Jestr

Journal of Engineering Science and Technology Review 13 (5) (2020) 33 - 41

JOURNAL OF
Engineering Science and Technology Review

Research Article

www.jestr.org

Long-term Object Tracking Based on Improved Continuously Adaptive Mean Shift Algorithm

Jinping Sun[1,2], Enjie Ding[1*], Dan Li[2], Adeel Akram[2] and Matthew Keith Kerns[3]

[1]*School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221008, China*
[2]*School of Information Engineering (School of Big Data), Xuzhou University of Technology, Xuzhou 221008, China*
[3]*G2K Equipment and Services LLC, Mckinney, TX 75071, United States*

Received 6 July 2020; Accepted 29 September 2020

Abstract

Long-term object tracking encounters complex scene changes, such as deformation, short-term departure from sight, occlusion, and lighting changes, resulting in complex and unstable tracking. To improve the accuracy and success rate of long-term object tracking in complex scenes, an improved continuously adaptive mean shift (CAMShift) algorithm was proposed. The joint probability density distribution of the target model was obtained by using the Bhattacharrya coefficient to calculate the contribution of the color features and texture features. Combining with the fused target model and Kalman filter, the target position was obtained by implementing CAMShift algorithm. Finally, a template pool was designed to store high-confidence tracking results. The target template was updated online by retrieving the initial frame from the template pool to recover re-detection after tracking drift or failure. The accuracy of the proposed algorithm was verified by simulation analysis. Results show that the distance precision and success rate of the proposed algorithm are 0.9 and 0.83, respectively. The proposed algorithm effectively solves long-term target tracking problems affected by complex scenes, such as occlusion, similar colors, and deformation. This study provides references for the automatic detection of traffic incidents in the intelligent traffic monitoring system.

Keywords: Long-term tracking, Color features, Texture features, Kalman filter, Template pool, Re-detection

1. Introduction

The object tracking technology [1] is an important research direction in the current computer vision field and is a complex and challenging technology that integrates pattern recognition, image processing, and computer applications. Given the size and position in the initial frame, the task of object tracking is to predict the size and position of the tracking object in subsequent framese. At present, object tracking algorithms are mainly based on support vector machines, correlation filters, deep learning, and continuously adaptive mean shift (CAMShift). In recent years, mean shift algorithm has been widely used in many algorithms because of its simple calculation and high performance. When the scale of the tracking object changes, the mean shift algorithm [2] cannot update the target model, which affects the accuracy of object tracking.

To solve the problem that the mean shift algorithm cannot update the target model in real-time, Bradski [3] proposed a continuously adaptive mean shift algorithm based on color probability distribution. The tracking effect for solid-color objects under a black and white background is good, but tracking will fail if the background color is close to the target. Scholars have improved the CAMShift algorithm in feature extraction and position prediction. On the basis of color probability distribution, texture features [4] with distinguishing ability were introduced to make up for the shortcomings of traditional CAMShift algorithm. Target histogram template [5] was generated based on the combination of hue component H and saturation component S, and the adaptability to complex environments was enhanced using complete target color information features. The Kalman filter was employed to predict the position of a specific target in the next frame, but the accuracy of the target position was affected by the interference of similar colors [6]. How to extract the effective features of the tracking target and improve the accuracy and robustness of the tracking is an urgent problem to be solved.

In this study, the robust features are extracted and the adaptive fusion methods is used to fuse these features. The CAMShift algorithm is employed to converge the area to be tracked, and the Kalman filter is employed to predict the position of a specific target in the next frame, reducing the algorithm time. A re-detection mechanism is proposed to solve the problem of tracking loss caused by occlusion and short-term out of sight.

2. State of the art

The object tracking in complex scenes, such as similar target colors, occlusion, deformation, and fast motion [7–8], is the core issue in the field of intelligent video monitoring. Wang [9] proposed a continuous adaptive mean shift tracking algorithm for the background suppression histogram model. The method improved the accuracy and stability of tracking by suppressing the hue belonging to the background in the original color model, but the method could not be used for target tracking. The target model was updated, and the target was easily lost when the size changed. Howard [10] extended the CAMShift algorithm, adding a fast-moving target state prediction algorithm with adaptive kernel

*E-mail address: sjp614@163.com
ISSN: 1791-2377 © 2020 School of Science, IHU. All rights reserved.
doi:10.25103/jestr. 135.05

