J. COMBIN. MATH. COMBIN. COMPUT. 127a (2025) 3887--3905

Journal of Combinatorial Mathematics and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Optimization strategies for real-time target detection and tracking in complex scenes

Jingjing Xia[1,✉]

[1] College of Information Engineering, Henan University of Animal Husbandry and Economy, Zhengzhou, Henan, 450046, China

ABSTRACT

In this paper, some important algorithms in the field of target detection and tracking are optimized. Firstly, Gaussian modeling is performed in the color space for the dynamic background, and the priority is set for ranking. Then introduce adaptive Gaussian component number mixing, adaptively change the weights, and adaptively change the number of mixed Gaussian components according to the pixel color change in the scene to improve the convergence speed of the complex scene. Finally, Kalman filtering and mean drift algorithm are combined to ensure the robustness of target tracking in complex scenes. The single-frame detection time, accuracy, and average tracking error of the algorithm designed in this paper are examined on the dataset, and it is found that the time consumed by the algorithm in this paper in the three scenarios is 242ms, 323ms, and 274ms, respectively, with the highest accuracy of 96.9% and the average tracking error of only 1.5 pixels. The optimization algorithm designed in this paper is able to adapt to the slight disturbance of the background scene and overcome the influence of noise and ambient lighting, which is a target detection and tracking algorithm with good robustness.

Keywords: image filtering, Gaussian component, Kalman filter, target detection

1. Introduction

With the development of artificial intelligence technology, many scholars are committed to making computers have a "visual system" similar to humans. Computer vision is a greatly challenging and important research direction in computer science. With the computer in the military, physical therapy, security and video surveillance and other fields have more and more extensive role, the computer has more and more influence on people, get more and more attention and concern, the application of this technology becomes more and more feasible. The computer through its own camera to the outside world image information acquisition, and will be converted into digital signals into the computer system, the computer system simulates the human brain using computer-related technology to obtain

✉ Corresponding author.
E-mail address: xiajj_9182@126.com (J. Xia).
Received 10 January 2024; Revised 15 March 2024; Accepted 25 December 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-220
© 2025 The Author(s). Published by Combinatorial Press. This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

the outside world information for analysis and processing, to achieve the perception of real-time scene [1-3].

Motion target detection and tracking technology, as one of the important research directions in the field of computer vision, is also the basis for a series of advanced semantic analysis and processing, such as later motion target recognition and analysis as well as anomalous event analysis, and thus has been widely studied by many scholars [4-7]. The detection and tracking technology of moving targets integrates the knowledge of image processing, pattern recognition, intelligent computing, machine vision and automatic control and other fields [8-9]. It is involved in visual navigation, traffic monitoring, visual coding, military guidance, and security monitoring [10]. At present, the research on motion target detection and tracking technology has made great achievements, and new techniques and algorithms are constantly emerging [11-12]. However, in practical applications, the complexity of natural environmental factors and the maneuverability of targets have interfered with the process of target detection and tracking, resulting in inefficiency and even incorrectness of detection and tracking [13-14].

Javed, S. et al. proposed a spatio-temporal structured sparse RPCA algorithm for the problem of spatio-temporal structure loss of moving objects during target detection and tracking by using the sparse components of the data as the eigenvectors of the spatial and temporal graph Laplacian and solving the constructed objective function by using linear alternating directional multiplier method [15]. Runz, M. et al. utilized the MaskFusion system for real-time recognition and tracking of multiple moving targets, and its ability to output maps with real-time perceptual and dynamic semantic features even in the face of complex scenarios, so the system can recognize and track multiple independently moving objects [16]. Cao, J. et al. proposed a target tracking algorithm in complex scenes and applied it to the unmanned driving field, which was evaluated through experimental tests and found that the algorithm possessed higher tracking accuracy and speed than other algorithms in various complex scenes and showed strong robustness and anti-jamming ability [17]. Meimetis, D. et al. showed that detecting the similarity and high density of targets as well as occlusion and viewpoint changes during target movement are the main challenges in the task of real-time multi-target detection and tracking, and proposed a real-time multi-target tracking framework based on an improved depth ranking algorithm, which detects whether a target is successfully tracked or not based on the previous frame of the image, and achieves a good result in the performance test [18]. Chen, C. et al. developed a lightweight spatio-temporal network to sense temporal and spatial information in complex scenes and consider the interaction between temporal and spatial information to significantly improve the accuracy of target detection as well as the efficiency of real-time recognition [19]. Luo, W. et al. combined deep neural networks with 3D detection to generate tracking and motion predictions by performing joint inference on data collected by 3D sensors, which is robust to handling occluded and sparse data [20].

This paper optimizes the traditional real-time target detection and tracking algorithm and designs a real-time target detection and tracking algorithm based on adaptive Gaussian mixing with improved mean drift. Firstly, the image in color space is transformed to space and components are combined to remove the shadows of the complex scene. Then adaptive Gaussian mixing is used to realize spatio-temporal features, adaptive learning control, adaptive variable weighting and adaptive model number selection, so that the K Gaussian components of each pixel can be adaptively adjusted according to the situation of the complex background and different lighting conditions. Finally, the mean drift algorithm is combined with Kalman filtering to eliminate the effect of structural perturbations on the system and realize the tracking of occluded targets. The algorithm is tested on a public dataset and compared with traditional target detection algorithms to check its optimization effect.

2. Method

2.1. Image Filtering and Gaussian Background Modeling

2.1.1. Color space conversion and image filtering:

1) Color space conversion. The color model consisting of the three elements of hue, saturation, and luminance is referred to as the HSV model or HSI model.

Hue H represents the color information, i.e., the position of the spectral color in which it is located; saturation S is a proportional value ranging from 0 to 1, which is expressed as the ratio between the purity of the selected color and the maximum purity of that color. When S=0, there is only grayscale; V represents the brightness of the color, ranging from 0 to 1, and there is no direct connection between it and the light intensity. There are two important features of the HSV model: firstly, the luminance component is separate from the chromaticity component, and the V component is independent of the color information of the image. Second, the H and S components are closely related to the way people perceive color. These features make the HSV model very suitable for image algorithms based on the processing and analysis of color perception characteristics by the human visual system.

The conversion formula between RGB space and HSV space is as follows:

$$\begin{cases} H = \arccos\dfrac{\frac{1}{2}((R-G)+(R-B))}{((R-G)^2+(R-B)(G-B))^{\frac{1}{2}}} & \text{if } B \le G \\ H = 2\pi - \arccos\dfrac{\frac{1}{2}((R-G)+(R-B))}{((R-G)^2+(R-B)(G-B))^{\frac{1}{2}}} & \text{if } B > G \\ S = \dfrac{Max(R,G,B)-Min(R,G,B)}{Max(R,G,B)} \\ V = \dfrac{Max(R,G,B)}{255} \end{cases} \tag{1}$$

2) Image Filtering. Image noise is a random interference signal to which an image is subjected during shooting or transmission. The suppression of these disturbing signals is called noise suppression of the image. Almost all images have more or less noise, which will affect the effectiveness of image processing.

