

## Continuously Adaptive Mean-shift Tracking Algorithm Based on Improved Gaussian Model

Jinping Sun<sup>1,2\*</sup>, Enjie Ding<sup>1</sup>, Dan Li<sup>2</sup>, Kailiang Zhang<sup>2</sup> and Ximin Wang<sup>3</sup>

<sup>1</sup>School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221008, China

<sup>2</sup>School of Information Engineering (School of Big Data), Xuzhou University of Technology, Xuzhou 221008, China

<sup>3</sup>School of Computer Science, University of Nottingham, Nottingham NG8 1AF, The United Kingdom

Received 6 August 2020; Accepted 29 September 2020

### Abstract

In complex scenes with light changes, deformations, and occlusions, target tracking easily contains a large amount of background color information when building a target color model. Thus, the tracking effect is reduced. To improve the accuracy of the traditional continuously adaptive mean-shift algorithm (CAMShift) in complex scenarios, a target tracking algorithm based on an improved Gaussian mixture model was proposed. Using the Gaussian mixture model, the tracking image was divided into the foreground and background superposition. The histograms of the hue component were respectively established in the foreground and background of the target area. By suppressing the same hue as the background color in the tracking image, the target color model was established. The target position was iteratively obtained by implementing the CAMShift algorithm using the enhanced target color model. The Bhattacharyya distance between the candidate target and the target template was used as basis for updating the target model. Simulation analysis under benchmark data sets and actual monitoring scenarios verified the accuracy of the proposed algorithm. Results show that the distance precision and overlap success rate of the proposed algorithm are 0.88 and 0.625, respectively. The proposed algorithm effectively solves long-term target tracking problems with complex scenes, such as occlusion, background clutters, and illumination variation. This study eliminates the problem of target recognition caused by environmental changes and provides references for real-time monitoring of abnormal traffic conditions.

*Keywords:* Target tracking, Hue, Gaussian mixture model, Bhattacharyya distance, Continuously adaptive mean-shift tracking algorithm

### 1. Introduction

With the rapid development of computer-related technologies, moving target detection and tracking technologies are widely used in the field of intelligent traffic monitoring [1]. Intelligent analysis of the events in the monitoring requires accurate detection and continuous tracking of the moving target in the image [2-3]. Motion detection is a comparative analysis of data frames generated after encoding, and is a feasible method to identify image changes by comparing video frames. The current popular detection methods mainly include frame difference, optical flow, and background modeling. In recent years, the mean shift algorithm [4] has been widely used because of its simple calculation and high performance. However, the mean shift algorithm cannot update the target model in real time. When the size of the target changes during the movement, the tracking performance of the mean shift algorithm decreases significantly. On the basis of the mean shift algorithm, Bradski proposed the continuously adaptive mean shift algorithm (CAMShift) [5] that can adaptively adjust the target size and have good adaptability to changes in target translation, rotation, and scaling.

However, the mean shift and the CAMShift algorithms build target models in the form of color histograms, and thus are both suitable for target tracking systems with clearly distinguished target and background. Thus, the tracking accuracy decreases if the background and the target have

similar colors. The background information is inevitably calculated when the histogram of the tracking target is established, which affects the tracking accuracy.

To solve the above-mentioned problems, auxiliary features such as edges and textures are integrated in the color-based tracking model [6-7] to improve the tracking effect. However, such integration increases the time complexity of the algorithm, and deviations occur in the extraction process of auxiliary features [8], thereby reducing the algorithm tracking performance. If the difference between the foreground and the background is increased and the expressive ability of the foreground target is emphasized, the tracking effect improves while ensuring the calculation speed.

On this basis, this study examines such core problem in the background model design. Subsequently, the CAMShift algorithm is used to converge the area to be tracked and the accuracy of target detection is improved without the addition of auxiliary features.

### 2. State of the art

Moving object tracking under complex background is an important issue in the field of intelligent video surveillance. Extensive studies have attempted to improve the tracking performance of the CAMShift algorithm. Bae [9] proposed a multi-target tracking method that achieved competitive tracking accuracy while maintaining low complexity. However, the algorithm lacked in analysis in actual

\*E-mail address: sjp614@163.com

ISSN: 1791-2377 © 2020 School of Science, IHU. All rights reserved.