bandwidth and state estimation and reducing the average displacement iteration. However, the tracking was easy to lose when the target was occluded. Hayat [11] used Canny edge detection and inter-frame difference method to extract moving targets to initialize the CAMShift algorithm. When similar color interference was observed in the background, CAMShift algorithm combined the Kalman filter to achieve accurate target tracking. However, tracking would be unsuccessful when the target moved fast. Yu [12] combined Kalman filtering and LBP texture to improve the CAMShift algorithm. The algorithm tracked texture features and color features as target features, showing a certain degree of accuracy and robustness. However, the algorithm lacked a re-detection mechanism. When the target was out of sight for a short time and then reappeared, the target could not be tracked correctly. Razzaq [13] used the difference between frames for target detection, used a set of morphological operations to extract features, and then applied Kalman filtering for tracking. When the algorithm was fast for the target, tracking would fail. Chu [14] proposed an improved CAMShift target tracking algorithm, which combined color and texture histograms, and used particle filtering to estimate the state of moving targets to solve the problem of similar colors. However, when the target was occluded, the estimation result of the particle filter deviated from the actual one. Zhang [15] used the Kalman filter algorithm to predict the state of the target, judged whether the target was occluded, and marked the target to predict the target that was occluded in the later stage. In the long-term target tracking process, given the changeable tracking environment, the target had different deformations, severe occlusion, and other problems, which could cause tracking failure. How to restore the target tracking function quickly was the basis for long-term target tracking. Zhang [16] established the descriptors of rotation and scale normalization, fused the color and texture features to perform the optimal similarity matching of the descriptors in the candidate frames of the previous and subsequent frames, and obtained the optimal solution for target tracking. However, the real-time performance of the algorithm was not enough. Voigtlaender [17] proposed Siam re-detection architecture combined with a trajectory-based dynamic programming algorithm using the first frame of labeling and the previous frame prediction for dual detection, modeling the complete history of tracking objects and potential interference objects. The tracked object could be re-detected after a long period of occlusion. Given that each frame of the image was double-checked, the algorithm complexity was high. Liu [18] proposed an algorithm for long-term target tracking. When the tracking failed, EdgeBox was employed to generate suggested regions. However, failure to consider the validity of the reference frame could cause tracking drift again. Xiong [19] proposed a target scale and rotation parameter estimation method based on kernel correlation filtering for the problem of target scale and rotation changes caused by long-term target tracking. When the target tracking was lost, the target search method combining color histogram and variance was started to determine the possible position of the target in the current frame, but the distance between the suggested area and the real position was often large. Gade [20] proposed a multi-object tracking algorithm suitable for team sports to solve the problem of target tracking in scenarios with similar target colors and rapid actions. Tracking failed after changing different scenes. Pak [21] designed a tracker to solve the problem of multi-object tracking in extremely crowded scenes, but the algorithm was not robust. Karunasekera [22] discussed the latest trends and the progress of tracking algorithms, compared the performance of trackers based on correlated filters and non-correlated filters, and provided an important reference for the research of target tracking algorithm. A slice of scholars have introduced a feature model based on the Bayesian probability framework into the algorithm, but the real-time performance is difficult to achieve given the high computational complexity. As the number of tracking targets increases, the computational complexity increases proportionally [23].

The abovementioned methods are not universal. In complex scenes, such as target occlusion, scale transformation, deformation, illumination, and motion blur, tracking loss or tracking errors are prone to occur. This study mainly focuses on the problem of long-term target tracking in complex environments, especially when the target object is severely occluded, deformed, and temporarily out of sight. The color feature has a small dependence on the size, direction, and viewing angle of the image, and the proposed method has a good tracking effect for a pure-color object under a black and white background and thus has high robustness. In a complex environment where the background color is similar to the target color, target tracking based on color features ignores the spatial distribution characteristics, and targets are easily lost. Texture feature is a statistical feature, and texture information is generally not disturbed by light or background color. Therefore, this study fuses the two features of image color and texture and uses the Bhattacharyya distance of different features between the candidate target and the target model to calculate the feature contribution. The CAMShift algorithm is executed by using the fused target features, combined with the Kalman filter, and finally iterated to the candidate target position. To make full use of the previous effective tracking results, a template pool is designed, which can be automatically updated for re-detection, and the elements of the template pool are used as the re-detection head to restore quickly the re-detection ability after the tracking target reappears.

The remainder of this study is organized as follows. Section 3 describes the extraction methods of different features, establishes a target feature model, and uses the CAMShift algorithm to converge to the candidate target. A re-detection algorithm is designed by combining the Kalman filter and the template pool. Section 4 verifies the effectiveness and robustness of the algorithm through experiments in two aspects, namely, quantitative and qualitative analyses. Section 5 summarizes the conclusions.

3. Methodology

The proposed algorithm mainly includes three parts, namely, the feature extraction module, the CAMShift algorithm module, and the re-detection module. A single feature is difficult to meet the needs of long-term target tracking in complex environments. Combined with color and texture features to represent an object, a single feature cannot solve the problems of target tracking, such as serious occlusion, deformation, illumination change, short-term out of sight, and similar color. The feature extraction module is detailed in Section 3.1. In the CAMShift algorithm module, mean shift algorithm is extended to video sequences and combined with the Kalman filter to achieve tracking (see Sections 3.2, 3.4, and 3.5 for details). In the re-detection module, a template pool is designed to store the correct tracking results.

The optimal elements in the template pool are selected as the initial frame of re-detection. The re-detection results are obtained by the mean shift algorithm (see Sections 3.3 and 3.4 for details). The algorithm model is shown in Fig. 1, and the main workflow of the algorithm is described in Section 3.5.

