Journal of Computers Vol. 31 No. 2, 2020, pp. 264-276
doi:10.3966/199115992020043102022

Moving Target Detection and Tracking Based on Improved Mean Shift Algorithm[1]

Lin Zhang[1], Xiao-ping Li[2*], Fan-bo Zhang[3], Xu-long Ren[2]

[1] Beijing University of Civil Engineering and Architecture, Beijing, China

[2] Beijing Institute of Technology, Beijing, China

[3] Dalian Maritime University, Dalian, China
zhanglin@bucea.edu.cn

Received 20 January 2020; Revised 20 February 2020; Accepted 30 March 2020

Abstract. Based on the basic principle of Mean Shift algorithm, this paper proposes an improved target detection and tracking method based on mixture gauss model and Mean Shift algorithm, aiming at the complex background problems such as occlusion, shadow, illumination change, etc. The method uses the color feature in YCbCr color space as the target feature, uses the weighted operation of background and target to eliminate the interference of environmental noise, and highlights the effective information of the target itself. In the process of tracking, the target template is constantly updated to keep as consistent as possible with the target state, so as to achieve accurate, real-time and stable tracking of the target in the video stream. The experimental results show that the improved algorithm can effectively reduce the number of iterations and has a good tracking effect.

Keywords: Mean Shift algorithm, mixed gaussian model, template updating, video target tracking

1 Introduction

Moving object detection and tracking is an active branch in the field of image processing and computer vision. By detecting, extracting and tracking moving objects in image sequences, we can obtain moving parameters of moving objects, such as target centroid position, speed, acceleration, etc., which can be widely used in security defense, traffic coordination, positioning navigation, human-computer interaction and other fields [1]. Due to the complexity of the real world environment and the thousands of scene changes, many disturbing factors and challenges are brought to target detection and tracking. The main problems include:

(1) Complex background interference: in the tracking process, most of the background has noise, fragments and objects similar to the target, which have serious interference on the target detection, and the processing results will directly affect the accuracy of the subsequent target tracking [2].

(2) Multitarget problem: there are multiple objects in complex scene at the same time, which may cause interference and occlusion between the tracking object and other objects. However, it is very difficult to model and solve the background of multitarget occlusion, and the problem of data cascade must be considered, otherwise, it is easy to cause errors when matching the tracking result with each object [3].

(3) Occlusion problem: in the process of target tracking, because the size, occurrence time, location and occlusion time of the occluded target are unpredictable random phenomena, it brings great difficulties to target tracking [4].

(4) Target scale change: as the distance between the target and the video acquisition device changes,

[1] Supported by Beijing Key Laboratory of Intelligent Processing for Building Big Data.
* Corresponding Author

the target's own scale changes at the same time. If the tracking frame does not change after the scale changes, the extracted features will contain background information, and the recognition accuracy will be greatly reduced [5].

In view of the above problems, a large number of studies put forward different algorithms. In reference [6], a moving target detection and tracking algorithm based on Gaussian mixture model is proposed, which solves the problem of light mutation sensitivity in the traditional algorithm. By adjusting the number of scenes using Gaussian model, the adaptability of the scene itself is improved. In reference [7], adaptive learning rate and learning rate are introduced into the Gaussian mixture model to improve the accuracy of background model establishment and target detection rate. In reference [8], aiming at the problem of slow convergence speed of the model, an online K-means clustering method is proposed to initialize the model and speed up the implementation of the algorithm. In reference [9], aiming at the problem of background color interference in complex environment, a Mean Shift tracking algorithm based on target motion information is proposed. By introducing the significance detection MSS algorithm, the algorithm improves the discrimination between the target and the background, reduces the interference of the background information to the target location, and realizes the real-time tracking of the moving target in the video stream. In reference [10], aiming at the problem of drift caused by occlusion in the field of image tracking, a fusion method of Mean Shift algorithm and Kalman prediction filter based on the background weighting of the direct graph ratio is proposed. By increasing the difference of Bhattacharyya coefficient under normal tracking state and occlusion state, the tracking performance under occlusion is improved.

The above research focuses on the realization of background modeling algorithm from the perspective of light, color, occlusion and other single interference. In this study, the characteristics of environmental noise in complex background are integrated. Based on the mixture gauss model and Mean Shift algorithm, the target color histogram feature is used to eliminate the interference of environmental noise and achieve more accurate tracking effect by weighting the background and target. At the same time, the target template is updated from time to time in the tracking process to keep as consistent as possible with the status of the target and effectively improve the real-time operation under the condition of ensuring the accuracy.

2 Moving Target Detection

2.1 Moving Target Detection Method

Moving object detection is to find the changed part in the video sequence, so as to distinguish the background and foreground objects. Because the actual detection environment often has various dynamic changes, such as illumination, occlusion, shadow, etc., the detection algorithm is required to improve the robustness to the environment changes, and still can maintain good detection accuracy in the complex and changeable environment.

Common target detection algorithms include frame difference, background subtraction and optical flow.

(1) The frame difference method: It detects the foreground objects in the video sequence by continuously comparing the gray changes of the same position in the continuous video frame. In the case of differential continuous video frame, it needs to be carried out in the same coordinate system and background, that is to say, the change of gray value between the two frames is represented by the binary difference result. In the difference result, the background without gray level change is subtracted, and then the foreground target in the video sequence is extracted through the threshold value to highlight it [11].

(2) The background subtraction method: It assumes that the video background is static, subtracting the video frame from the fixed background, and then sets the threshold value in the difference result to obtain the moving target. The setting of threshold affects whether the correct moving target can be successfully captured from the binary image. If the threshold value is too large, the foreground target is prone to fragment in it; on the contrary, other noises may appear in the difference result, causing interference to the detection process [12].