doi:10.25103/jestr.135.07

application scenarios and proving its robustness was impossible. Wang [10] put forward a continuous adaptive mean-shift tracking algorithm for the background suppression histogram model. Despite its improvement of tracking accuracy and stability by suppressing the background hues in the original color model, the method could not be used for target tracking. The target model was updated and the target was easily lost when its size changes. Howard [11] extended the CAMShift algorithm by adding a fast-moving target state prediction algorithm with adaptive kernel bandwidth and reducing the average displacement iteration. However, the tracking was easily lost when occluded. Combining the CAMShift algorithm and the kernel-related filtering algorithm, Guo [12] and Su [13] respectively presented the background subtraction and the frame difference methods to realize target tracking in different situations. The resulting tracking effect was poor when the tracking target was severely deformed. Bankar [14] used the CAMShift algorithm to predict the head pose and provided a certain reference for target detection. However, recognition ability was poor in low-resolution scenes. Hayat [15] explored Canny edge detection and the inter-frame difference method to extract moving targets to initialize the CAMShift algorithm. The CAMShift algorithm combined with the Kalman filter to achieve accurate target tracking when the background had similar color interference. However, tracking was unsuccessful for a fast-moving target. Combining the hue H and saturation S components, Li [16] designed a multi-feature tracking algorithm (MFTA) based on improved CAMShift to enhance the method adaptability to complex environments. However, the algorithm had poor tracking performance in scenes with obvious lighting changes and low resolution. Razzaq [17] used the difference between frames and a set of morphological operations to extract feature, and then applied Kalman filter to detect the object. However, tracking was also unsuccessful for a fast-moving target. Guan [18] not only combined the CAMShift algorithm with the Kalman filter but also introduced the Bhattacharyya coefficient to judge the tracking accuracy, achieving high positioning accuracy and good real-time capability. However, the limited data set used for testing could not reflect the robustness of the algorithm. Dehghan [19] designed a tracker for multi-target tracking in extremely crowded scenes, but the algorithm lacked robustness. Nie [20] developed a target model to effectively solve the target tracking failure caused by occlusion. However, the algorithm could not track targets with long occlusion or serious deformation. Chu [21] fused color and texture histograms and used particle filter to estimate the state of moving targets, but did not consider target occlusion. Gade [22] and Voigtlaender [23] respectively proposed a multi-target tracking algorithm for target tracking in scenarios with similar target colors and rapid action. Tracking was unsuccessful when the target changed to different scenes. Islam [24] designed a tracker with scale adaptation and online re-detection to solve the problem of scale changes. However, the features proposed by the algorithm could not be used for non-rigid target tracking.

With the unsatisfactory or low efficiency of different tracking algorithms in solving different complex scenes, the target tracking technology requires further study to improve the tracking efficiency and effect. Target tracking based on color features easily ignores the spatial distribution characteristics in environments with similar background and target colors, resulting in target loss. A large amount of

background color information can be included by mistake when building the target color model, thereby reducing the tracking effect. Using the Gaussian mixture model, the tracking image is divided into the superpositions of the foreground and background. The histograms of the hue component are established in the foregrounds and background of the target area. The target color model is obtained by suppressing the same hue as the background color in the tracking image, thereby expanding their color differences. The target position is iteratively obtained by implementing the CAMShift algorithm on the enhanced target color model. The Bhattacharyya distance between the candidate target and the target template is used as basis for updating the target model.

The remainder of this study is organized as follows. Section 3 describes the process of establishing the target histogram model using the CAMShift algorithm to converge to the candidate target. A target update strategy is designed according to the Bhattacharyya distance between the candidate target and the target template. 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 target area is represented by extracting the histogram of the foreground image in the target model. Executing the CAMShift algorithm can iteratively obtain the candidate target position. The Bhattacharyya coefficient is used to calculate the distance between the candidate target and the target model, and serves as basis for whether to update the target model. As the Bhattacharyya distance decreases, the similarity between the two models increases.

#### 3.1 Establishment of target color model

In actual environments, the background has weak or obvious changes as time passes, such as the illumination change, environmental effect, and movement. If the original background model is always used, then a large error may occur. Thus, the background is needed to update in real time. A fixed threshold is unsuitable for illumination change in the environment. For detection of moving targets, the system uses an improved adaptive Gaussian mixture model.

For a location point on the image plane, the historical data of  $i$  are recorded as  $\{X_1, X_2, \dots, X_{t-1}\}$ . The pixel characteristics are described by L Gaussian distribution and the background gradient is represented by online updating. L=8 is selected for modeling in the program. The estimated probability distribution of t time observations is as follows:

$$P(X_t) = \sum_{i=1}^L \omega_{i,t-1,l} \times \eta_l(x_{i,t}, \mu_{i,t-1,l}, \sum_{i,t,l}), \quad (1)$$

where  $\omega_{i,t-1,l}$  represent the weights of  $l$  Gaussian distributions at  $t-1$  time,  $\mu_{i,t-1,l}$  is the mean vector of Gaussian distribution,  $\eta_l$  is the probability density function of  $l$  Gaussian distributions, and  $\sum_{i,t,l}$  is a covariance matrix, respectively. The L models are queued according to probability from large to small. The front and back of the Gaussian model represent the background and the foreground of the image, respectively. In the next frame, the

brightness values of the pixels in this position are matched with  $L$  models to determine whether they belong to the foreground or background. The updating model is obtained on the basis of the maximum matching degree.