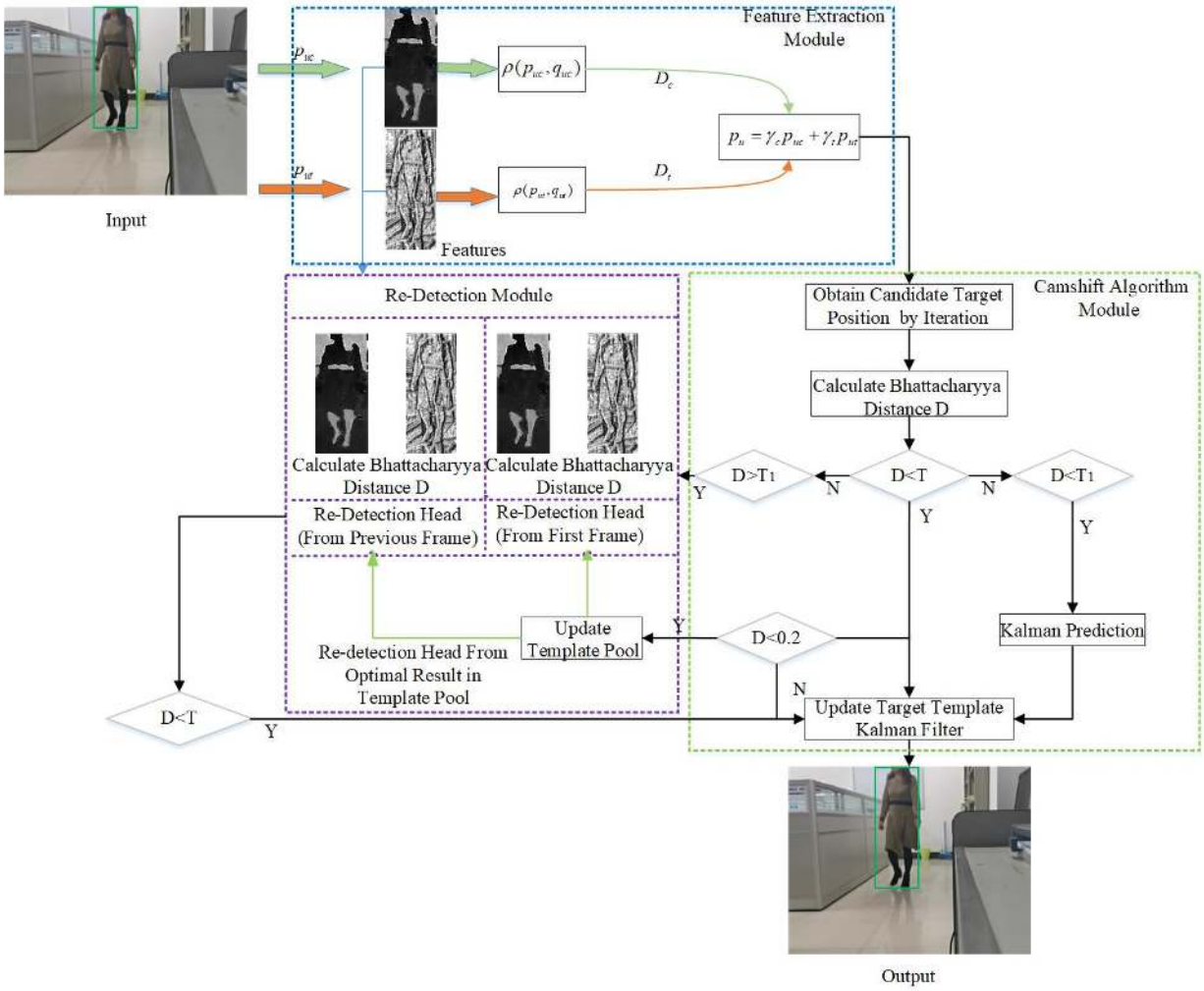


Fig. 1. Algorithm model

3.1 Feature extraction

The traditional CAMShift algorithm uses color probability density to represent the target. The color feature is not sensitive to deformation and rotation and is suitable for describing the target, whose color does not change. Changes in illumination cause the color of the target to change and make the tracking effect worse. In complex environments, where the background color and the target color are similar, target tracking based on color characteristics ignores spatial distribution characteristics, and the target is easy to lose. Texture features, as a statistical feature, are generally not disturbed by light or background color. LBP operator is employed to extract texture features to solve the problem of losing targets with similar colors.

3.1.1 Color feature model

The RGB color space is easily affected by changes in external lighting. Thus, converting the RGB color space into an HSV color space that is not sensitive to changes in lighting is necessary. To improve the robustness of tracking and reduce the influence of changes in illumination brightness, only the hue H component is extracted to build a color feature histogram.

Assuming that the center of the target area is x_0 and that $\{x_1, x_2, \cdots x_n\}$ are the other pixels in the target area, n is the total. The color feature value is set at $u = 1,2,\cdots,m$ of each pixel, where m is the grading of the color feature. The normalized histogram can be expressed as follows:

$$q_u(x_0) = C\sum_{i=1}^{n}\delta(b(x_i) - u), \tag{1}$$

where $\sum_{u=1}^{m} q_u = 1$. The histogram cannot suppress noise and easily leads to target loss. Therefore, the histogram can be weighted by increasing the kernel function, and different weights are assigned to pixels at different positions such that the weights of the points close to the target center are large. The kernel function estimates the color probability density as

$$f(x_0) = \frac{1}{nh^2}\sum_{i=1}^{n} K\left(\left\|\frac{x_0 - x_i}{h}\right\|^2\right). \tag{2}$$

$K()$ is called a kernel function, which usually conforms to symmetry and satisfies $\int K(x)dx = 1$. In the formula, h represents bandwidth, and the selection principle is to

minimize the mean square error. Combining Formulas (1) and (2), the probability density formula of the color feature of the target model can be obtained as follows:

$$q_{uc}(x_0) = C\sum_{i=1}^{n} K\left(\left\|\frac{x_0 - x_i}{h}\right\|^2\right)\delta(b(x_i) - u) \ , \tag{3}$$

where C is the normalization coefficient, $b(x_i)$ is the function that maps pixel x_i to the corresponding color feature, and δ is the impulse function.