(3) The optical flow method: It can detect the target by estimating the optical flow field of the video sequence. This algorithm mainly obtains the velocity field from the video sequence by analyzing the gray

level change of the video frame, and obtains the moving target from it according to certain restrictions. Motion field is the projection of spatial motion such as target, lens and background on the image. Optical flow field and motion field are not the same, but in many cases, the former is a rough approximation of the real motion field [13].

2.2 Moving Target Detection Based on Hybrid Gaussian Model

In this study, the foreground and background of video sequence images are classified based on Gaussian mixture model, and then background subtraction is used to detect moving objects. In this method, the Gaussian probability density function is used to build a probability model for each different category, and the characteristics of different categories are taken as the random results of the model.

Suppose that the model contains k Gaussian distributions to define the attributes of a point in the video sequence. At any sampling time t, the probability of this point being considered as background is:

$$p(X_t) = \sum_{k=1}^{k} \omega_{k,t} \eta(X_t, \mu_{k,t}, \Sigma_{k,t}) \tag{1}$$

In the formula, K is generally determined by the calculation and storage capacity of the computer, and the value is usually 3-5. $\omega_{k,t}$, $\mu_{k,t}$, $\Sigma_{k,t}$ represent the weight, mean and covariance of the k-th distribution, and $\sum_{k=1}^{K} \omega_{k,t} = 1$; $\Sigma_{k,t} = \sigma_k^2 I, \sigma_k$ is the standard deviation; η represents the k-th Gaussian distribution:

$$\eta(X_t, \mu_{k,t}, \Sigma_{k,t}) = \frac{1}{(2\pi^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}})} e^{-\frac{1}{2}(X_t - \mu_k)^\tau \Sigma^{-1} (X_t - \mu_k)} \tag{2}$$

It can be seen from equation (2) that the distribution is mainly determined by two parameters of variance and mean.

When initializing the background model, we need to set the initial value for each parameter, including the variance, mean value and corresponding proportion of each Gaussian distribution. At this time, the mean value and variance of all models are equal, and the pixel value of the first video frame is taken as their mean value. In general, the weight is determined by estimating the prior probability of the background model. Generally, the first model is assigned to a larger proportion, and the rest is correspondingly reduced, that is

$$\omega_k = \begin{cases} W & k = 1 \\ \dfrac{1-w}{k-1} & k = 1 \end{cases} \tag{3}$$

In the process of updating the model, every time a new pixel X_{t+1} is read in, it will be compared with all the current distributions in turn. As long as the difference between it and the mean value of a certain distribution meets a certain range, it can be determined that this point matches with it.

$$| X_{t+1} - \mu_{k,t} | < \delta \sigma_{k,t} \tag{4}$$

Among them, δ is generally taken as 2.5~3.5.

If there is a matching distribution in the background model, all parameters of the distribution are updated by using online K-means estimation.

$$\omega_{k,t+1} = (1-\alpha)\omega_{k,t} + \alpha(M_{k,t}) \tag{5}$$

$$\mu_{k,t+1} = (1-\rho)\mu_{k,t} + \mu \cdot X_t \tag{6}$$

$$\sigma_{k,t+1}^2 = (1-\rho)\sigma_{k,t}^2 + \rho(X_t - \sigma_{k,t})^\tau + (X_t - \mu_{k,t}) \tag{7}$$

Where, α is the learning rate, and

$$M_{(k,t)} = \begin{cases} 1 & \text{matching} \\ 0 & \text{mismatching} \end{cases} \quad \textbf{(8)}$$

$$\rho = \alpha\eta(x_{t+1} \mid \mu_{k,t}, \sigma_{k,t}) \quad \textbf{(9)}$$

For other mismatched distributions, only their weights are updated:

$$\omega_{i,t+1} = (1-\alpha)\omega_{i,t} \quad \textbf{(10)}$$

If no matching model is found after all comparisons, replace the model with the minimum proportion with the new distribution with the mean value of X_{t+1}, and set a smaller proportion and a larger covariance for it. By continuously performing similar operations during the detection process, the system can always provide the latest background for detection.

In the process of target detection, it is generally considered that the pixels in the image belong to the background most of the time, while the foreground target is only a temporary dynamic phenomenon. Based on the above logic, all models are sorted in the order of ω / σ decreasing. And the top B distribution is regarded as the background, and the rest is the foreground target, that is to say, the background should have a slightly larger proportion and a slightly smaller variance.

$$B = arg\min_b(\sum_{k=1}^{b} w_k > T) \quad \textbf{(11)}$$

Among them, the threshold value t ∈ [0, 1], which specifies the minimum proportion of models belonging to background in all distributions.

This algorithm determines whether each image point belongs to the background according to the background model, in fact, it is to compare the matching between the pixels in the image and all the distribution of the model. If we can't find out the corresponding distribution in the first B distribution, we can consider it as the former scenic spot, and all the former scenic spots constitute the moving targets to be detected. In addition, matching and updating are carried out synchronously in the process of processing to cope with the complex changes of environment and scene. The algorithm flow is shown as in Fig. 1.