The model with the maximum matching degree requires a large amount of computation, which is not conducive to real-time processing. Both the probabilities of the luminance value of the new input pixel in a certain model and of each Gaussian model are needed for the calculation. The probability of the occurrence of a Gaussian model can be approximately expressed by its weight  $\omega$ . A larger  $\omega$  is easier to match than a small one. A small variance indicates high model stability. Therefore, the approximate algorithm is adopted to rank the Gaussian distribution from smallest to largest by  $\omega/\sigma$ .  $\mu_{i,t-1,k}$  and  $x_{i,t}$  are selected for their proximity as the best matched Gaussian distribution, which is  $|x_{i,t} - \mu_{i,t-1,k}|^2 < (\beta\sigma)^2$ .

The model is updated according to the matching results of the current pixel with  $K$  Gaussian distributions. The mean and variance of the unmatched model should remain unchanged, while the  $i$ -th successfully matched model is updated by the following formula,

$$\begin{cases} u_{i,t} = (1-\beta)\mu_{i,t-1} + \beta X_{i,t} \\ \sigma_{i,t}^2 = (1-\beta)\sigma_{i,t-1}^2 + \beta(X_{i,t} - \mu_{i,t-1})^T (X_{i,t} - \mu_{i,t-1}), \\ \omega_{i,t,k} = (1-\beta)\omega_{i,t-1,k} + \beta M_{i,t,k} \end{cases} \quad (2)$$

where  $\beta$  is the learning rate that reflects the rate at which the current pixel is integrated into the background model and  $M_{i,t,k}$  is a marker function.  $n=i$  indicates that the current observation value matches the  $i$ -th Gaussian model. When the weight increment of the Gaussian model is  $\beta(1-\omega_{i,t-1,k}) > 0$ , then  $M_{i,t,k}=1$ ; otherwise  $M_{i,t,k}=0$ . The weight increment of the Gaussian model is  $-\alpha(\omega_{n,t-1,k}) < 0$ , and the weighted sum of  $K$  Gaussian models after adjustment remains 1. If the learning constant  $\beta$  is too large, the foreground melts into and can easily be mistaken for the background while the foreground is moving too slowly. If the learning constant  $\beta$  is too small, the background extraction time increases. The learning constant increases and is greater when the object is moving fast than when movement is slow. The steps to obtain the foreground histogram are as follows:

1) Create histogram  $q_{b|f}$  of the foreground and background within the search window  $(W, H)$ , assuming  $L$  tonal intervals occur.

2) For the background, use the improved Gaussian mixture model and establish histogram  $q_b$  in the same area as  $(W, H)$ .

3) For the background histogram  $q_b$ , calculate the background weighting factor  $\alpha_i$  of each hue interval according to Formula (3),

$$\alpha_i = 1 - \frac{q_i}{\sum_j q_j}, \quad (3)$$

where  $q_i$  represents the histogram statistical value of the  $i$ -th hue interval,  $i \in [1, 2, \dots, L]$ .

4) The interference caused by the background histogram is reduced by recalculating the histogram  $q_{b|f}$  according to the background weighting factor. The calculation formula is as follows:

$$q'_{b|f}[i] = \alpha_i q_{b|f}[i] = q_{b|f}[i] - \frac{h_i}{\sum_j h_j} \times q_{b|f}[i], \quad (4)$$

where  $q_{b|f}[i]$  is the statistical value of the  $i$ -th hue interval in the histogram  $q_{b|f}$  and  $q'_{b|f}[i]$  is the statistical value after reducing the background interference.

### 3.2 Update of target template

$q_{b|f}$  is the color distribution of the target and  $p_{b|f}(x_t, y_t)$  is the color distribution of the  $t$ -th frame candidate target. The Bhattacharyya coefficient  $\rho(p, q)$  measures the similarity of color distribution. The proposed method causes the most similarity of  $p_{b|f}(x_{t+1}, y_{t+1})$  and  $q_{b|f}$  by looking for the new target  $(x_{t+1}, y_{t+1})$  in the  $t+1$  frame.

$$\rho(p, q) = \sum_{u=1}^m \sqrt{p_{b|f}(x_{t+1}, y_{t+1}) q_{b|f}} \quad (5)$$

The similarity between  $p$  and  $q$  is measured according to the Bhattacharyya coefficient  $\rho(p, q)$  in Formula (5). The color features of the target model and the candidate target are  $q_{b|f}[i]$  and  $p_{b|f}[i]$ , respectively, where  $i \in [1, 2, \dots, L]$ . The distance between the target model and the candidate target is defined as

$$D = \sqrt{1 - \rho(p, q)} = \sqrt{1 - \sum_{i=1}^L \sqrt{p_{b|f}[i] q_{b|f}[i]}}. \quad (6)$$

The Bhattacharyya coefficient is used to calculate the distance  $D$  between the candidate target and the target model. As the value of  $D$  decreases, the similarity between the two models increases. When  $D$  is less than the threshold  $T$ , the candidate target matching is high. The prediction result of the CAMShift algorithm is used to update the target template. When  $D$  is greater than the threshold  $T$ , the target template is not updated when the tracked candidate area has low similarity to the target template.