The noise effect in an image system can be represented by adding a noise-free or very noisy image signal $f(x,y)$ and a random noise signal $n(x,y)$ [21]. The observed image signal is:

$$g(x,y) = f(x,y) + n(x,y) \tag{2}$$

The basic starting point of the denoising problem is to obtain an estimate of the original image from $g(x,y)$ that is as close as possible to $f(x,y)$. One approach to suppressing additive noise is to linearly filter the noisy signals in the image, which results in minimizing the mean-square error of averaging over the image and the interfering system.

The spatial domain image $f(x,y)$ is converted to the frequency domain by Fourier transform, and the original image is converted to $F(u,v)$. Then, the function $H(u,v)$ processes the frequency domain image $F(u,v)$, and transforms it to $G(u,v)$. The result of the processing is then inverted by the Fourier transform to obtain the spatial domain image $g(x,y)$. This process is the filtering process in the frequency domain, where, $H(u,v)$ is the filtering function. $F(u,v)$ and $H(u,v)$ are the Fourier transform forms of $f(x,y)$ and $h(x,y)$ respectively, which have the following convolution relationship:

$$f(x,y) * h(x,y) \Leftrightarrow F(u,v)H(u,v) \tag{3}$$

Gaussian filters are a class of linear smoothing filters in which the weights are selected based on the shape of the Gaussian function.Gaussian smoothing filters are effective in removing noise that obeys a normal distribution. The one-dimensional zero-mean Gaussian function is:

$$g(x)=e^{-\frac{x^2}{2\sigma^2}} \tag{4}$$

To filter an image with a Gaussian function, the larger the Tuning σ parameter, the wider the band of the Gaussian filter, and the better the smoothing effect of the image.

2.1.2. Mixed Gaussian background modeling. Detection of moving targets in video with background elimination is accomplished by using the current frame in the video and comparing it to a background reference model. The most commonly used probability distribution to describe the color distribution of background points is the Gaussian distribution.

Use $\eta(x,\mu,\Sigma)$ to denote the probability density function of a Gaussian distribution with mean μ and covariance matrix Σ. Since each pixel point is treated completely independently in the background model, the descriptions of the background model in this paper are all for the same pixel point if not otherwise specified. The single Gaussian distribution background model is applicable to the unimodal background situation, which establishes a model for the color distribution of each pixel point represented by a single Gaussian distribution $\eta(x,\mu_i,\Sigma_i)$, where the subscript t denotes time. Let the current color metric of a pixel point be X. If $\eta(x,\mu_t,\Sigma_t)\le T_p$ (here T_p is a probability threshold), the point is determined to be a foreground point, otherwise it is a background point.

When the background model cannot be described by a single Gaussian distribution, traditional Gaussian modeling methods cannot accurately represent the scene, resulting in false targets in target detection.

1) Establishment of the background model. Define its distribution model for a pixel (x_0,y_0) in a video frame, and let the set of values taken by that pixel up to the t moment be $\{X_1,\ldots,X_i\}=\{I(x_0,y_0,i):1\le i\le t\}$ where I is the video frame. If all historical values of this pixel are approximated by a K Gaussian model, then the probability of observing the current pixel value is

$$P(X,=x)=\sum_{i=1}^{K}\omega_{i,j}\eta(x,\mu_{i,j},\Sigma_{i,j}) \tag{5}$$

where n is the dimension of X, which in the case of a grayscale image is $n=1$. To determine the number of components for the model, Raja uses an approach that starts with a low-order Gaussian model to compute the a priori probability and selects the components by iteratively dividing them and evaluating the likelihood. In each iteration, the least likely component is found and divided into two components, and the likelihood is recalculated for the legitimate dataset until the prior probability of the legitimate dataset peaks. Considering the reduction of computational complexity, the covariance matrix is defined as $\sum_{i,i}=\sigma_i^2 I$.

2) Update of background model. In order to enhance the adaptability of the background model, the hybrid Gaussian model needs to be updated for each distribution model according to the actual situation. The update of the background model is to update its model parameters including the parameters of the Gaussian distribution itself and the weights of each Gaussian distribution.

The weights are updated:

$$\omega_{i,j}=(1-\alpha)\omega_{i,j-1}+\alpha M \tag{6}$$

Where α is the learning rate, if α is taken small, the ability to adapt to environmental changes is low, and can adapt to slow environmental changes, or, given enough time, the model can eventually adapt to environmental changes; if α is taken large, the ability to adapt to environmental changes is strong, but it is susceptible to noise and not stable enough.

Update of mean and variance:

For the unmatched Gaussian distribution, it is not updated, and for the matched i th Gaussian distribution as follows.

$$\mu_{i,j} = (1-\rho)\mu_{i,t-1} + \rho X_t \tag{7}$$

$$\sigma_{i,t}^2 = (1-\rho)\sigma_{i,t-1}^2 + \rho(X_t - \mu_{i,t})^T (X_t - \mu_{i,t}) \tag{8}$$

where $\rho = \alpha\eta(X_t | \mu_{i,j}, \sigma_{i,j})$.

α is the learning rate and $0<\alpha<1$, it reflects how fast or slow the background is updated, when the learning rate α is high, the background changes will be incorporated into the background model very quickly, when α is low, the background changes will be incorporated into the background model more slowly. M_t is the model matching operator, it only takes two values 0 and 1, it takes 1 when the new pixel value matches the model, otherwise it takes 0, and T is the threshold threshold.

$$\begin{cases} M_t = 1, if(X_t - \mu_t) > T\sigma \\ M_t = 0, else \end{cases} \tag{9}$$

More accurate automatic multi-person detection algorithms are needed for complex backgrounds and changing lighting conditions in crowded scenes. In this paper, we propose an automatic multiplayer detection method based on hybrid adaptive Gaussian mixture models that can effectively handle complex backgrounds and illumination variations. The hybrid algorithm proposed in this paper utilizes spatio-temporal features, adaptive learning control, adaptive variable weighting, and adaptive model number selection, so that the K Gaussian components of each pixel can be adaptively adjusted according to the situation of the complex bamboo scene and different lighting bars.

2.2. Multi-target detection algorithm based on adaptive Gaussian mixture modeling

2.2.1. Background Modeling and Motion Target Detection. In this paper, we propose a fusion of Gaussian Mixture Model (GMM) and Improved Adaptive Gaussian Component Number K Mixture Model (IAKGMM) algorithms, where observed pixels are modeled and updated by Gaussian Component Number K adaptive mixture model to deal with complex backgrounds as well as multiple people detection in crowded videos. The principle of the algorithm is shown in Fig. 1.