3.1.2 Texture feature model

Before extracting texture features, performing an unsharp mask preprocessing operation on the image is generally necessary to improve the high-frequency components in the image and enhance image contour. The texture feature $LBP(x_c, y_c)$ of the image is obtained using the LBP operator. The LBP operator is defined in a 3×3 area. The pixel at the center of the area is taken as the threshold T, and the remaining 8 pixels in the area are compared with the center pixel. If the pixel value is greater than the central pixel value, the position is recorded as 1; otherwise, the position is recorded as 0. Convert the obtained 8-bit binary number into the decimal system as the LBP value of the window. If the local area of the pixel is represented by (P,R), $P$is the number of neighboring pixels (the value is 8), $R$is the distance between neighboring pixels and the center pixel (the value is 1), and (x_c, y_c) is the coordinate of the center pixel of the region. Then, the LBP texture feature of the center pixel is expressed as follows:

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(g_p - g_c) \tag{4}$$

$$s(g_p - g_c) = \begin{cases} 1, & |g_p - g_c| > T \\ 0, & |g_p - g_c| \le T \end{cases}. \tag{5}$$

In Formula (4), g_p represents the pixel value in the neighborhood, and g_c represents the pixel value in the center of the area. After the texture feature is extracted, the texture feature is modeled using a histogram. According to Formulas (1)–(4), the probability density formula of the texture feature of the target model can be easily obtained (the symbols in the formula have the same meaning as in Section 3.1.1) as follows:

$$q_{ut}(x_0) = C\sum_{i=1}^{n} K\left(\left\|\frac{x_0 - x_i}{h}\right\|^2\right)\delta(b(x_i) - u). \tag{6}$$

3.1.3 Feature fusion

Meanshift is the core of the CAMShift algorithm. $\mathrm{q_u}$ is the target color distribution, and $p_{\mathrm{u}}(x_i)$ is the color distribution of the i-th frame image candidate target. The Bhattacharrya coefficient $\rho(p,q)$ measures the similarity of the color distribution. The proposed method makes $p_{\mathrm{u}}(x_{i+1})$ and q_u the most similar by way of looking for x_{i+1} in the $i+1$ frame, where x_{i+1} is the new target center.

$$\rho(p,q) = \sum_{u=1}^{m} \sqrt{p_{\mathrm{u}}(x_{i+1})\mathrm{q}_u} \tag{7}$$

Given that the color feature and texture feature models have different abilities to express and distinguish the target, the contribution of the color feature and the texture feature is dynamically calculated according to the actual scene to realize the adaptive fusion of the two. According to the Bhattacharrya coefficient $\rho(p,q)$ in Formula (7), the similarity between p and q is measured. The color features of the target model and the candidate target are q_{uc} and p_{uc}, respectively, and the texture features are q_{ut} and p_{ut}, respectively, where $u = 1,2,\cdots,m$. The distance between the target model and the candidate target is defined as:

$$D = \sqrt{1 - \rho(p,q)}. \tag{8}$$

According to Formulas (7) and (8), the distances D_c and D_t between the color and texture features of the candidate target and the target model are obtained, respectively.

$$D_c = \sqrt{1 - \rho(p_{uc}, q_{uc})} = \sqrt{1 - \sum_{u=1}^{m} \sqrt{p_{uc} q_{uc}}} \tag{9}$$

$$D_t = \sqrt{1 - \rho(p_{ut}, q_{ut})} = \sqrt{1 - \sum_{u=1}^{m} \sqrt{p_{ut} q_{ut}}} \tag{10}$$

By calculating D_c and D_t, the feature contributions of the color and texture features are obtained, which are represented by γ_c and γ_t, respectively. The candidate target feature model after fusing the color and texture features is as follows:

$$p_u = \gamma_c p_{uc} + \gamma_t p_{ut}, \tag{11}$$

$$\gamma_c = \frac{D_c}{D_c + D_t}, \gamma_t = \frac{D_t}{D_c + D_t}. \tag{12}$$

3.2 Kalman position prediction

To make the candidate target search window closer to the position of the real target, Kalman prediction is introduced to reduce the search time of algorithm tracking further. Using the linear system state equation through the input and output observation data, the state of the system is optimally estimated. The basic principle of the prediction process is as follows:

The system dynamic state equation at time k is

$$X(k) = AX(k-1) + BU(k) + W(k). \tag{13}$$

The system measurement value at time k is

$$Z(k) = H(k)X(k) + V(k-1). \tag{14}$$

Among them, A is the system state transition matrix, $B$is the system parameter generally taking the value 0, and $U(k)$is the system control variable. $H(k)$ is the observation

matrix of the system, and $W(k)$ and $V(k)$ are white noise sequences with a mean value of 0. The proposed algorithm uses the optimal result $X(k-1|k-1)$ of the previous state of the system to predict the current state of the system, and the prediction result is $X(k|k-1)$:

$$X(k|k-1) = AX(k-1|k-1) + BU(k). \quad (15)$$

The error estimate covariance matrix of the optimal result of the last state of the system is $P(k-1|k-1)$. The covariance of $X(k|k-1)$ can be updated to

$$P(k|k-1) = AP(k-1|k-1)A^T + Q, \quad (16)$$

where Q is the covariance of the system. According to Formulas (15) and (16), the optimal prediction result $X(k|k)$ of the current state is obtained, and $K_g(k)$ is the Kalman gain.

$$X(k|k) = X(k|k-1) + K_g(k)(Z(k) - HX(k|k-1)) \quad (17)$$

$$K_g(k) = P(k|k-1)H^T / HP(k|k-1)H^T + R \quad (18)$$

Finally, to achieve the purpose of continuous tracking, the covariance $P(k|k)$ of $X(k|k)$ in the state k is updated:

$$P(k|k)=(I - K_g(k)H)P(k|k-1). \quad (19)$$

The Kalman filter can predict where the target may appear in the next frame based on specific target information.