### 3.3 Target tracking algorithm based on color features

The coordinates of the target center position of the initial frame are given as  $(x_c, y_c)$  and rectangular areas are drawn around their center. The rectangular area is 1.5 to 2 times of the target area, and can contain enough samples and the background information to increase the robustness of the trained filter template. The rectangular area is expressed as  $(W, H) = \text{sizeof}(target) * (1 + padding)$ , where  $W$  and  $H$  are the width and height of the area, respectively.

For 2D image probability distributions, the size and direction angle of the tracked target can be estimated by calculating the zero, first, and second moments within the search window  $(W, H)$ . Find the zero moment,

$$M_{00} = \sum_{x_i} \sum_{y_i} I(x_i, y_i). \quad (7)$$

The first moment of the search window is described as follows:

$$\begin{cases} M_{01} = \sum_{x_i} \sum_{y_i} y_i I(x_i, y_i) \\ M_{10} = \sum_{x_i} \sum_{y_i} x_i I(x_i, y_i) \end{cases} \quad (8)$$

The 2D orientation of the probability distribution is also easy to obtain by using the second moments during the CAMShift operation. Second moments are calculated as follows:

$$\begin{cases} M_{20} = \sum_{x_i} \sum_{y_i} x_i^2 I(x_i, y_i) \\ M_{02} = \sum_{x_i} \sum_{y_i} y_i^2 I(x_i, y_i) \\ M_{11} = \sum_{x_i} \sum_{y_i} x_i y_i I(x_i, y_i) \end{cases} \quad (9)$$

where  $I(x_i, y_i)$  is the pixel (probability) value at position  $(x_i, y_i)$  in the search window. The mean search window location (centroid)  $(x_0, y_0)$  is described as follows:

$$(x_0, y_0) = \left( \frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \right). \quad (10)$$

After obtaining the centroid coordinate  $(x_0, y_0)$ , the search window continues to move to the centroid until it converges to the target area. Then the object orientation (major axis), length  $l$ , and width  $w$  are calculated by Formula (11),

$$\begin{cases} l = \sqrt{\left( (a+c) + \sqrt{b^2 + (a-c)^2} \right) / 2} \\ w = \sqrt{\left( (a+c) - \sqrt{b^2 + (a-c)^2} \right) / 2}, \\ \theta = \frac{\arctan(b/(a-c))}{2} \end{cases} \quad (11)$$

where the intermediate variables  $a$ ,  $b$ , and  $c$  are calculated by the  $n$ -th moment of the search window, as follows:

$$\begin{cases} a = M_{20}/M_{00} - x_0^2 \\ b = 2(M_{11}/M_{00} - x_0 y_0) \\ c = M_{02}/M_{00} - y_0^2 \end{cases} \quad (12)$$

The target histogram model in Section 3.1 strengthens the role of the salient hue area in the target recognition, and can suppress the interference of the background hue in target recognition. Using the above histogram model to achieve target tracking based on the CAMShift algorithm can improve the performance of the tracking algorithm. The algorithm is described as follows:

Step 1. Initialize the initial center position  $(x_c, y_c)$  and size of the search window  $(W, H)$ .

Step 2. Calculate the color probability distribution in the search window of the  $t$ -th frame. If  $t$  is greater than the total number of frames, go to step 8, otherwise, proceed to the next steps.

Step 3. Execute the mean-shift algorithm and obtain the new size and position of the search window.

Step 4. Calculate the Bhattacharyya distance  $D$  between the candidate target and the target template and compare it with the threshold.

Step 5. If  $D$  is less than the threshold ( $T$ ), use the result obtained in Step 3 as the initial size and position of the next frame.

Step 6. If  $D$  is greater than the threshold ( $T$ ), the template is not updated.

Step 7.  $t+1 \rightarrow t$ , traverse the entire video sequence and return to Step 3.

#### 4. Result analysis and discussion

The proposed tracker is implemented in MATLAB R2018b on an Intel Core i7-8550U 2.0GHZ CPU with 8GB RAM. The effectiveness of the algorithm and long-term tracking performance is verified by comparing the proposed, traditional CAMShift [5], and the MFTA [16] algorithms on the OTB2015 [25]. OTB2015 is annotated with 11 attributes that cover various challenging factors, including scale variation (SV), illumination variation (IV), occlusion (OCC), motion blur (MB), deformation (DEF), fast motion (FM), out-of-plane rotation (OPR), out-of-view (OV), in-plane rotation (IPR), background clutters (BC), and low resolution (LR). The learning rate  $\beta$  and the threshold  $T$  are 0.5 and 0.6, respectively.

##### 4.1 Quantitative analysis

The algorithms are compared and analyzed from four aspects: center position error (CPE), distance precision rate (DPR), overlap rate (OR), and overlap success rate (OSR). The tracking algorithm may be more sensitive to the initial position given in the first frame, and starting at different positions, have increased impact on the tracking. After disrupting the initial state in time (different frames) and space (different target positions), temporal robustness evaluation (TRE) and spatial robustness evaluation (SRE) are obtained, respectively.