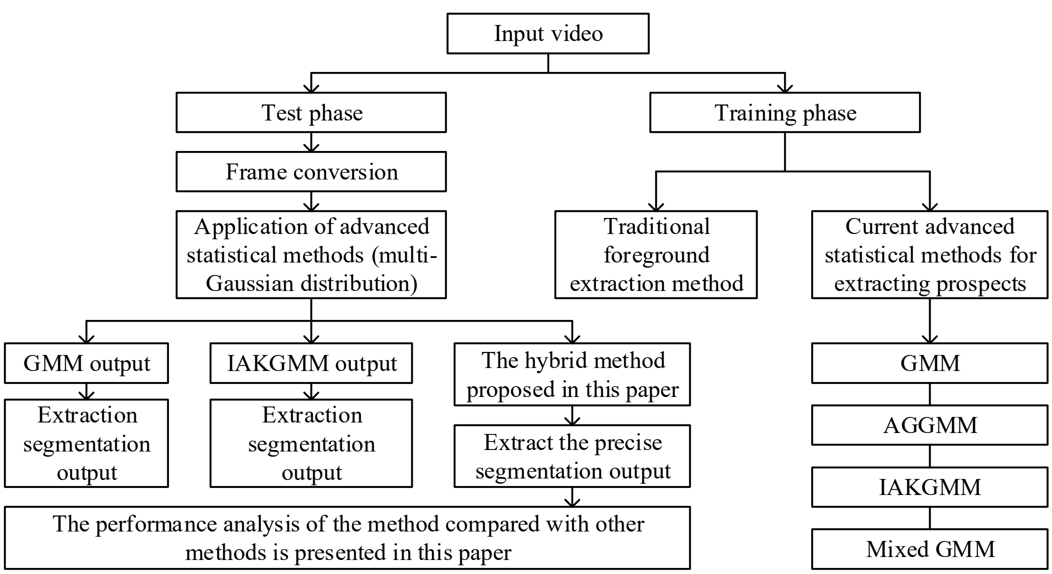


Fig. 1. Structural block diagram of the proposed algorithm

First, the historically observed pixels are modeled by K Gaussian distributions, and then the probabilities of these pixels are estimated. Pixel-level background segmentation involves a binary classification problem based on $P(BG|x)$, where x is the pixel value at the moment t and BG denotes the background class. The temporal distribution $P(x)$ of the hybrid model is denoted as:

$$P(x)=\sum_{m=1}^{K}P(G_m)P(x|G_m)=\sum_{m=1}^{K}\pi_m\cdot g(x;\mu_m;\sigma_m) \tag{10}$$

The posterior probabilities can be expressed as $P(G_m)$ and $P(x|G_m)$, with G_m being the mth Gaussian manifold though. The density estimate of $P(BG|x)$ is shown below:

$$P(BG|x)=\sum_{m=1}^{K}P(BG|G_m)P(G_m|x)=\frac{\sum_{m=1}^{K}P(x|G_m)P(G_m)P(BG|G_m)}{\sum_{m=1}^{K}P(x|G_m)P(G_m)} \tag{11}$$

Background subtraction determines whether a pixel belongs to the background (BG) or some foreground object (FG). According to the Bayesian decision rule, whether a pixel belongs to the background or foreground can be determined by their likelihood or probability ratio. To categorize the background and foreground pixels, the Bayesian decision coefficient R is shown in Equation (12):

$$R=\frac{P(BG|x)}{P(FG|x)}=\frac{P(x|BG)P(BG)}{P(x|FG)P(FG)} \tag{12}$$

By using the above decision rule, the observed image cord values are categorized as moving or static pixels. If R in Equation (12) is greater than 1, pixel x is decided as background; conversely pixel x is decided as foreground. The target detection can be achieved by classifying the background pixels from the observed values using the pixel value equations in the hybrid method proposed in this paper.

$$\{background, if (P(FG|x)) < b_{threshold} \tag{13}$$

$$\{foreground, if (P(FG|x)) > b_{threshold} \tag{14}$$

From equations (13) and (14), if the values of the observed pixels are lower than $b_{ehershold}$, these pixels are considered as background pixels, and the pixels in other cases are considered as foreground pixels. The initial value of the threshold $b_{fherthold}$ is set to 50, and the threshold value is adaptively changed according to the observed values of the pixels.

A time interval T is set, and the sample pixels observed at the t moment are denoted as $X_T=\{x^i,\ldots,x^{i-T}\}$ and the observed pixels are classified based on the history of previous pixels. For each new sample, the training dataset X_T is updated and re-estimated $P(x|X,BG)$. However, in the latest history sample, depending on the weight values of the Gaussian distribution, there may be some pixels belonging to the foreground objects and the rest of the pixels belonging to the background pixels, and the estimation should be denoted $P(x|X,BG+FG)$. The estimation model for the K components is defined as:

$$p(x|X_7,BG+FG)=\sum_{m=1}^{K}\pi_m g(x;\mu_m;\sigma_m^2;I) \tag{15}$$

where I is the unit matrix in the appropriate dimension. π_m denotes the mixing weights whose values are non-negative and add up to 1. Given a t moment new data sample x, the recursive update equation is:

$$\pi_m \leftarrow \pi_m+\alpha(o_m^{(t)}-\pi_m) \tag{16}$$

$$\mu_m \leftarrow \mu_m + o_m^t (\alpha / \pi_m) \delta_m \tag{17}$$

$$\sigma_m^2 \leftarrow \sigma_m^2 + o_m^t (\alpha / \pi_m)(\delta_m^T \delta_m - \sigma_m^2) \tag{18}$$

Among them, $\delta_m = x^{(t)} - \mu_m, o_m^{(t)} = \begin{cases} 1 & match \\ 0 & otherwise \end{cases}$.

The Marginal distance is used to determine whether a new sample is close to the Gaussian component. A sample is defined to be "close" to a Gaussian component if the Marginal Distance of the sample to the Gaussian component is less than three times the standard deviation. For ease of calculation, the distance from the sample to the mnd component is expressed as the square of the distance, denoted as $D_m^2(x^{(t)}) = \delta_m^T \delta_m / \sigma_m^2$. If there is no component close to the sample, a new component is created with the parameter set to $\pi_{M+1} = \alpha, \mu_{M+1} = x^{(l)}, \sigma_{M+1} = \sigma_0$, and σ_0 is the appropriate initial variance. If the maximum number of components has been reached, the component with the smallest weighting π_m is discarded.

Sort these M Gaussian distributions in descending order based on the value of π_m. A higher value means a higher weight, which increases the probability of determining that the pixel is background. The top B of these Gaussian components are taken out to approximate the background model.

$$p(x \mid X_T, BG) \sim \sum_{m=1}^{B} \pi_m g(x; \mu_m, \sigma_m^2 I) \tag{19}$$

If these components are arranged in descending order of weighting π_m, Eq. (20) can be obtained:

$$B = \arg\min_b \left(\sum_{m=1}^{b} \pi_m > (1 - c_f) \right) \tag{20}$$

The matching matrix A is constructed during the parameter selection process. it is a temporary matrix containing the history of pixel values during the learning process. The dimension of matrix A is set. The matrix A contains the weight values of the hybrid Gaussian model.

$$A = \begin{pmatrix} W_{1,1} & W_{1,2} & \cdots & W_{1,N} \\ W_{2,1} & W_{2,2} & \cdots & W_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ W_{M,N} & W_{M,N} & \cdots & W_{M,N} \end{pmatrix} \tag{21}$$

where the number of rows M, the number of columns N, and $W_{1,1}, W_{2,2}, \cdots\cdots, W_{M,N}$ denotes the weights of the mixed Gaussian.