3.3 Template pool

To solve the problems of tracking drift and failure that may occur in long-term tracking, selecting an initial frame is the key to recover the tracking quickly and accurately. If the effective tracking results generated during the tracking process are not saved, redetection will be challenging. Therefore, a template pool is designed to store the correct tracking results. The records in the template pool are called elements, and the elements record location information, the Bhattacharyya distance, and the use times of the tracking results. First, the first frame of the image is permanently placed in the template pool, and the frames whose Bach distance is less than the threshold T_2 are input into the template pool during the tracking process. To avoid wasting space and improve the efficiency of the algorithm, the update interval of the template pool is set to γ. When the template pool is full, the outdated tracking results are replaced by factors such as the Bhattacharyya distance, the least recent used time, and used time.

3.4 Template update

The Bhattacharyya coefficient is employed to calculate the distance D between the candidate target and the target model. The smaller the value of D, the higher the similarity between the two models. The threshold is set to T. When D is less than the threshold, the candidate target matches high. When D is greater than the threshold, the tracked preferred region has low similarity to the target template. The template update principle of the combination of CAMShift algorithm and Kalman filtering is as follows:

1) If the Bhattacharyya distance between the candidate target and the target model is $D \le T$, then occlusion does not exist or has a little effect. The prediction result of the CAMShift algorithm is used to update the target template. If $D \le T_2$, then the prediction result is added to the template pool at the same time.

2) If the Bhattacharyya distance is $T < D \le T_1$, then the target is blocked by a large area, or the background noise is large. The Kalman filter is employed to store the motion information of the tracking target to establish the target motion equation, predict the target position, and update the target template.

3) If the Bhattacharyya distance is $T_1 < D \le 1$, then the target is completely occluded, and the position predicted by the Kalman filter is no longer accurate. Thus, the re-detection procedure starts.

In the normal tracking process, the target template of the previous frame is generally employed to find the target position of the current frame. However, the tracking result of the previous frame is no longer reliable during re-detection. Selecting an image from the template pool as the re-detection head is necessary to re-detection (the first frame target template is one of the re-detection heads by default). The Bach distances of all images in the template pool are read, and the Euclidean distance between the current frame (b_t) and the images in the template pool ($b_j, j = 1 \to 10$) is calculated.

$$d(b_t, b_j) = \exp(-\frac{1}{2\sigma^2}\left\|(x_t, y_t) - (x_j, y_j)\right\|^2), \quad (20)$$

where σ is the diagonal length of the initial target size. The best element is found based on the Bhattacharyya distance and the Euclidean distance as the re-detection head, and the normal tracking mode is resumed.

$$\arg\min_j \beta D_t + (1-\beta)d(b_t, b_j), \quad (21)$$

where β is a weight parameter, which adaptively adjusts the contribution of the Bhattacharyya distance and Euclidean distance. According to Formula (20), the best matching result β is found from the template pool and used as the initial frame of re-detection.

3.5 Algorithm flow

The Meanshift algorithm is extended to the whole video sequences, all image frames are involved in the calculation. The HSV color model and texture features are selected as tracking features, and the joint probability distribution in the region is calculated. Using the center and size of the search window in the previous frame as the initial value, the kernel bandwidth is adaptively updated to define the size of the search box, iteratively positioning the target center position in the current frame, continuously adapting to target changes, and achieving target tracking. When the target is occluded and is out of sight for a short time, according to the Kalman filter and the re-detection module, the tracking ability can be quickly restored.

The algorithm is described as follows:

Step 1. Read the video frame and determine the initial target window.

Step 2. Use the center position of the target tracking window to initialize the relevant parameters of the Kalman filter.

Step 3. Extract the color and texture features q_{uc} and q_{ut} of the target window and calculate the target joint feature model $q=\{q_u\}, u=1,2,\cdots m$ according to Formula (11).

Step 4. Execute the Meanshift algorithm and iterate to the candidate target position.

Step 5. Calculate the Bhattacharyya distance D between the candidate target and the target template and compare D with the threshold.

Step 6. If D is less than the threshold (the threshold is T), use the result obtained in Step 4 as the tracked target position and turn to Step 10.

Step 7. If D is greater than the threshold, use the Kalman filter to predict the location of a specific target and turn to Step 10.

Step 8. If D is close to 1, completely block the target, and start the re-tracking procedure (see Section 3.4 for details).

Step 9. Select the appropriate element in the template pool as the target template and turn to Step 3.

Step 10. Use the result obtained in Step 5 to update the Kalman filter and initialize the target position as the next frame of the image. Turn to Step 3 to continue execution until the end of the video sequence.

4. Result analysis and discussion

To verify the effectiveness of the algorithm and long-term tracking performance, the proposed algorithm and the traditional CAMShift algorithm [12][14] are compared on the OTB2015 [24]. These image sequences have various problems, such as target deformation, occlusion, similar background color, scale conversion, motion blur, and lighting effects.

4.1 Parameter settings

The experimental simulation environment is MATLAB R2018b. The computer configuration is as follows: Intel Core i7-8550U CPU, 2.0 GHZ frequency, 8GB memory, Windows 10 operating system. The update interval γ of the template pool is set to 0.2 seconds, and the thresholds T, T_1, and T_2 are 0.5, 0.9, and 0.2, respectively.