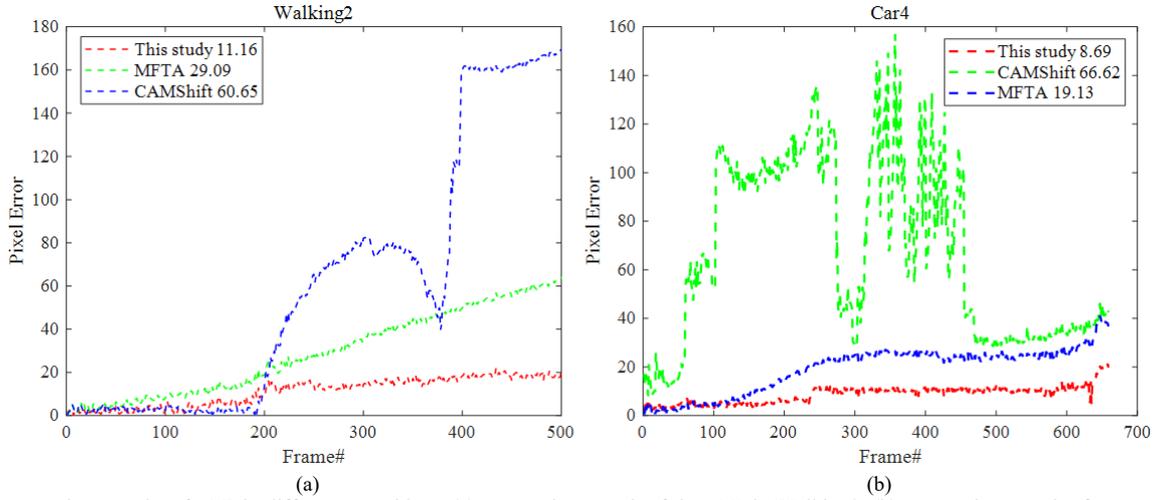
###### (1) Center position error

The center position error refers to the Euclidean distance between the estimated position  $(x', y')$  obtained by iteration and the true position  $(x, y)$ , which can be calculated with Formula (13):

$$D = \sqrt{(x-x')^2 + (y-y')^2} \quad (13)$$

As  $D$  decreases, the accuracy and stability of the algorithm increases. Fig. 1 shows the results of the center position error in the Walking2 and Car4. Results show that the traditional CAMShift algorithm easily leads to tracking loss when the background and foreground colors are similar. Despite its better tracking results compared with the other

models, the MFTA algorithm is inefficient due to the addition of feature extraction and fusion operations. The CPE of the proposed algorithm obtains a low value, with the maximum of only 20. The tracking window can converge to the target area and maintain better tracking results.

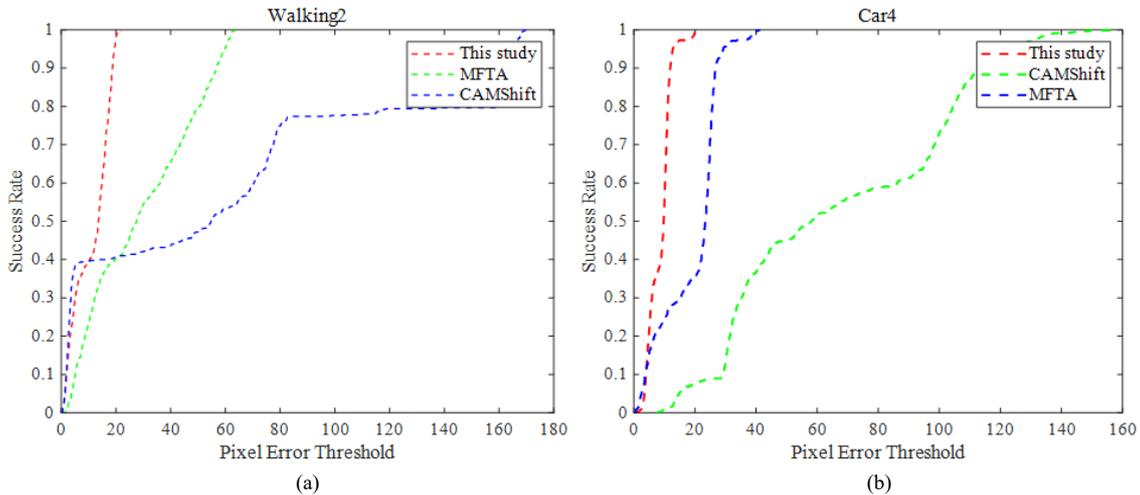


**Fig. 1.** Comparison results of CPE in different test videos. (a) Comparison result of the CPE in Walking2. (b) Comparison result of center position error in Car4.

(2) Distance precision rate

Distance precision rate represents the percentage of frames with smaller center location errors than a given threshold. With different thresholds, ratios differ, a curve can be obtained, and the threshold is set to 20 pixels. Fig. 2 shows the results of the distance precision rate in the Walking2 and Car4. The DPR of the MFTA algorithm in the Walking2 is only approximately 0.4. When the target is occluded, tracking fails because the model is not updated. The algorithm accuracy in Car4 increases as the threshold increases. The MFTA algorithm shows better robustness in

scenes with varying illumination. The distance accuracy of the traditional CAMShift algorithm is low, close to 0. When the threshold is set to 20, the distance precision rate of the proposed algorithm is close to 1. When the target is deformed, the background noise is large, illumination changes, and other complex effects occur, the proposed algorithm maintains a good tracking ability and stability under complex environments with deformation, background clutters, and illumination variation.