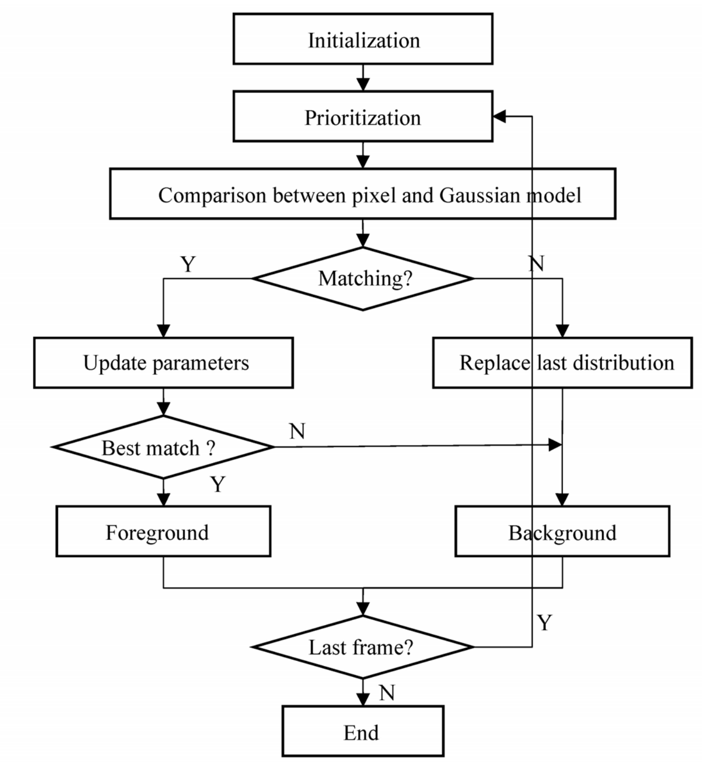


Fig. 1. Moving target detection algorithm

The experiment is programmed and implemented in Visual Studio 2019. The parameters are initialized, in which the number of distribution K=3, the learning rate α = 0.01, the mean and variance are equal, that is μ _i=σ_i=6, the matching parameter δ = 2.5, and the foreground detection threshold T = 0.75. The experimental results are shown in Fig. 2. The results show that the algorithm can adapt to other interference in the video environment of this research, and the accuracy of detection effect is high. The algorithm eliminates the noise in the background, which shows that the effect of the protective belt is removed from the experimental results, and the detection results are better.

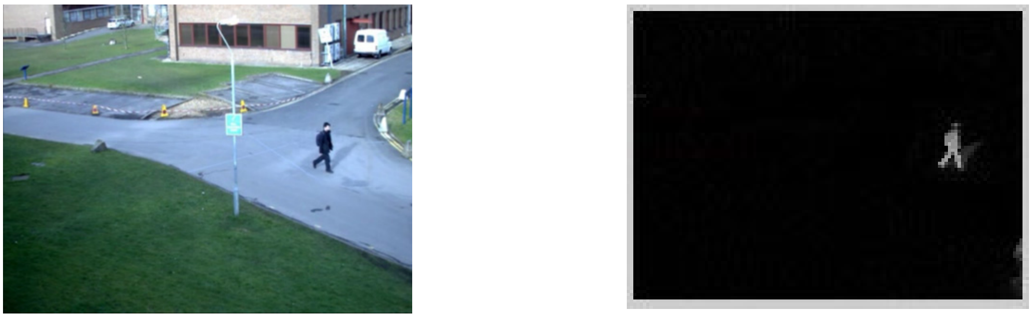


Fig. 2. Background modeling processing results

3 Moving Target Tracking

3.1 Theoretical Basis of Mean Shift

In 1975, Fukunaga [14] and others proposed Mean Shift algorithm, but its widespread concern began in 1995. Cheng [15] promoted Mean Shift algorithm and broadened its application range in the field of image vision. Mean Shift algorithm is a kind of nonparametric density estimation algorithm and a typical matching search algorithm. Because of its small computation and outstanding performance in the underlying image processing, it has been widely applied.

For d-dimensional space R^d, the vector mean value of any sample point X_i (i = 1, 2, … n) relative to x is

$$M_h(x) = \frac{1}{k}\sum_{x_i \in S_h}(x_i - x) \tag{12}$$

Where, k means that there are k points in the region S_h, and S_h is a set of high-dimensional spherical points satisfying the following conditions:

$$S_h(x) = \{y : (y-x)^T(y-x)\} \le h^2 \tag{13}$$

Obviously, for all sample points in S_h, the calculation of $M_h(x)$ is not affected by the distance from point x. But usually the sampling point near x is more conducive to the estimation of the statistical characteristics of this point. In addition, it is also considered that different sample points play different roles. Therefore, when calculating the Mean Shift vector, the non-flat kernel function is used to ignore the distance factor, and the weight coefficient $\omega(x_i)$ is used to represent the importance of sample points. In the end, Mean Shift takes the form

$$M(x) = \frac{\sum_{i=1}^{n} G_H(x_i - x)\omega(x_i)(x_i - x)}{\sum_{i=1}^{n} G_H(x_i - x)\omega(x_i)} \tag{14}$$

Among them, G (x) is a non-flat kernel function, and H is a positive-definite symmetric bandwidth matrix of $d \times d$. For simplicity of calculation, in practice, H is usually written as a unit proportional matrix H=h^{2}I. This formula only needs to determine a coefficient h. Therefore, formula (3) is rewritten as

$$M_h(x)=\frac{\sum_{i=1}^{n} G(\frac{x_i-x}{h})\omega(x_i)(x_i-x)}{\sum_{i=1}^{n} G(\frac{x_i-x}{h})\omega(x_i)} \tag{15}$$

The essence of the Mean Shift algorithm is the process of continuously approaching the dense area of the image toward the target with increasing probability density, that is, the process of the local average moving closer to the area where most points are located. At the same time, the analysis of feature space is closely related to the regional gradient. The larger the gradient, the more reliable information contained in the region. When moving to this position, you should slow down and reduce the step size. On the contrary, the smaller the gradient, the less reliable information is contained in the area. When moving to this position, you should increase the speed and enlarge the step size. This is conducive to the optimization of the analysis. Therefore, Mean Shift is also a gradient rising algorithm with a variable step size.