4.2 Evaluation index

1) Distance precision

The tracking algorithm estimates the Euclidean distance between the center point of the target position and the center point of the manually marked target. Otherwise, the smaller the Euclidean distance, the higher the tracking accuracy. Range accuracy refers to the percentage of video frames whose Euclidean distance is less than a given threshold to the total number of frames. With different thresholds, ratios are different, a curve can be obtained, and the threshold is set to 20 pixels.

2) Success rate

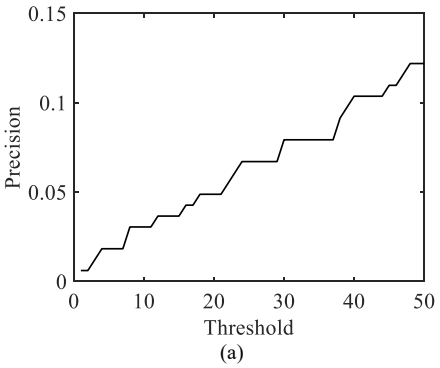
The tracking algorithm estimates the overlap rate of the target position bounding box S_p and the target position true bounding box S_g. It is calculated by the following formula:

$$S=\frac{\left|Area(S_p \cap S_g)\right|}{\left|Area(S_p \cup S_g)\right|} \tag{22}$$

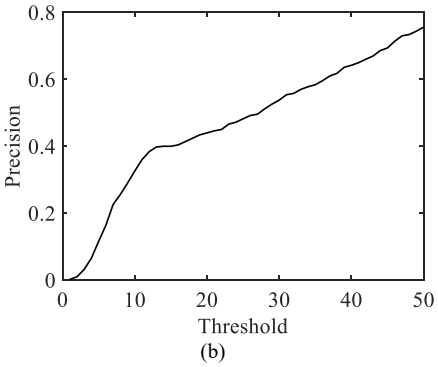
The larger the overlap rate, the higher the tracking success rate, and the threshold is 0.5. If the overlap rate is greater than 0.5, then the tracking result is successful.

4.3 Quantitative analysis

Tracking objects in MotorRolling video sequences have the characteristics of fast motion, rotation, and large light changes. Fig. 2(a) shows the accuracy of the tracking result of the traditional CAMShift algorithm in a MotorRolling video sequence. The accuracy is low and close to 0. The target can only be tracked in the first few frames of images. When the target reaches the highest point, the tracking drifts because of the influence of illumination. In the later tracking process, the correct target was never obtained. Fig. 2(b) is the accuracy curve of the improved CAMShift algorithm [12] that combines color and texture features, and the accuracy is greatly improved. However, at the beginning of 121 frames of images, the tracking drift phenomenon appears under the interference of motion blur and light reflection. Fig. 2(c) is the accuracy curve of the joint color and texture features designed in this study, which combined with the Kalman filtering algorithm. Given the addition of the re-detection module, the tracking capability can be quickly restored when individual frames drift. As the threshold continues to increase, the accuracy rate becomes higher, approaching 1. The proposed algorithm maintains good tracking ability and stability when dealing with complex environments, such as target deformation, large background noise, and light changes.



(a)



(b)

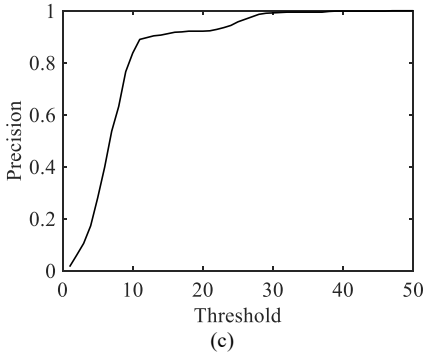


Fig. 2. Precision comparison of different algorithms. (a) Precision of the traditional CAMShift algorithm. (b) Precision of algorithm [12]. (c) Precision of this study

Fig. 2 shows that the proposed algorithm has the highest distance accuracy. Fig. 3 shows the comparison curve of success rates of different algorithms. The x-axis represents the overlap threshold and the y-axis represents the success rate. Pc1 represents the traditional CAMShift algorithm [4], Pc2 represents the tracking algorithm that combines color features and re-detection modules, Pt1 represents the tracking algorithm with only texture features, and Pt2 represents the combination of texture features and re-detection. Pct1 represents the tracking result of the comparison algorithm [12], and Pct2 represents the tracking result of the comparison algorithm [14]. According to the calculation rules of the tracking success rate, the proposed algorithm has the highest comprehensive score, with an average success rate of 0.83.

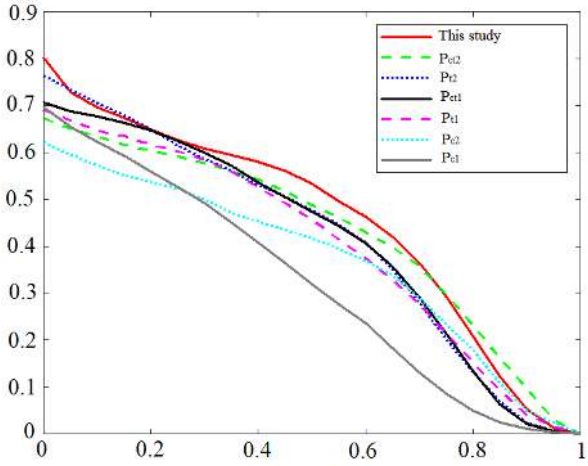
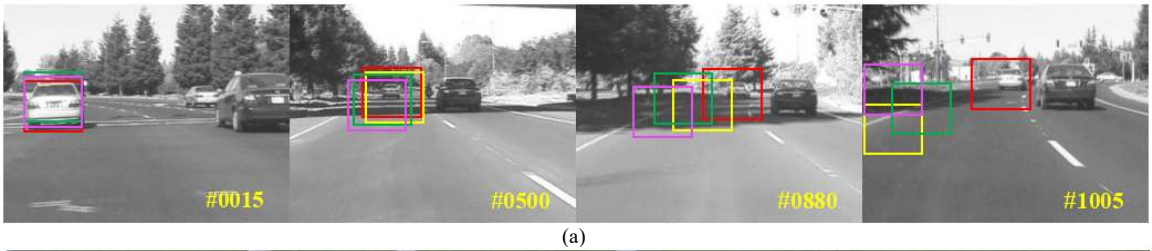