**Fig. 2.** Comparison of DPR of different algorithms. (a) Comparison result of the DPR in Walking2. (b) Comparison result of the DPR in Car4.

(3) Overlap rate

The overlap rate between the predicted bounding box  $S_p$  estimated by the tracking algorithm and the ground-truth bounding box  $S_G$  is calculated using Formula (14). As the overlap rate increases, the tracking success rate increases.  $\cap$  and  $\cup$  represent the intersection and union of these two bounding boxes, respectively.

$$S = \frac{|Area(S_p \cap S_G)|}{|Area(S_p \cup S_G)|} \quad (14)$$

Fig. 3 shows the comparison results of the tracking overlap rate of each frame in Walking2 and Car4. In the video sequence of Walking2, the target is blocked from frame 197, leading to a decrease in the overlap rate of the algorithm. The proposed algorithm can still successfully

track the target, but the overlap rate is not high. However, the comparison algorithms all show a tracking drift. In the video sequence of Car4, the overlap rate of the proposed and the MFTA algorithms have the lowest point affected by the varying illumination starting from frame 300. In subsequent

frames, the overlap rate gradually increases, and the algorithm shows good tracking recovery capabilities. The traditional CAMShift algorithm can only correctly track the target in the first 50 frames, and tracking drift occurs in subsequent frames.

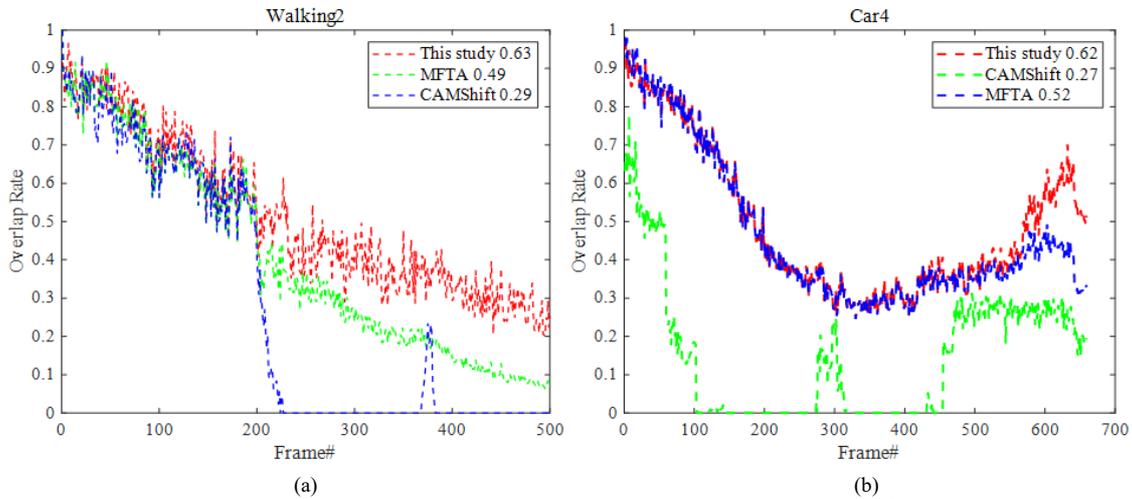


Fig. 3. Comparison of overlap rate of different algorithms. (a) Comparison result of the overlap rate in Walking2. (b) Comparison result of the overlap rate in Car4.

#### (4) Overlap success rate

Overlap success rate represents the percentage of frames with overlap rate that is greater than a given threshold. With different thresholds, ratios differ, a curve can be obtained, and the threshold is set to 0.3. Fig. 4 shows the comparison

results of the overlapping success rates of the different algorithms. According to the calculation rules of the tracking success rate, the overlap success rate of the proposed algorithm is the highest at 0.625, which is 17.6% higher than the second-ranking MFTA algorithm with 0.515.