Assume that $x^{(t)}$ is a foreground pixel and for each of the K components of the blended Gaussian, compute the following parameters;

$$\eta(x^{(t)}, \mu_{1,t}, \sum K, t) \tag{22}$$

2.2.2. Changing the number of Gaussian distributions in a mixture model. The number of Gaussian distributions, K, is an extremely important parameter for the hybrid Gaussian model because it directly affects the performance of the algorithm [22].

Suppose there are t data samples, each belonging to a component of the GMM. The number of samples belonging to the mnd Gaussian component is:

$$n_m = \sum_{i=1}^{i} o_m^{(i)} \tag{23}$$

The multinomial distribution assumed for $n_m - s$ is given its likelihood function $\Gamma = \prod_{m=1}^{M} \pi_m^{n_m}$. The sum of the mixture weights is constrained to 1, Lagrange multipliers are introduced λ and the maximum likelihood estimate (ML) is:

$$\frac{\partial}{\partial \pi_m}(\log \Gamma + \lambda(\sum_{m=1}^{M} \pi_m - 1)) = 0 \tag{24}$$

After removing λ, you can get:

$$\pi_m^{(t)} = \frac{n_m}{t} = \frac{1}{t}\sum_{i=1}^{t} o_m^{(i)} \tag{25}$$

The estimate of t sample is denoted as $\pi_m^{(t)}$, which can be reexpressed in recursive form, and it can be rewritten in recursive form as a function of the estimate $\pi_m^{(i-1)}$ of $t-l$ the sample and the ownership $o_*^{(t)}$ of the last sample, as follows.

$$\pi_m^{(t)} = \pi_m^{(t-1)} + 1/t(o_m^{(t)} - \pi_m^{(t-1)}) \tag{26}$$

The maximum a posteriori solution with the addition of the Delicacy prior is shown in equation (27):

$$\frac{\partial}{\partial \pi_m}(\log \Gamma + \log P + \lambda(\sum_{m=1}^{M} \pi_m - 1)) = 0 \tag{27}$$

where $P = \prod_{m=1}^{M} \pi_m^{-c}$, can be obtained:

$$\pi_m^{(i)} = \frac{1}{K}(\sum_{i=1}^{t} o_m^{(i)} - c) \tag{28}$$

where $K = \sum_{m}=1^{M}(\sum_{i=1}^{l} o_m^{(0)} - c) = t - Mc$, Eq. (28) can be rewritten as:

$$\pi_m^{(t)} = \frac{\Pi_m - c/t}{1 - Mc/t} \tag{29}$$

where $\Pi_m = \frac{1}{t}\sum_{i=1}^{t} o_m^{(0)}$ is the ML estimate and c/t denotes the bias from the a priori estimate. For larger data sets (larger t), the bias decreases. To obtain a small bias, a larger T can be chosen so that the bias is fixed at $c/t = c_T = c/T$. This means that the bias is all the same if samples with a data set of T are used. Using $c_T = c/T$ the recursive equation is obtained as:

$$\pi_m^{(t)} = \pi_m^{(t-1)} + 1/t(\frac{o_m^{(t)}}{1 - Mc_T} - \pi_m^{(t-1)}) - 1/t\frac{c_T}{1 - Mc_T} \tag{30}$$

Since only a small number of components M and c_r are usually expected to be small, assume $1 - Mc_r \approx 1$. As mentioned earlier, set $1/t$ to α to obtain the final modified adaptive update equation:

$$\pi_m \leftarrow \pi_m + \alpha(o_m^{(t)} - \pi_m) - \alpha c_T \tag{31}$$

2.2.3. Adaptive hybrid model parameter update. This method is applied to crowd videos in complex scenes and is able to obtain accurate multi-person detection results. It shows better results with crowds, swaying trees (dynamic background) and variable lighting. The realization steps are:

1) Initial $K = 1$, learning coefficients $\alpha = 0.01, b_i hreshold = 50, mean = \mu_m, \mathrm{var}iance = \sigma_\pi^2, co-\mathrm{var}iance = \sum K, t$, temporary matrix A, and π_m are Gaussian weights with standard deviation from 2 to 2.5.

2) Read in the video of the crowd scene, the number of frames is N and the number of Gaussian distributions is K.

3) Set the threshold value to start the learning process, x, y indicates the number of rows and columns in each frame. Set the time period as T and the observation moment as t.
4) Preprocess the noisy video and construct the temporary matrix A by starting the learning process of GMM in the given crowd video.
5) Calculate the mean $mear\mu_m$, variance $\mathrm{var}iance\sigma_m^2$ and then the covariance matrix $co-\mathrm{var}ianc\sum K, t$.
6) Using decision rule $R = \frac{P}{(BG/x^{(t)})P(BG)}P(FG/x^{(t)})P(FG)$, adaptively adjust the threshold value based on the observed pixels and select the larger $b_{threxhold}$ value to deal with the multinomial distribution.
7) In the matching process, determine whether the current pixel and the observed pixel match. If the match is successful, utilize IAKGMM and random update process.
8) For the polymorphic distribution, the threshold, learning rate, weights and the number of Gaussian components K are dynamically and adaptively updated.
9) The prior weights π_m, mean μ_m, and standard deviation σ_m of the Kth Gaussian component are adaptively adjusted.
10) If the current pixel does not match any of the K distributions, μ_m, σ_m of the Kth distribution is kept at its original value, and then, a new Gaussian distribution is added.
11) If the weights π_m are small, then the threshold $b_{threshold}$ changes as the weights are normalized.
12) In the hybrid approach, the GMM deals with gradual illumination changes and cluttered backgrounds. IAKGMM utilizes a stochastic pass-approximation procedure to update the parameters with the number of optimal hybrid components.

2.3. Improved mean drift algorithm based on Kalman filtering

Kalman filtering can make up for the loss of tracking in severe occlusion and the inability to track fast-moving targets that occurs when the mean-drift algorithm tracks targets.

2.3.1. System modeling. In the field of target tracking, the selection of target features, the representation model of the target, the similarity measure, and the tracking algorithm, each of which has an impact on the tracking effectiveness.

The histogram of the kernel function obtained by mean drift algorithm at the convergence point is referred to as the observation model. The model obtained by Kalman filtering the observed model and the target model after optimization is called the candidate model. In the kernel function histogram, since each component is relatively independent, we filter and update each component separately.

For a dynamic system with linear, Gaussian nature, let k denote the sequence number state equation of the frame as:

$$x_{k+1} = F_{k+1,k}x_k + w_k \tag{32}$$

In equation (32), x_k denotes the system state vector at moment k, and $F_{k+1,k}$ denotes the system state transfer matrix.

w_k denotes the state noise, Gaussian white noise with mean and covariance, respectively:

$$E[w_k] = 0 \quad Cov[w_n w_k^T] = \begin{cases} Q_k & n = k \\ 0 & n \neq k \end{cases} \tag{33}$$

The observation equation for the system is:

$$z_{k+1} = H_{k+1}x_{k+1} + v_{k+1} \tag{34}$$

In the above equation z_k denotes the measurement vector of the system and H_{k+1} denotes the system observation matrix. v_k denotes the system observation noise, Gaussian white noise, is independent of the state noise w_k. Its mean and covariance are respectively:

$$E[v_k]=0 \quad Cov[v_n v_k^T]=\begin{cases} R_k & n=k \\ 0 & n\neq k \end{cases} \tag{35}$$

2.3.2. Updating of observational models. For two consecutive frames $k-1, k$, for each component u, a one-step prediction equation and a measurement update equation are given according to Kalman filter theory.