The Mean Shift algorithm is a continuous loop execution process. The following parameters need to be provided during initialization: start position x, kernel function G (x), and error threshold, The algorithm continuously performs the following steps:

Step 1: Compute vector $M_h(x)$;

Step 2: Translate x with $M_h(x)$;

Step 3: If $\| M_h(x)-x \| < \varepsilon$ or the number of loops is reached, the loop ends, otherwise execution continues step 1.

3.2 Mean Shift Target Tracking

In the process of video target tracking, the calculation of the mean offset vector consists of three parts: target modeling, similarity measurement and model matching. First, initialize the target model and calculate the approximation function of the target model. Secondly, by virtue of the property that the mean deviation is beneficial to cyclic operation, the average deviation vector is obtained, so that all points are moved to the maximum value of the density function, that is, the position of the target in the current frame.

It is necessary to select appropriate target model for target tracking. In the usual tracking environment, colors and outlines are often used to describe the properties of the target. Considering the computation and adaptability of the algorithm, the color histogram is used in the feature model of the target.

When initializing the tracking algorithm, the target area is often selected by the ellipse or rectangle, and the attribute space is divided into m uniform sub-intervals. If $\{x_i^*\}_{i=1,2,\dots,n}$ is the equalized pixel of the target area in the video frame, then the target model q at x_0 is

$$q_u = C\sum_{i=1}^{n} k(\| x_i^* \|^2)\,\delta[B(x_i^*)^2-u] \tag{16}$$

Where, q_u represents the probability of the u-th attribute of the target range, and $B(x_i^*)$ represents the corresponding attribute value at the position x_i^*, $\delta[B(x_i^*)-u]$ is used to determine whether the color value of point x_i^* in the range can be included in the u-th attribute interval. $K(x)$ is the kernel profile function, defined as

$$k(x)=\begin{cases} 1-x^2 & x<1 \\ 0 & other \end{cases} \tag{17}$$

Similarly, the candidate target model $p(y)$ at y is

$$p_u(y)=C_h\sum_{i=1}^{n_h} k(\| \frac{y-x_i}{h} \|^2)\,\delta[B(x_i^*)-u] \tag{18}$$

Therefore, target tracking can be understood as continuously searching for y with the highest similarity, so that $p_u(y)$ and q_u are the closest. In the process of target tracking based on Mean Shift method, it is necessary to use the gap between them to measure its approximation. There are many evaluation criteria for approximation, and the Bhattacharyya coefficient ρ is adopted in this study.

$$\rho(y) = \rho[\rho_u(y), q_u] = \sum_{u=1}^{m} \sqrt{p_u(y) q_u} \tag{19}$$

The Bhattacharyya distance between the target model and the undetermined model is expressed as

$$d(y) = \sqrt{1 - \rho(y)} \tag{20}$$

The essence of tracking is to continuously search for the smallest d(y), so that the target model best matches the candidate target model. In order to calculate the candidate target region with the maximum similarity coefficient ρ, this study directly calculates the gradient of ρ to position y_0 (current candidate target position), then

$$\nabla \rho(y) = \frac{C_h}{\sigma^2} \sum_{i=1}^{n_h} \{(y_0 - y_i) k'(\| \frac{y_0 - y_i}{\sigma} \|^2) \omega_i\} \tag{21}$$

Where, weight

$$\omega_i = \sum_{u=1}^{m} \sum_{v=1}^{n} s_u \mu_{uv} \sqrt{\frac{q_u}{p_u(y_0)}} \delta[B(x_i) - v] \tag{22}$$

Define the mapping $g(x) = -k'(x)$, then the gradient of ρ with respect to the variable y_0 is expressed as

$$\nabla \rho(y) = \frac{C_h}{\sigma^2} [\sum_{i=1}^{n_h} g(\| \frac{y_0 - y_i}{\sigma} \|^2) \omega_i][\frac{\sum_{i=1}^{n_h} y_i \omega_i g(\| \frac{y_0 - y_i}{\sigma} \|^2)}{\sum_{i=1}^{n_h} \omega_i g(\| \frac{y_0 - y_i}{\sigma} \|^2)} - y_0] \tag{23}$$

According to the properties of kernel function, the first term of formula (12) is non-zero positive number. If the gradient $\nabla \rho$ of similarity function is 0 at y_0, only the second term is equal to 0. At the beginning, usually select the optimal region in the previous frame of video to start the calculation. According to the current coordinate y_0, recalculate the undetermined target coordinate, that is

$$y_1 = \frac{\sum_{i=1}^{l} x_i \omega_i g(\| \frac{y_0 - x_i}{h} \|^2)}{\sum_{i=1}^{l} \omega_i g(\| \frac{y_0 - x_i}{h} \|^2)} \tag{24}$$

Loop iteration formula (24) until the target converges to the correct coordinates and the algorithm is completed.

3.2 Improvement of Mean Shift Algorithm and Target Tracking

Mean Shift algorithm is a typical data-driven algorithm. The algorithm obtains the target state by analyzing the image content directly, and does not rely on prior knowledge. Under normal conditions, the calculation efficiency is relatively high. Mean Shift tracking algorithm uses color, texture and other visual information to establish the target template, uses matching metrics to measure the similarity, and constantly finds the target area in the current video frame to complete the target tracking. This method has a small amount of computation and is robust to local occlusion and deformation. However, due to the lack of necessary template update in the traditional Mean Shift algorithm, the window width remains unchanged during the tracking process. When the target scale changes, it is easy to cause tracking failure.