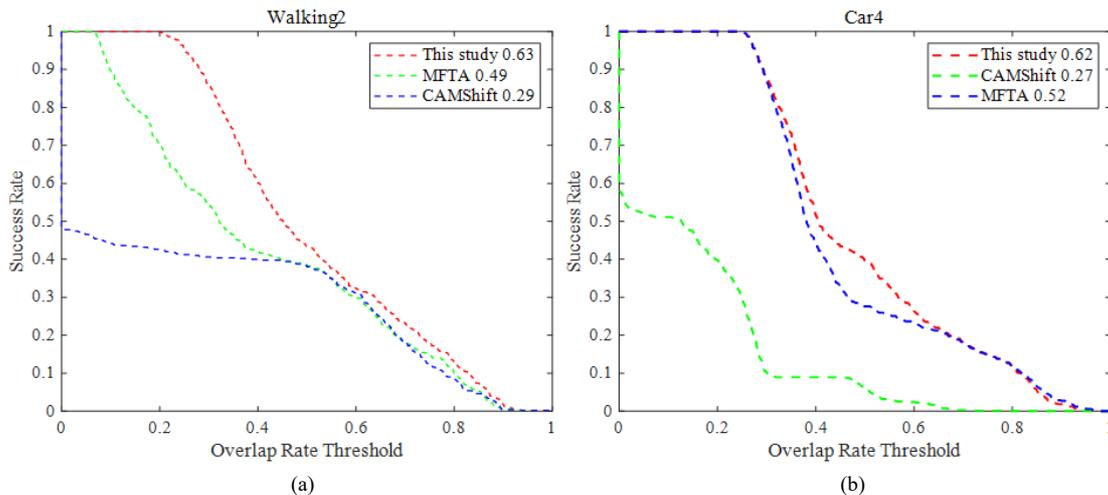


Fig. 4. Overlap success rates. (a) Overlap success rate in Walking2. (b) Overlap success rate in Car4.

#### 4.2 Qualitative analysis

Fig. 5 shows the comparison results between the proposed and the other two algorithms in two typical video sequences, namely, Walking2 and Car4. In Fig. 5(a), the target is occluded from frame 197, and the comparison algorithm named MFTA shows a tracking drift. However, the traditional CAMShift algorithm completely fails, and the tracking window stays at the position before the occlusion. The proposed algorithm can detect the target position despite the occlusion. Starting from frame 400, the overlap rate declines to 0.3 as the target moves away from the line of sight. In Fig. 5(b), the object in the video sequence of Car4

has a fast speed and large illumination variations. The proposed algorithm results in an error when the target changes lanes, but tracking continues. The MFTA algorithm shows varying degrees of drift and even leads to tracking failures. The traditional CAMShift algorithm shows a tracking drift from frame 100, and the overlap rate is 0. At approximately frame 500, the overlap rate increases to 0.3, which still does not satisfy the standard of correct tracking. The qualitative analysis results show the clear advantages of the proposed algorithm when the target has illumination variations, severe occlusion, and background interference.

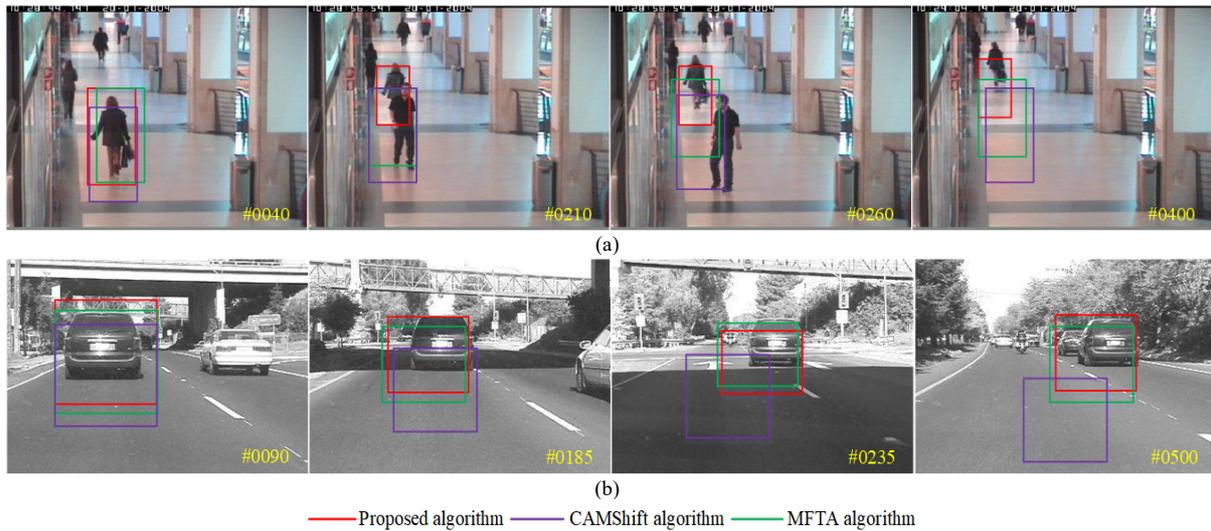


Fig. 5. Comparison results of algorithm visual effects. (a) The video sequence of Walking2. (b) The video sequence of Car4.

The robustness of the proposed algorithm is determined more intuitively using the images collected by the actual application video surveillance system. Fig. 6 shows the results of the target detection and tracking. The tracking target is a car passing the university gate, and a short-term

occlusion occurs. The foreground region extracted by the background model based on the improved Gaussian mixture model has low noise, and the target can be tracked correctly in actual application scenarios.