The one-step prediction equation is as follows:

$$x_u^{k|k-1}=F^k X_u^{k-1|k-1} \tag{36}$$

In Eq. (36), X_u^{kk-1} is the current result predicted using the previous state, and $X_u^{k-1|k-1}$ is the optimal result of the previous state. This process, i.e., the system is updated. Using $P_u^k \mid k-1$ to denote the covariance of $X_u^{k|k-1}$ and $P_u^{k-1|k-1}$ to denote the covariance of $X_u^{k-1|k-1}$, then:

$$p_u^{k|k-1}=F^k P_u^{k-1|k-1} A^{k^T}+Q^{k-1} \tag{37}$$

The measurement update equation is as follows:

After finding the prediction result of the current state according to the one-step prediction equation, the measurement result of the current state is collected.

Combining the measurement results and prediction results, we can get the optimized estimate of the current state X_u^{kk}:

Where Kg_k is the Kalman gain:

$$X_u^{k|k}=X_u^{k|k-1}+Kg_k(z_u^k-H^k X_u^{k|k-1}) \tag{38}$$

$$Kg_k=\frac{P_u^{kk-1}H^{k^T}}{H_k P_u^{k|k-1}H_k^T+R^k} \tag{39}$$

P_u^{kk} is the covariance of the optimized estimate X_u^{kk} in the current state, then

$$P_u^{k|k}=P_u^{k|k-1}-Kg_k H^k P_u^{k|k-1} \tag{40}$$

2.3.3. Kalman Optimal Filtering. Equation (32)(33) models a linear discrete system, since a linear discrete system can be considered as a linear continuous system when the sampling period tends to zero infinitely [23]. Therefore, we can realize the conversion of linear discrete system Kalman filter to linear continuous system Kalman filter by the following method:

$$\dot{x}(t)=A(t)x(t)+G(t)w(t) \tag{41}$$

$$z(t)=H(t)x(t)+v(t) \tag{42}$$

The equivalent mathematical description between Eqs. (32)(34) and Eqs. (41)(42) can be expressed by the following relationship:

$$x_k=x(t+\Delta t) \tag{43}$$

$$F_{k+1,k}=F(t+\Delta t,t) \tag{44}$$

$$z_k=z(t+\Delta t) \tag{45}$$

$$H_k=H(t+\Delta t) \tag{46}$$

$$P_{k|k-1} = P(t+\Delta t, t) \tag{47}$$

$$P_{k|k} = P(t+\Delta t, t+\Delta t) \tag{48}$$

$$Kg_k = Kg(t+\Delta t) \tag{49}$$

$$Q_{k-1} = Q(t)/\Delta t \tag{50}$$

$$R_k = R(t+\Delta t)/\Delta t \tag{51}$$

Taking into account the Kalman filter formulation for discrete systems, we get

$$x(t+\Delta t) = F(t+\Delta t, t)x(t) + Kg(t+\Delta t)\left[z(t+\Delta t) - H(t+\Delta t)F(t+\Delta t, t)x(t)\right] \tag{52}$$

$$Kg(t+\Delta t) = \frac{P(t+\Delta t, t)H^T(t+\Delta t)}{H(t+\Delta t)P(t+\Delta t, t)H^T(t+\Delta t) + R(t+\Delta t)/\Delta t} \tag{53}$$

$$P(t+\Delta t, t+\Delta t) = P(t+\Delta t, t) - Kg(t+\Delta t)H(t+\Delta t)P(t+\Delta t, t) \tag{54}$$

As the discrete system model is transformed into a continuous system model, it can be approximated by the following:

$$F(t+\Delta t, t) = I + A(t)\Delta t \tag{55}$$

where $I = F(t,t)$, substituting the above equation into Eq. (44) yields

$$\begin{aligned} x(t+\Delta t) &= \left[I + A(t)\Delta t\right]x(t) \\ &+ Kg(t+\Delta t)\left[z(t+\Delta t) - H(t+\Delta t)[I + A(t)\Delta t]x(t)\right] \end{aligned} \tag{56}$$

Further changes as:

$$\frac{x(t+\Delta t) - x(t)}{\Delta t} = A(t)x(t) + \frac{Kg(t+\Delta t)}{\Delta t}[z(t+\Delta t) - H(t+\Delta t)[I + A(t)\Delta t]x(t)] \tag{57}$$

The above equation is obtained by taking the limit when $\Delta t \to 0$:

$$\dot{x}(t) = A(t)x(t) + Kg(t)\left[z(t) - H(t)x(t)\right] \tag{58}$$

Equation (58) is the Kalman optimal filtering equation for a linear continuous system.

2.3.4. Kalman Filtering Combined with Improved Mean Drift Algorithm. In order to solve the tracking problem in the case where the target is partially or fully occluded, an algorithm that combines adaptive Kalman filtering with a modified mean drift algorithm is used in this section to realize the target tracking in the case of occlusion.

Let set $\{X_b^k\}_{b=1,2,\ldots,m}$ be the candidate model of the tracking target in frame k. Calculate the probability density estimate of the $u(u = 1,\ldots,m)$ rd feature in the feature space of the target region centered at this position as:

$$\hat{q}_u = C\sum_{i=1}^{n} k\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)\delta\left[b(x_i) - u\right] \tag{59}$$

Set $\{Y_b^k\}_{b=1,2,\ldots,n}$ is the observation model of the tracked target for frame k, and the probability density estimate of the $u(u = 1,\ldots,m)$ rd feature of the target feature space is computed as:

$$\hat{p}_u(y) = C_h\sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right)\delta\left[b(x_i) - u\right] \tag{60}$$

The matching process is the process of finding the maximum value of the similarity function, and in each iteration of mean drift, the center position of the target region moves from y_0^k to a new position y_1^k:

$$y_1^k = \frac{\sum_{i=1}^{n_k} x_i \omega_i g\left(\left\|\frac{y_0^k - x_i^k}{h}\right\|^2\right)}{\sum_{i=1}^{n_k} \omega_i g\left(\left\|\frac{y_0^k - x_i^k}{h}\right\|^2\right)} \tag{61}$$

In Equation (61), ω_i is the weighting function expressed as follows:

$$\omega_i = \sum_{u=1}^{m} \delta\left[b(x_i) - u\right]\sqrt{\frac{q_u^{k-1}}{q_u^k}} \tag{62}$$

After obtaining y_1^k by computation, the distance between the old and new coordinates y_0^k and y_1^k is computed $\left\|y_1^k - y_0^k\right\|$. If $\left\|y_1^k - y_0^k\right\| < \varepsilon$ (ε is a very small threshold), it is read into the next frame, otherwise it continues to compute the histogram of the kernel function of the candidate model centered on the new coordinates y_1^k.

3. Results and Discussion

3.1. Comparative Experiments on Performance Testing of Target Detection Algorithms

Three video test sets of different scenes from the online public database are selected for simulation experiments to evaluate the performance of the target detection algorithm through the accuracy and false detection rate of the motion target detection. This section of the simulation experiment is completed under the Windows 10 operating system by configuring OpenCV3.3.0 with VS2017 simulation software.