When the tracking speed is fast, the tracking effect is not good. Histogram features are lack of description of target color features, lack of spatial information. When the background environment is complex, it is easy to lose the tracking target, or even lead to tracking failure. In order to reduce the influence of background interference and occlusion on target detection and tracking, the following aspects are improved in this study.

3.2.1 Color Space Selection

In tracking algorithms, people often distinguish different targets by color. When color is used as a feature for target tracking, RGB color space is generally used for image processing. This method is simple and direct, but changes in light conditions and occlusion will greatly affect the tracking results. In order to make the application of the algorithm more extensive, it is necessary to select the appropriate color space for tracking and improve the anti-interference ability of the algorithm to the light environment.

In this study, YCbCr color model is chosen to represent the color features, and the histogram of chroma component under the model is used as the target feature. It is also a common color coding method, which is used to describe the color in JPEG image. Where Y is luminance, CB and Cr are chrominance information. When the target is tracked according to the color attribute, using the property that brightness and chroma can be isolated in YCbCr model, we can have the anti-interference ability to the light intensity change by ignoring the brightness content.

YCbCr color space has the following advantages:

(1) Simple calculation. The components Y, Cb and Cr can be obtained by transformation of R, G and B.

(2) Brightness isolation. The chromaticity component is insensitive to light and shade changes.

(3) High real time capability. Color space coordinates and calculation process is relatively simple.

The formula for transforming RGB model into YCbCr model is as follows

$$\begin{bmatrix} Y \\ C_b \\ C_r \\ 1 \end{bmatrix} = \begin{bmatrix} 0.2990 & 0.5870 & 0.1140 & 0 \\ -0.1687 & -0.3313 & 0.5000 & 128 \\ 0.5000 & -0.4187 & -0.0813 & 128 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \\ 1 \end{bmatrix} \tag{25}$$

For the same object under different brightness conditions, Cb and Cr in the YCbCr color model were used for color matching. In the corresponding histogram, the two components maintained good invariance. Compared with RGB color model, YCbCr model has better matching degree and robustness, so color matching has better light intensity adaptability. Schematic diagram of color space contrast with different brightness is shown as in Fig. 3.

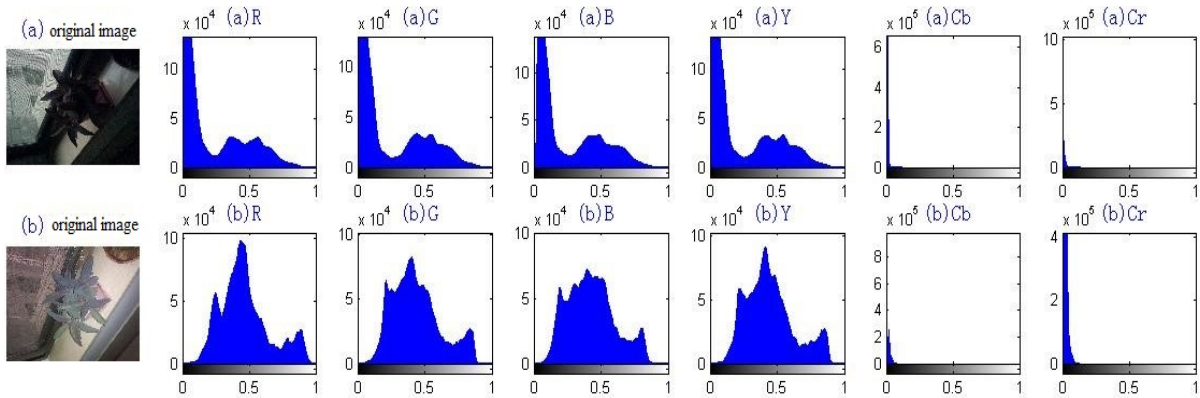


Fig. 3. Schematic diagram of color space contrast with different brightness

3.2.2 Background Weighting and Target Weighting

In the actual conditions, the background content is closely related to the characteristics of the target, which causes great interference to the accurate tracking of the target. By using the background weighting operation mechanism, the irrelevant information in the environment can be passivated, the accuracy of matching can be improved, the search range of the algorithm can be narrowed, and the result of target tracking can be optimized finally. Suppose $\{F_u\}_{u=1,2,\ldots,m}(\sum_{i=1}^{m} F=1)$ is the free attribute point in the attribute space background, and F* is the minimum value of all the positive attributes. Define the weight of two template transformations as follow:

$$\omega_i' = \sqrt{\min(\frac{F_u^*}{F_u}, 1)\omega_i} \tag{26}$$

Its purpose is to reduce the use of low weight attributes in the target template.

In addition, in the video environment, the tracking effect is often biased or even lost due to occlusion. The purpose of setting up the target weighting mechanism is to sharpen the key features of the target. In this study, the weight of the target center is set as 1. According to the increasing direction of the distance from the center, the weight decreases, and the weight at the edge is almost 0. Then, the weight of a point in the target area ω_i'' is

$$\omega_i'' = 1 - \sqrt{\frac{(x_i - x_0)^2}{a^2 + b^2} + \frac{(y_i - y_0)^2}{a^2 + b^2}} \tag{27}$$

Among them, a and b represent half of the length and width of the initial rectangle respectively, (x_0, y_0) is the centroid of the rectangle.

