEN.2014.2354331.
2. Horst, P., Mauthner, T., Bischof, H. (2015). In defense of colour-based model-free tracking. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2113–2120. Boston, MA, USA. DOI 10.1109/CVPR.2015.7298823.
3. Dorin, C., Ramesh, V., Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 142–149. Hilton Head, SC, USA. DOI 10.1109/CVPR.2000.854761.
4. Robert, C., Liu, Y., Leordeanu, M. (2005). Online selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10), 1631–1643. DOI 10.1109/TPAMI.2005.205.
5. Danelljan, M., Khan, F., Felsberg, M., Weijer, J. (2014). Adaptive colour attributes for real-time visual tracking. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1090–1097. Columbus, USA.
6. Briechle, K., Hanebeck, U. (2001). Template matching using fast normalized cross correlation. *Proceedings of SPIE, AeroSense Symposium*, vol. 4387, pp. 95–102. Florida.
7. Gong, X., Zheng, Y. (2013). Ultrasound ventricular contour extraction based on an adaptive GVF snake model. *IEEE Seventh International Conference on Image and Graphics*, pp. 231–235. Qingdao, China.
8. Michael, B., Jepson, A. (1998). Eigen tracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Conference on Computer Vision*, 26(1), 63–84. DOI 10.1023/A:1007939232436.
9. Ross, D., Lim, J., Lin, R., Yang, M. (2008). Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1), 125–141. DOI 10.1007/s11263-007-0075-7.
10. Kwon, J., Lee, K. (2010). Visual tracking decomposition. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1269–1276. San Francisco, California, USA.
11. Helmut, G. (2006). Real-time tracking via on-line boosting. *Proceedings of British Machine Vision Conference*, pp. 47–56. Edinburgh.
12. Zdenek, K., Matas, J., Mikolajczyk, K. (2010). P-N learning: Bootstrapping binary classifiers by structural constraints. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, vol. 238, pp. 49–56. San Francisco, California, USA.
13. Smeulders, A., Chu, D., Cucchiara, R., Calderara, S., Dehghan, A. et al. (2014). Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7), 1442–1468. DOI 10.1109/TPAMI.2013.230.
14. Wang, H., Nguang, S. (2016). Multi-target video tracking based on improved data association and mixed kalman filtering. *IEEE Sensors Journal*, 16(21), 7693–7704. DOI 10.1109/JSEN.2016.2603975.
15. Yang, Y., Zhang, Y., Li, D., Wang, Z. (2019). Parallel correlation filters for real-time visual tracking. *Sensors*, 19, 1–22. DOI 10.3390/s19102362.
16. Chen, K., Tao, W. (2018). Convolutional regression for visual tracking. *IEEE Transactions on Image Processing*, 27(7), 3611–3620. DOI 10.1109/TIP.2018.2819362.

17. Liu, Y., Sui, X., Kuang, X., Liu, C., Gu, G. et al. (2019). Object tracking based on vector convolutional network and discriminant correlation filters. *Sensors (Basel)*, 19(8), 1–14. DOI 10.3390/s19081818.
18. Fan, X., Yang, X., Ye, Q., Yang, Y. (2018). A discriminative dynamic framework for facial expression recognition in video sequences. *Journal of Visual Communication and Image Representation*, 56, 182–187. DOI 10.1016/j.jvcir.2018.09.011.
19. Li, F., Yang, Y., (2015). Real-time moving targets detection in dynamic scenes. *Computer Modeling in Engineering & Sciences*, 107(2), 103–124. DOI 10.3970/cmcs.2015.107.103.
20. Gundogdu, E., Alatan, A. (2018). Good features to correlate for visual tracking. *IEEE Transactions on Image Processing*, 27(5), 2526–2540. DOI 10.1109/TIP.2018.2806280.
21. Javanmardi, M., Qi, X. (2020). Appearance variation adaptation tracker using adversarial network. *Neural Networks*, 129, 334–343. DOI 10.1016/j.neunet.2020.06.011.
22. Danelljan, M., Hager, G., Khan, F., Felsberg, M. (2017). Discriminative scale space tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(8), 1561–1575. DOI 10.1109/TPAMI.2016.2609928.
23. Wu, Y., Lim, J., Yang, M. (2015). Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9), 1834–1848. DOI 10.1109/TPAMI.2014.2388226.
24. Henriques, J., Rui, C., Martins, P., Batista, J. (2015). High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3), 583–596. DOI 10.1109/TPAMI.2014.2345390.
25. Zhang, T., Liu, S., Ahuja, N., Yang, M., Ghanem, B. et al. (2015). Robust visual tracking via consistent low-rank sparse learning. *International Journal of Computer Vision*, 111(2), 171–190. DOI 10.1007/s11263-014-0738-0.
26. Liu, F., Zhou, T., Fu, K., Yang, J. (2016). Robust visual tracking via constrained correlation filter coding. *Pattern Recognition Letters*, 84, 163–169. DOI 10.1016/j.patrec.2016.09.009.
27. Grabner, H., Leistner, C., Bischof, H. (2008). Semi-supervised on-line boosting for robust tracking. *Proceedings European Conference on Computer Vision*, pp. 234–247. Springer-Verlag, Marseille, France.
28. Dalal, N., Triggs, B. (2005). Histograms of oriented gradients for human detection. *Proceedings IEEE Conference Computer Vision and Pattern Recognition*, vol. 20–26, pp. 1–8. San Diego, CA, USA.
29. Nam, H., Han, B. (2016). Learning multi-domain convolutional neural networks for visual tracking. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4293–4302. Las Vegas, NV.
30. Zhang, J., Zhang, Y., Wei, Q. (2018). Robust target tracking method based on multi-view features fusion. *Journal of Computer-Aided Design & Computer Graphics*, 30(11), 2108–2124. DOI 10.3724/SP.J.1089.2018.17037.
31. Xu, S., Jiang, Y. (2020). Research on trajectory tracking method of redundant manipulator based on PSO algorithm optimization. *Computer Modeling in Engineering & Sciences*, 125(1), 401–415. DOI 10.32604/cmcs.2020.09608.
32. Yuan, J., Zhang, G., Li, F., Liu, J., Xu, L. et al. (2020). Independent moving object detection based on a vehicle mounted binocular camera. *IEEE Sensors Journal*, 21(10), 11522–11531. DOI 10.1109/JSEN.2020.3025613.