The test videos V1, V2, and V3 are three video sequences selected from the online public dataset. The traditional hybrid Gaussian algorithm, the traditional three-frame difference method, the classical ViBe algorithm, the traditional mean drift algorithm, and the algorithm designed in this paper were used to detect the moving targets in the three scenes, respectively.

The detection results of frame 165 of video V1, frame 82 of video V2 and frame 39 of video V3 are selected for comparison.

The single-frame detection times of the five algorithms in the three scenes are shown in Table 1.

The time consumption of this paper's algorithm in the three scenarios is 242ms, 323ms and 274ms, respectively, and the time used for single-frame detection is higher compared to that of the traditional three-frame difference algorithm due to the existence of the establishment of Gaussian mixture model and the background updating, but the single-frame detection of the improved algorithm in this paper is much lower than that used by the traditional hybrid Gaussian algorithm, and it is much faster compared to that of the classical Vi Be algorithm and the traditional mean drift algorithm, the processing speed is faster, and the algorithm has a complete detection result in three different experimental scenarios while having a high computational efficiency, which ensures the algorithm's real-time performance.

Table 1. The single frame detection of five detection algorithms is compared(ms)

Algorithm	Scenario 1	Scenario 2	Scenario 3
Traditional hybrid gaussian algorithm	637	697	667

Traditional three frame difference algorithm	77	83	74
Classic vibe algorithm	385	425	386
Traditional mean drift algorithm	297	368	332
This algorithm	242	323	274

In order to quantitatively analyze the accuracy of the improved algorithms for motion target detection in this chapter, two commonly used judging criteria: accuracy DR and false detection rate FAR are used to objectively evaluate the detection performance of different algorithms.

In this paper, the completeness of target detection is measured by the accuracy DR in the judgment index, and the larger the calculated DR value is, the higher the completeness of target detection by the detection algorithms used, and vice versa, the lower the completeness is. The accuracy curves of five different algorithms for detecting moving targets in scene 1 are shown in Fig. 2. The accuracy curves of the improved algorithms of this paper for target detection in three different experimental scenes are shown in Fig. 3.

From the accuracy curve, it can be seen that the improved detection algorithm in this paper has the best overall performance compared with the other four algorithms, which can acquire the motion target more completely, and the target detection accuracy can be maintained at a high level in different scenes, with strong adaptive ability.

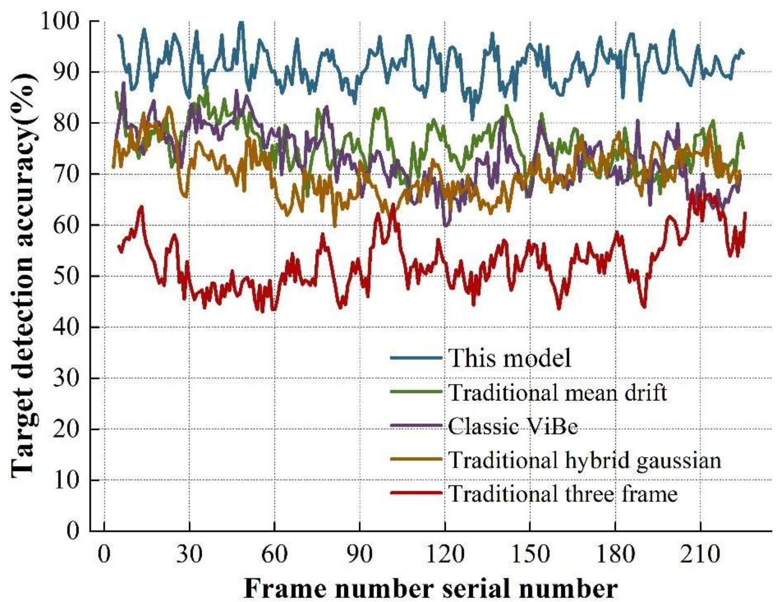


Fig. 2. The DR Curve of five different algorithms

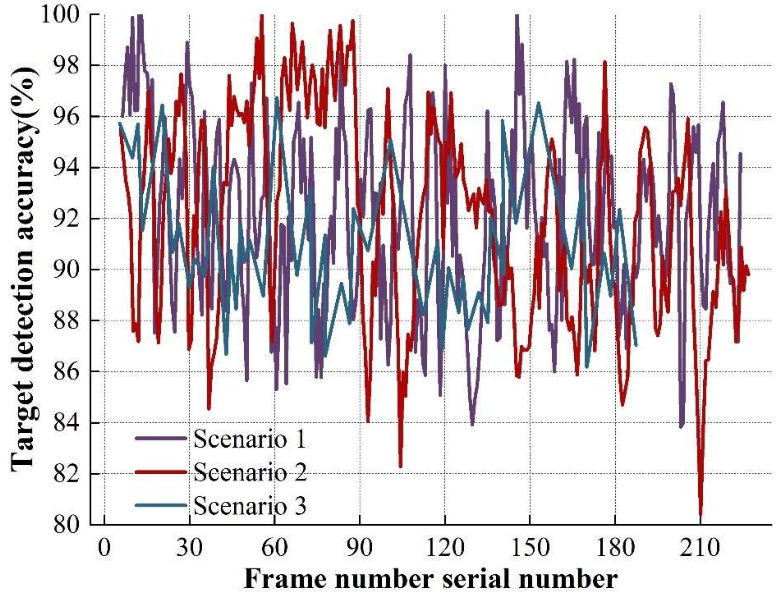


Fig. 3. The DR Curve of this article in three scenarios

In addition, for the interference of the above five algorithms in three different scenarios, the average accuracy and average false detection rate of all frames are calculated as shown in Tables 2 and 3, respectively.

From the data in the table, it can be seen that in terms of target detection accuracy, the accuracy rate of the interframe difference method is the lowest, the accuracy rate of the traditional mean drift algorithm is relatively improved, and the accuracy rate of this paper's algorithm reaches the highest of 96.9%, which greatly improves the completeness of the target detection; in terms of the target detection misdetection rate, the misdetection rate of the traditional hybrid Gaussian algorithm is the highest, the traditional mean drift algorithm is reduced relative to it, and the misdetection rate of this paper's algorithm is the lowest among the five detection algorithms. The false detection rate of this paper's algorithm is the lowest among the five detection algorithms, which is 8.4%, 3.5%, and 2.6% in the three scenarios, respectively, and possesses obvious detection advantages.