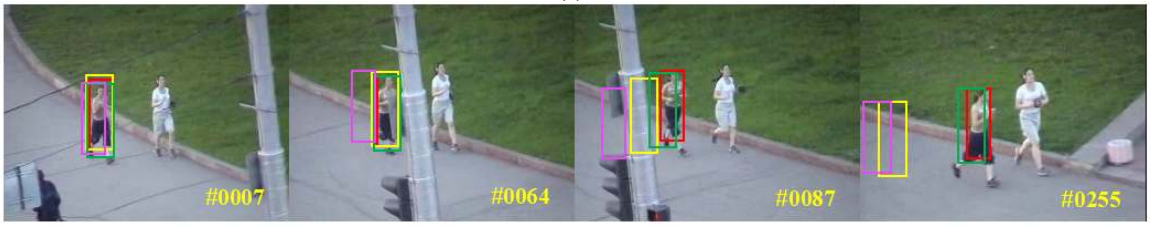
Fig. 3. Comparison of success rates of different algorithms

4.4 Qualitative analysis

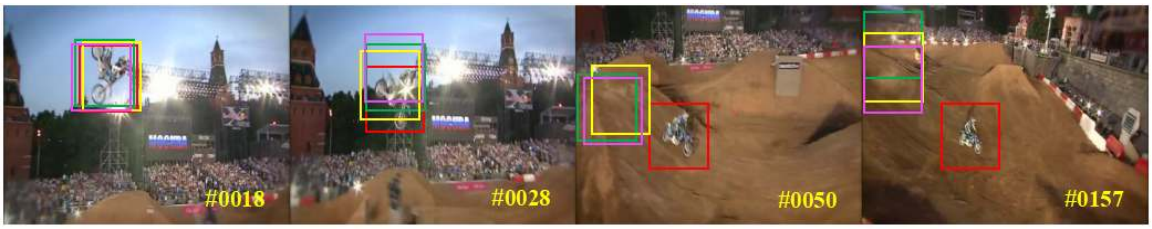
To illustrate intuitively the accuracy and robustness of the algorithm in this study, Fig. 4 shows the comparison results of three typical tracking video sequences, namely, Car1, Jogging, and MotorRolling, on the OTB2015 dataset with the comparison algorithm. Given that the scale change and illumination change of the tracking target in the video sequence Car1 are excessively large and the resolution is low, the proposed algorithm causes quite a few tracking drifts when the target is turning. In other video sequences, the proposed algorithm has achieved better tracking results. As the target gradually moves away from the line of sight, other comparison algorithms show the tracking drift in the video sequence Car1. When the target turns, the target can no longer be tracked, and the tracking is not resumed. In the Jogging video sequence, the target appears completely occluded from frame 69 and reappears in frame 77. The proposed algorithm resumes normal tracking immediately after the target appears. The comparison algorithm [12] uses the Kalman filter to predict the position of the target after the target appears, but the tracking result is somewhat shifted. In the MotorRolling video sequence, the target at frame 19 is at the highest point, and the background light interference is large. In the following tracking, only the proposed algorithm can track the target normally, and other comparison algorithms have tracking drift. The qualitative analysis results show that the advantages of the proposed fusion algorithm are obvious, which further verifies the effectiveness of the re-detection module in the tracking process.



(a)



(b)



(c)

The proposed algorithm CAMShift algorithm Algorithm [12] Algorithm [14]

Fig. 4. Comparison results of algorithm visual effects. (a) Video sequence of Car1. (b) Video sequence of Jogging. (c) Video sequence of MotorRolling

5. Conclusions

To solve the problem of long-term moving target tracking in complex environments, such as target deformation, short-term out of sight, occlusion, and similar colors, the proposed algorithm started with the extraction of robust features, analyzed the features with different discrimination capabilities, and used the CAMShift algorithm to converge. The tracked area was combined with the Kalman filter to predict the location of a specific target in the next frame, reducing the algorithm time. The following conclusions could be drawn:

(1) After combining the color features and texture features to establish a target feature model and using the mean shift algorithm to converge to the candidate target, the proposed algorithm has improved the robustness of target tracking in complex environments, such as target deformation, severe occlusion, short line of sight, and similar colors.

(2) The proposed algorithm combines the Kalman filter and the re-detection module of the template pool to solve the problem of target tracking failure caused by complex factors, such as complete occlusion, uneven illumination, deformation, and similar colors.

(3) According to the Bhattacharyya distance between the candidate target and the target model, the proposed algorithm automatically selects the CAMShift algorithm or the prediction result of the Kalman filter to update the target template to ensure accurate tracking.