Combined with the original model, solve the target model again

$$q_u = C'\omega_i'\omega_i'' \sum_{i=1}^{n} k(\| x_i^* \|^2)\, \delta[B(x_i^*) - u] \tag{28}$$

Similarly, the target model to be determined is

$$p_u(y) = C_h'\omega_i'\omega_i'' \sum_{i=1}^{n_h} k(\| \frac{y - x_i}{h} \|^2)\, \delta[B(x_i) - u] \tag{29}$$

Where, constant $C_h' = \dfrac{1}{\sum_{i=1}^{n} \omega_i'' k(\||\frac{y - x_i}{h}\||^2) \sum_{u=1}^{m} \omega_i'\, \delta[B(x_i) - u]}$

Therefore, according to the current position y_0, the new position is

$$y_1 = \frac{\sum_{i=1}^{n_h} x_i \omega_i''' g(\||\frac{y_0 - x_i}{h}\||^2)}{\sum_{i=1}^{n_h} \omega_i''' g(\||\frac{y_0 - x_i}{h}\||^2)} \tag{30}$$

Where, $\omega_i''' = \sum_{i=1}^{n} \omega_i'' \sum_{u=1}^{m} \sqrt{\frac{q_u}{p_u(y_0)}} \omega_i' \delta[B(x_i) - u]$

3.2.3 Template Updating

The target template plays an important role in the tracking process. Because the background information may change at any time in the process of motion, the noise will continue to accumulate. If the original target template is still used to match the changed candidate template, it will cause a certain degree of interference to the matching accuracy. Therefore, it is necessary to keep updating the model in tracking to achieve the best matching results.

In this study, the following methods are used to update the model [16]:

Calculate the background discrete model of the current frame and model updating factor ρ :

$$\rho = \sum_{u=1}^{m} \sqrt{F_u F_u'} \tag{31}$$

If ρ > t, update the model

$$q' = (1 - \rho + T) \times q + (\rho - T) \times p \tag{32}$$

Where, q is the target model in the current image, P is the target object obtained therein, T is the threshold value for judging whether the update operation is needed, and q' is the updated target model.

The improved Mean Shift algorithm is as follows:

Step 1: Select target area, set i=1, calculate the target model q at y_0 ;

Step 2: Get next video frame;

Step 3: Select candidate area, calculate $p(y_0)$;

Step 4: Calculate weight w_i^m ;

Step 5: Calculate the new position $\mathbf{y_1}$ and its target model $p(y_1)$ in the candidate target area;

Step 6: Determine the Bhattacharyya coefficient of the current frame at y_1 and y_0. If $\rho[p(y_1), q] \geq \rho[p(y_0), q]$, then go to step 9, else go to step 7;

Step 7: Reset y_1 to the average of y_1 and y_0 , $i = i + 1$;

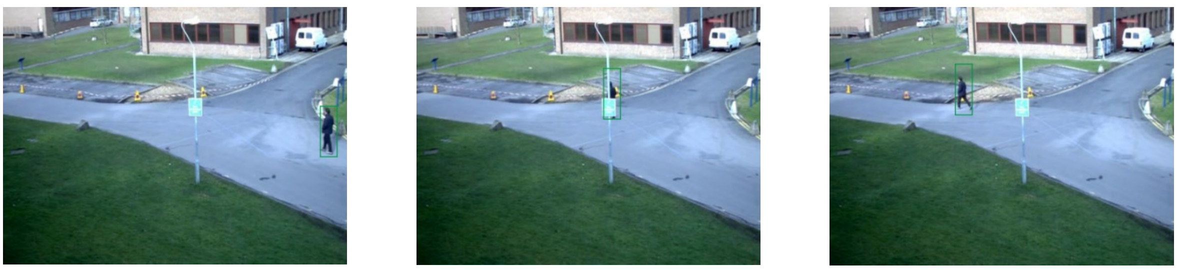
Step 8: Suppose e is the threshold and N is the maximum number of iterations, if $\| y_1 - y_0 \| > e \;\&\&\; i < N$, then y_1 is assigned to y_0 , and it turns back to the step 3, else update the template, do not iterate the current image and go back to the step 2, get the next video frame.

Step 9 : End of trace and output results.

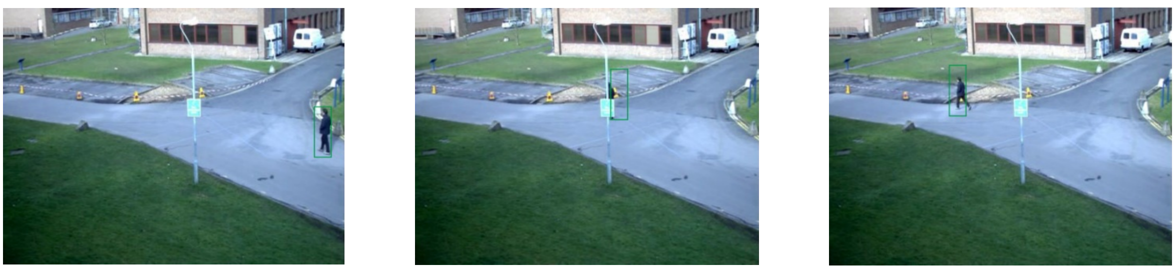
4 Experimental Results and Analysis

In this study, two groups of experiments are carried out to compare and analyze the tracking effect of the improved algorithm. Experiment 1 uses pet-09 video database for tracking simulation. The frame rate of this video is 30 frames per second, and the picture size is 720 × 480. The tracking target is partially occluded in the process of motion, and there is wind in the video environment, and the protective belt is fluttering in the wind, but the tracking environment is relatively simple as a whole. The comparison of the tracking results before and after the algorithm improvement is shown as in Fig. 4.

11 frames　　35 frames　　64 frames



(a) Improved algorithm



(b) Original algorithm

Fig. 4. Tracking results

In addition, in order to compare the differences in the calculation speed of the algorithm before and after the improvement, the experiment takes the number of iterations of the algorithm to search for the optimal candidate target as the statistical object, and takes the number as a measure of the speed of the operation and makes a comparison. The experimental results are shown as in Fig. 5. The average number of iterations after the improvement is 3.4, while the number before the improvement is 4.12.