Table 2. The accuracy of five detection algorithms is compared

Algorithm	Scenario 1	Scenario 2	Scenario 3
Traditional hybrid gaussian algorithm	71.30%	70.70%	88.80%
Traditional three frame difference algorithm	52.50%	59.70%	40.90%
Classic vibe algorithm	74.60%	69.90%	89.70%
Traditional mean drift algorithm	85.50%	84.00%	90.50%
This algorithm	92.20%	94.70%	96.90%

Table 3. The error rate of five detection algorithms is compared

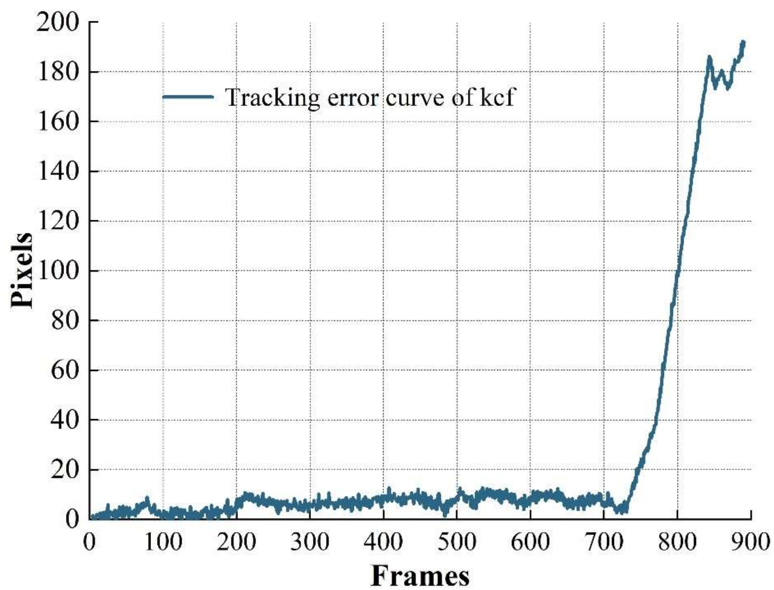
Algorithm	Scenario 1	Scenario 2	Scenario 3
Traditional hybrid gaussian algorithm	22.60%	53.40%	48.90%
Traditional three frame difference algorithm	15.10%	4.50%	8.50%
Classic vibe algorithm	15.80%	25.00%	42.60%

Traditional mean drift algorithm	18.10%	30.00%	24.10%
This algorithm	8.40%	3.50%	2.60%

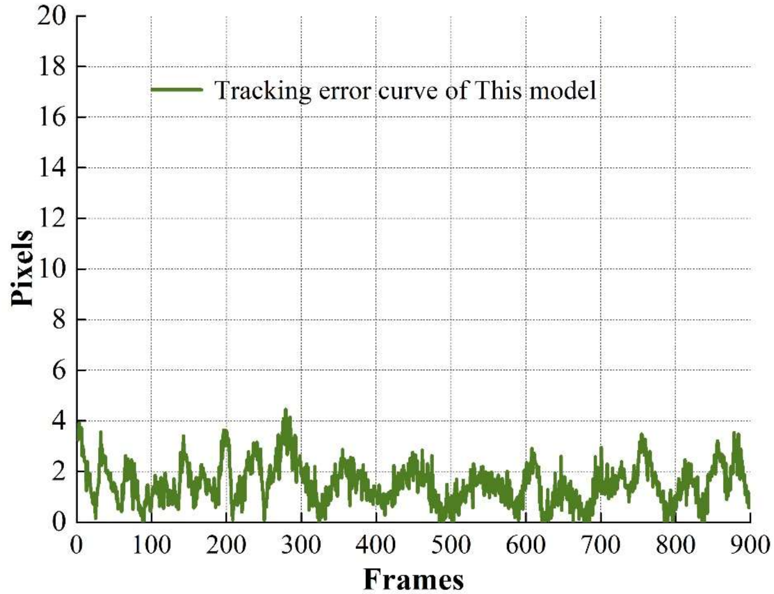
3.2. Tracking performance comparison experiment

In order to demonstrate the effect of the improved tracking algorithm more intuitively, the video image sequence Car1 on vehicles in Visual Tracker Benchmark platform is selected for experiments. The tracking error is utilized to quantitatively evaluate the tracking algorithm, and the tracking error of one frame is obtained every 4 frames, and the results are shown in Fig. 4. Where (a) is the tracking error curve of the kernel correlation filtering algorithm and (b) is the tracking error curve of the algorithm designed in this paper.

The tracking effect of the target vehicle is relatively good until frame 730, but after driving into the shadow area, the tracking error of the kernel correlation filter tracking algorithm increases significantly, because its tracking window cannot change the size and is affected by the shadow noise. The tracking algorithm in this paper keeps good tracking effect, the average tracking speed can reach 32 frames per second, and the average tracking error is only 1.5 pixels.



(a) Tracking error curve of KCF



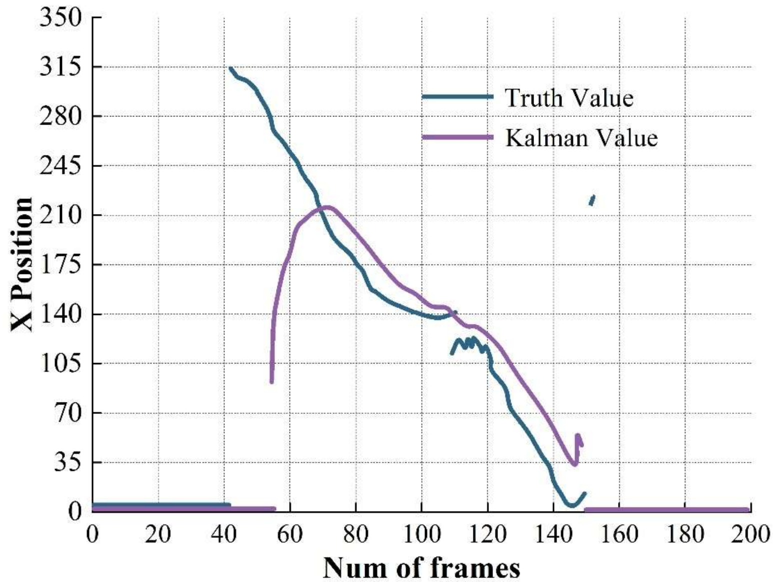
(a) Tracking error curve of This model

Fig. 4. Tracking error comparison

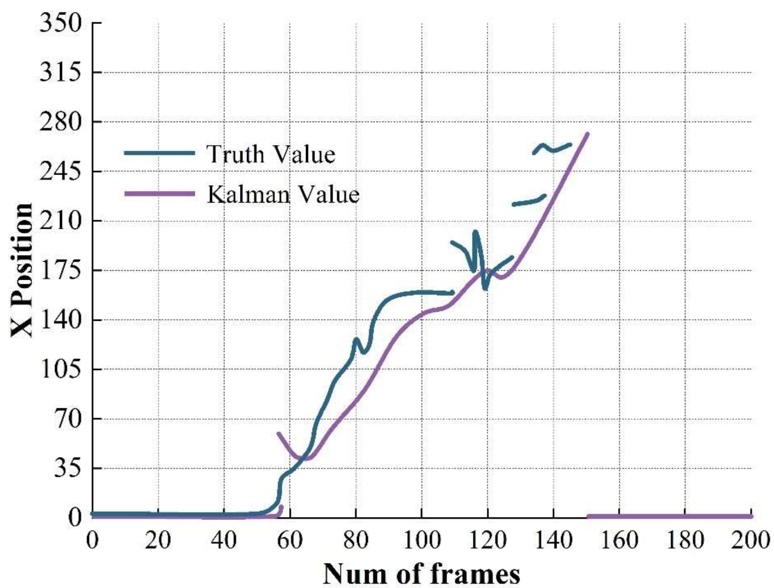
3.3. Simulation of multi-motion target detection and tracking in complex scenes

The CCD camera is used to collect image sequences of students' daily life in the school for multi-motion target detection and tracking experiments, the video sequences of the two students appear and are walking at nearly constant speed, in the video between 90 frames to 115 frames, the two motion targets meet basically stationary conversation, 120 frames after the two people cross walk away and appear to be blocking the phenomenon of a short period of time after the blocking of their respective walk away until they are out of the video surveillance. The video format is 720*1080.