The proposed algorithm shows good tracking performances when dealing with challenging video scenes, such as occlusion, scale conversion, uneven illumination, and similar colors. This study has a certain reference for the subsequent development of the automatic detection of traffic incidents in intelligent traffic monitoring systems. However, when tracking video sequences with motion blur and target rotation attributes, tracking ability is slightly weaker. In the following study, the algorithm will be improved by way of image preprocessing and scale filter.

Acknowledgements

This work was supported by the Ministry of Housing and Urban-rural Development Science and Technology Planning Project (2016-R2-060), the National Key Research and Development Plan of China (No.2017YFC0804400, No.2017YFC0804401), the Project of Jiangsu Province Construction System(2018ZD077).

This is an Open Access article distributed under the terms of the Creative Commons Attribution License

References

1. Li, Q. J., Li, L. M., Huang, Y. Q., "Scale adaptive tracker based on kernelized correlation filtering". *Journal of Computer Applications*, 36(12), 2018, pp.253-257.
2. Comaniciu, D., Meer, P., "Mean shift: a robust approach toward feature space analysis". *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 24(5), 2002, pp.603-619.
3. Bradski, G. R., Clara, S., "Computer vision face tracking for use in a perceptual user interface". *Intel Technology Journal*, 2, 1998, pp.1-15.
4. Gao, C. Q.,"*Video tracking based on camshift algorithm and multi-feature fusion*". Master thesis of Shandong Normal University, China, 2020, pp.27-49.
5. Li, X. B., Xu, G. Q., Yang, X. Y., Zhao, G. Y., "A multi feature tracking algorithm based on camshift".*Computer & Digital Engineering*, 48(1), 2020, pp.73-77.
6. Zhao, G. H., Zhuo, S., Xu, X. L., "Multi-object tracking algorithm based on kalman filter". *Computer Science*, 45(8), 2018, pp.253-257.
7. Liu, Q. Y., Wang, Y. R., Zhang, Y. L., Yin, M. H., "Research progress of visual tracking methods based on correlation filter". *Acta Automatica Sinica*, 45(2), 2019, pp.265-275.
8. Majd, M., Safabakhsh, R., "Correlational convolutional LSTM for human action recognition". *Neurocomputing*, 396, 2020, pp.224-229.
9. Wang, X. D., Wang, Y. W., Yan, H., "Continuously adaptive mean-shift tracking algorithm with suppressed background histogram model". *Journal of Electronics & Information Technology*, 41(6), 2019, pp.1480-1487.
10. Howard, W., Nguang, S. K., Wen, J. W., "Robust video tracking algorithm: a multifeature fusion approach". *IET Computer Vision,* 12(5), 2018, pp.640-650.
11. Muhammad, A. H., Goutian, Y., Atif, I., "Autonomous swimmers tracking algorithm based on kalman filter and camshift". In: *2019 13th International Conference on Open Source Systems and Technologies*, KICS UET Lahore, Pakistan: IEEE, 2019, pp.1-6.
12. Yu, G. Q., Wu, S. M.,"Improved camshift algorithm combining with kalman filtering and lbp texture". *Modern Electronics Technique*, 43(12), 2020, pp.65-68.
13. Razzaq, M. A., Quero, J. M., Cleland, I., Nugent, C., Akhtar, U., Syed, H., Bilal, M., Rehman, U. U., Lee, S., "uMoDT: An unobtrusive multi-occupant detection and tracking using robust kalman filter for real-time activity recognition". *Multimedia Systems*, 26(5), 2020, pp.553-569.
14. Chu, H. X., Xie, Z. Y., Wang, K. J., "An improved camshift target tracking algorithm based on joint color-texture histogram". *Journal of Xi'an Jiaotong University*, 52(3), 2018, pp.145-152.
15. Zhang, Z. L.,Wang, Y. X., "SiamRPN target tracking method based on kalman filter". *Intelligent Computer And Applications*, 10(3), 2020, pp.44-50.
16. Zhang, J., Huang, H. M.,Wang, J. M., Bao, J. R., "An improved tld real-time target tracking algorithm based on cn algorithm". *Computer Engineering & Science*, 42(7), 2020, pp.1215-1225.
17. Voigtlaender, P., Luiten, J., Torr, P., Leibe, B., "Siam R-CNN: visual tracking by re-detection". In: *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, USA: IEEE, 2020, pp.1-17.

18. Liu, H., Hu, Q. Y., Li, B., Guo, Y. L., "Robust long-term tracking via instance-specific proposals". *IEEE Transactions on Instrumentation and Measurement*, 69(4), 2020, pp.950-962.
19. Xiong, D., Lu, H. M., Xiao, J. H., Zheng, Z. Q., "Robust long-term object tracking with adaptive scale and rotation estimation". *Acta Automatica Sinica*, 45(2), 2019, pp.289-304.
20. Gade, R., Moeslund, T. B., "Constrained multi-target tracking for team sports activities". *Ipsj Transactions on Computer Vision & Applications*, 10(1), 2018, pp.1-11.
21. Pak, J. M., "Visual odometry particle filter for improving accuracy of visual object trackers". *Electronics Letters*, 56(17), 2020, pp.884-887.
22. Karunasekera, H., Wang, H., Zhang, H., "Multiple object tracking with attention to appearance, structure, motion and size". *IEEE Access*, 7, 2019, pp.104423-104434.
23. Lee, D-H., "One-shot scale and angle estimation for fast visual object tracking". *IEEE Access,*7, 2019, pp.55477-55484.
24. Wu, Y., Lim, J., Yang, M-H., "Object tracking benchmark". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9), 2015, pp.1834-1848.