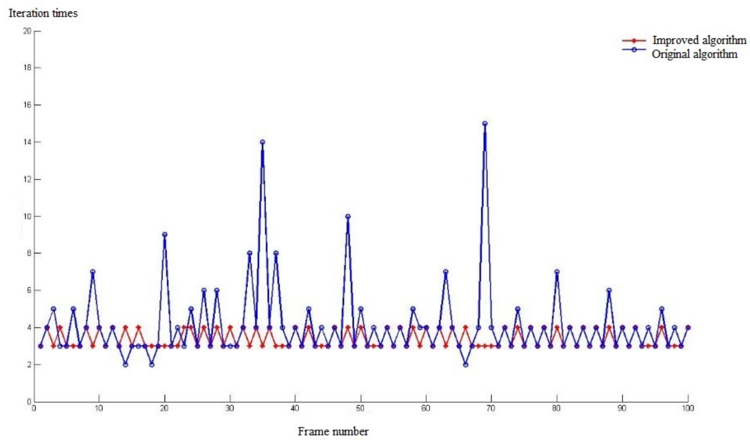


Fig. 5. Comparison of algorithm iterations

In Experiment 2, self-made video sequence was used for tracking simulation, and the location of shooting was a common scene in the campus, and pedestrians wearing red clothes were tracked in the image. The frame rate of the video is 29 frames / second, and the screen size is 1280 × 720. The result is shown in Fig. 6.

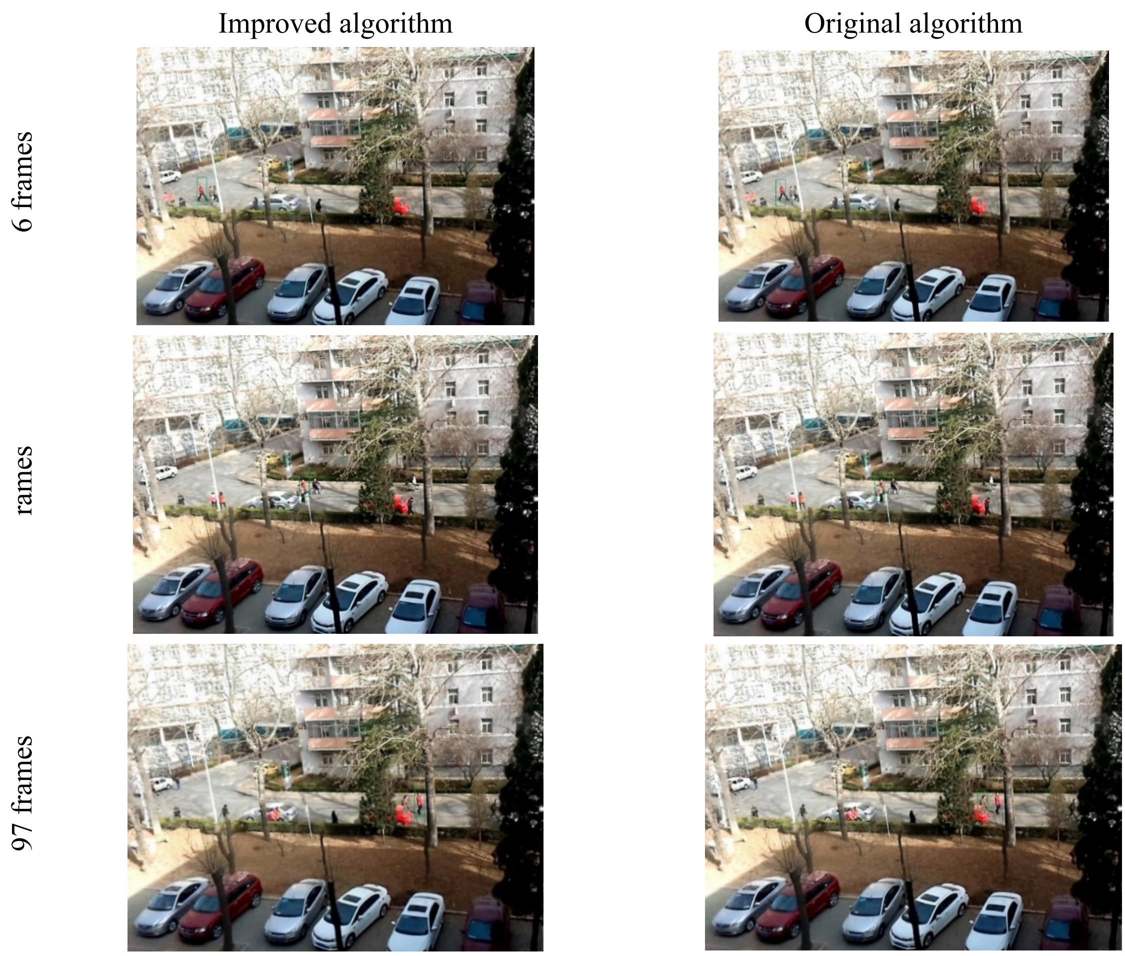


Fig. 6. Self-made video tracking results

Because there are many moving objects in the video, the tracking environment is very complex, and the shooting distance is far, the color of pedestrian's dress is close to the color of parking, which brings a lot of interference to the target tracking. In frame 97, because the original algorithm did not date the target state in time, the tracking error occurred, and the car was mistakenly used as the tracking object, However, the improved algorithm using template update strategy is not affected by similar areas in the environment, and can still accurately capture the target. Similarly, we continue to compare the number of iterations in the experiment, and the results are shown in Fig. 7. For the self-made video, the average number of iterations is 6.35 and 5.67 before and after the algorithm improvement. The speed of the improved algorithm is still improved.