Figure 5 shows the detection and tracking tests for new moving-in targets, disappearing targets and brief occlusion between targets. The multi-targets are predicted and the center-of-mass coordinate information of the detected objects is extracted by Kalman filtering, and a rectangular box is used to represent the position of the predicted center-of-mass. In Fig. 5(a), the improved algorithm is able to detect and track the position information of the targets in a timely manner for the right target A, which is moving at a uniform speed, and the left target B, which is about to enter into the video surveillance. In Fig. 5(b), when target B is occluded by A for a period of time, the target tracking rectangle box can still accurately capture the target's motion trajectory. Therefore, for target detection and tracking in multiple motion situations, the detection algorithm designed in this paper can accurately extract the foreground motion target, and the Kalman filter can effectively predict the trajectory of the motion target.



(a) Target A



(b) Target B

Fig. 5. The target A and B Kalman prediction tracking

4. Conclusion

In this paper, a real-time target detection and tracking algorithm based on adaptive Gaussian mixing with improved mean drift is designed to improve the quality of target detection and tracking in the case of complex scenes. The proposed method is experimented in a public video test set, and the accuracy of this paper's algorithm is as high as 96.9%, which is more complete than other algorithms in detecting moving targets, and has strong adaptability to complex environments while ensuring real-time performance. The tracking window of the algorithm can change with the size of the target more accurately, and can realize normal tracking when the target is covered and slightly rotated, and the test results are basically in line with the actual motion trajectory of the detected object, which verifies the robustness and real-time performance of this paper's method for target tracking in complex road scenes.

Funding

Henan Province science and technology breakthrough project: Research on the operation image object detection of plant protection UAV based on deep learning (242102110358).

References

[1] Simon, M., Amende, K., Kraus, A., Honer, J., Samann, T., Kaulbersch, H., ... & Michael Gross, H. (2019). Complexer-yolo: Real-time 3d object detection and tracking on semantic point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (pp. 0-0).

[2] Abba, S., Bizi, A. M., Lee, J. A., Bakouri, S., & Crespo, M. L. (2024). Real-time object detection, tracking, and monitoring framework for security surveillance systems. Heliyon, 10(15).

[3] Mauri, A., Khemmar, R., Decoux, B., Ragot, N., Rossi, R., Trabelsi, R., ... & Savatier, X. (2020). Deep learning for real-time 3D multi-object detection, localisation, and tracking: Application to smart mobility. Sensors, 20(2), 532.

[4] Pal, S. K., Pramanik, A., Maiti, J., & Mitra, P. (2021). Deep learning in multi-object detection and tracking: state of the art. Applied Intelligence, 51, 6400-6429.

[5] Kalake, L., Wan, W., & Hou, L. (2021). Analysis based on recent deep learning approaches applied in real-time multi-object tracking: a review. IEEE Access, 9, 32650-32671.

[6] Wang, K., Zhou, T., Li, X., & Ren, F. (2022). Performance and challenges of 3D object detection methods in complex scenes for autonomous driving. IEEE Transactions on Intelligent Vehicles, 8(2), 1699-1716.

[7] Liu, S., Wang, S., Liu, X., Lin, C. T., & Lv, Z. (2020). Fuzzy detection aided real-time and robust visual tracking under complex environments. IEEE Transactions on Fuzzy Systems, 29(1), 90-102.

[8] Wang, Q., Tang, S., Zhai, D., & Hu, X. (2018). Salience based object tracking in complex scenes. Neurocomputing, 314, 132-142.

[9] Hu, W. C., Chen, C. H., Chen, T. Y., Huang, D. Y., & Wu, Z. C. (2015). Moving object detection and tracking from video captured by moving camera. Journal of Visual Communication and Image Representation, 30, 164-180.

[10] Elhoseny, M. (2020). Multi-object detection and tracking (MODT) machine learning model for real-time video surveillance systems. Circuits, Systems, and Signal Processing, 39(2), 611-630.

[11] Sultana, M., Mahmood, A., & Jung, S. K. (2020). Unsupervised moving object detection in complex scenes using adversarial regularizations. IEEE Transactions on Multimedia, 23, 2005-2018.

[12] Singh, D. K., & Kushwaha, D. S. (2016). Tracking movements of humans in a real-time surveillance scene. In Proceedings of Fifth International Conference on Soft Computing for Problem Solving: SocProS 2015, Volume 2 (pp. 491-500). Springer Singapore.

[13] Hong, X., Cui, B., Chen, W., Rao, Y., & Chen, Y. (2022). Research on multi-ship target detection and tracking method based on camera in complex scenes. Journal of Marine Science and Engineering, 10(7), 978.

[14] Goyal, K., & Singhai, J. (2018). Texture-based self-adaptive moving object detection technique for complex scenes. Computers & Electrical Engineering, 70, 275-283.

[15] Javed, S., Mahmood, A., Al-Maadeed, S., Bouwmans, T., & Jung, S. K. (2018). Moving object detection in complex scene using spatiotemporal structured-sparse RPCA. IEEE Transactions on Image Processing, 28(2), 1007-1022.

[16] Runz, M., Buffier, M., & Agapito, L. (2018, October). Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In 2018 IEEE International Symposium on Mixed and

Augmented Reality (ISMAR) (pp. 10-20). IEEE.

[17] Cao, J., Song, C., Song, S., Xiao, F., Zhang, X., Liu, Z., & Ang Jr, M. H. (2021). Robust object tracking algorithm for autonomous vehicles in complex scenes. Remote Sensing, 13(16), 3234.

[18] Meimetis, D., Daramouskas, I., Perikos, I., & Hatzilygeroudis, I. (2023). Real-time multiple object tracking using deep learning methods. Neural Computing and Applications, 35(1), 89-118.

[19] Chen, C., Wang, G., Peng, C., Fang, Y., Zhang, D., & Qin, H. (2021). Exploring rich and efficient spatial temporal interactions for real-time video salient object detection. IEEE Transactions on Image Processing, 30, 3995-4007.

[20] Luo, W., Yang, B., & Urtasun, R. (2018). Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 3569-3577).

[21] Kunal Pradhan & Swarnajyoti Patra. (2024). A semantic edge-aware parameter efficient image filtering technique. Computers & Graphics104068-104068.

[22] Simone Maxand. (2020). Identification of independent structural shocks in the presence of multiple Gaussian components. Econometrics and Statistics(prepublish),55-68.

[23] E. Mohajeri,Seid H. Pourtakdoust & F. Pazooki.(2025).A neural-metaheuristic kalman filter for moving microburst wind shear identification.Journal of the Franklin Institute(1),107390-107390.