Fig. 6. Target detection and tracking results in actual application scenarios

## 5. Conclusions

The accuracy of the target-tracking algorithm in complex scenes was improved by extracting the adaptive background model to suppress the interference of the background color on the target color, and an enhanced target color model was obtained. The CAMShift algorithm was used to converge the area to be tracked, and the accuracy of target detection improved without the addition of auxiliary features. The following conclusions could be drawn:

- (1) Based on the improved Gaussian mixture model, a background model that changes with the scene is obtained, which reduces the error caused by background interference in target tracking.
- (2) The histogram of the hue component is established in the foreground and background of the target area. The target color model is obtained by suppressing the same hue as the background color in the tracking image, which improves the resolution ability of the target model and solves the target tracking problem in color interference scenes.
- (3) The target position is iteratively obtained by implementing CAMShift algorithm on the enhanced target color model.
- (4) The Bhattacharyya distance between the candidate target and the target template is used as the basis for

updating the target model to reduce the tracking error caused by deformation and to prevent tracking drift.

The proposed algorithm shows good tracking performance in challenging video scenes, such as those with occlusion, background clutter, and illumination variation. Thus, the proposed algorithm provides references for real-time monitoring of abnormal traffic conditions. Given the lack of sufficient testing in actual application scenarios, the histogram of oriented gradient features and the target color model are combined to improve the robustness of the algorithm in future studies.

## Acknowledgments

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), School-level Scientific Research Project of Xuzhou Institute of Technology(XKY20191070).

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



## References

1. Huang, K. Q., Chen, X. T., Kang, Y. F., Tan, T. N., "Intelligent visual surveillance: A review". *Chinese Journal of Computers*, 38(6), 2015, pp.1093-1118.
2. Chen, D. D., "Research on moving target tracking algorithm based on Camshift". Master thesis of Harbin Institute of Technology, China, 2017, pp.30-50.
3. Lu, H. C., Li, P. X., Wang, D., "Visual object tracking: a survey". *Pattern Recognition and Artificial Intelligence*, 31(1), 2018, pp.61-76.
4. 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.
5. Bradski, G. R., Clara, S., "Computer vision face tracking for use in a perceptual user interface". *Intel Technology Journal*, 2, 1998, pp.1-15.
6. Lan, J. H., Jiang, Y. L., Fan, G. L., Yu, D. Y., Zhang, Q., "Real-time automatic obstacle detection method for traffic surveillance in urban traffic". *Journal of Signal Processing Systems*, 82(3), 2016, pp.357-371.
7. Kim, Y., Han, W., Lee, Y. H., Kim, C. G., Kim, K. J., "Object tracking and recognition based on reliability assessment of learning in mobile environments". *Wireless Personal Communications*, 94(2), 2017, pp.267-282.
8. Li, F. L., Zhang, R. H., You, F., "Fast pedestrian detection and dynamic tracking for intelligent vehicles within V2V cooperative environment". *IET Image Processing*, 11(10), 2017, pp.833-840.
9. Bae, S. H., "Online multi-object tracking with visual and radar features". *IEEE Access*, 8, 2020, pp.90324-90339.
10. 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.
11. Howard, W., Nguang, S. K., Wen, J. W., "Robust video tracking algorithm: a multi-feature fusion approach". *IET Computer Vision*, 12(5), 2018, pp.640-650.
12. Guo, S. Q., Sun, X., "Indoor car tracking based on improved CAMShift algorithm". *Electronic Measurement Technology*, 42(14), 2019, pp.105-111.
13. Su, J., Gao, L. H., "Improved TLD tracking algorithm using frame difference method and correlation filter". *Computer Engineering and Design*, 41(6), 2020, pp.1694-1700.
14. Bankar, R., Salankar, S., "Improvement of head gesture recognition using Camshift based face tracking with UKF". In: *2019 9th International Conference on Emerging Trends in Engineering and Technology-Signal and Information Processing*, Nagpur, India: IEEE, 2019, pp.1-5.
15. Hayat, M. A., Yang, G. T., Iqbal, A., "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.
16. 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.
17. 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.
18. Guan, W. P., Liu, Z. P., Wen, S. S., Xie, H. Y., Zhang, X. J., "Visible light dynamic positioning method using improved Camshift-Kalman algorithm". *IEEE Photonics Journal*, 11(6), 2019, pp.1-22.
19. Pak, J. M., "Visual odometry particle filter for improving accuracy of visual object trackers". *Electronics Letters*, 56(17), 2020, pp.884-887.
20. Nie, J. S., Guo, X., "A target tracking method based on improved camshift algorithm". In: *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference*, Chongqing, China: IEEE, 2020, pp.254-258.
21. 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.
22. 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.
23. Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B. B. G., Geiger, A., Leibe, B., "MOTS: Multi-object tracking and segmentation". In: *IEEE Conference on Computer Vision and Pattern Recognition*, Los Angeles CA, United States: IEEE, 2019, pp.7934-7943.
24. Islam, M. M., Hu, G. Q., Liu, Q. B., Wang, D., Lyu, C. Z., "Correlation filter based moving object tracking with scale adaptation and online re-detection". *IEEE Access*, 6, 2018, pp.75244-75258.
25. Wu, Y., Lim, J., Yang, M-H., "Object tracking benchmark". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9), 2015, pp.1834-1848.