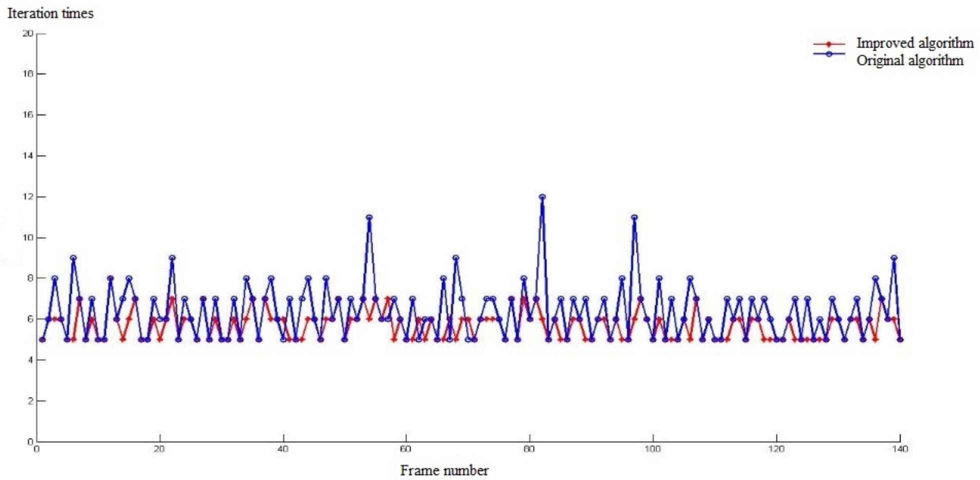


Fig. 7. Comparison of algorithm iterations

It is easy to see from the above experiments that in the case of partial occlusion, the advantages of the improved algorithm are more obvious. The reduction of the number of iterations is mainly because the improved algorithm makes the features of the target more obvious in the tracking process, reduces the search space, and has little impact on the operation speed by the environmental change.

5 Conclusion

This paper focuses on the problem of target detection and tracking in complex environment. Based on the theory of Mean Shift algorithm, aiming at the shortcomings of traditional Mean Shift algorithm, such as lack of template update, lack of spatial information and so on, an improved target detection and tracking method based on mixture Gaussian model and Mean Shift algorithm is proposed. In this method, color feature is used as the basis of target matching, and color histogram is calculated in YCbCr space to make the algorithm adapt to the changes of different light brightness. At the same time, considering the continuous change of the moving target's action state in the tracking process, we adopt the mechanism of target weighting and template updating to highlight the target's characteristics as much as possible and improve the tracking effect. The experimental results show that the improved method can effectively reduce the number of calculation iterations and the impact of complex external environment and objects similar to the target to be tested on the tracking results. This method can detect the moving target accurately and has good tracking effect.

References

[1] T. Zhou, M. Zhu, Moving target detection of the video images, Chinese Journal of Liquid Crystals and Displays 1(2017) 40-47.

[2] K.K. Halder, M. Tahtali, S.G. Anavatti, Target tracking in dynamic background using generalized regression neural network, in: Proc. International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation, 2016.

[3] C. Liu, L. Zhang, H. Huang, Visualization of cross-view multi-object tracking for surveillance videos in crossroad, Chinese Journal Of Computers 63(1)(2018) 367-370.

[4] H. Tian, Y. Ju, P. Wang, Improved anti occlusion Mean Shift tracking algorithm, Computer Engineering and Applications 52(6)(2016) 197-203.

[5] P. Hu, Y. Fan, L. Gao, Multi-object tracking based on basic color feature and adaptive scale, Application Research of Computers 12(2016) 3897-3900.

[6] X. Lang, Moving object detection and tracking algorithm based on mixture model, Modern Electronics Technique 2(2020) 70-77.

[7] M. Lv, J. Sun, Moving image target detection based on modified gaussian mixture model, Semiconductor Optoelectronics 12(2019) 874-878.

[8] Y. Xu, W. Tan, L. Chen, Moving object detection algorithm based on improved mixture Gaussian model, Control Engineering of China 25(4)(2018) 630-635.

[9] H. Wang, L. Yu, T. Guo, et al. Research on Mean Shift tracking algorithm based on target motion information, Experimental Technology and Management 3(2020) 153-157.

[10] S. Wu, Y. Mao, Q. Liu, Z. Li, Tracking method for solving occlusion problem, Chinese Journal of Liquid Crystals and Displays 2(2019) 188-193.

[11] Y. Hu, Y. Zhou, Q. Zhang, High point monitoring pedestrian detection based on background modeling and inter frame difference method, Research and Exploration in Laboratory 9(2018) 12-16.

[12] X. Liu, G. Cui, Z. Li, W. Xiong, Small target motion detection based on layering of vision system. Optics and Precision Engineering 27(10)(2019) 2251-2262.

[13] G. Ma, W. Yang, J. Wen, et al. Method of crowd anomaly detection based on scene similarity and optical flow, Modern Electronics Technique 1(2020) 90-96.

[14] K. Fukunaga, L.D. Hosteller, The estimation of the gradient of a density function, with application in pattern recognition, IEEE Transactions on Information Theory 21(1975) 32-40.

[15] Y.Z. Cheng, Mean shift, mode seeking, and clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 17(8)(1995) 790-799.

[16] S. Chen, S. Wang, Moving object tracking based on five frame difference and improved meanshift algorithm, Computer Science 6(2016) 203-206.