Monitoring of Ruminating Behavior of Dairy Cows Based on Scale-Adaptive KCF

Jiaxin Nie[*1], Jiandong Fang[*2], and Yvdong Zhao[*3]

[*1]Inner Mongolia University of Technology,Hohhot,Inner Mongolia,010080,China
E-mail: 845273938@qq.com
[*2]Inner Mongolia Key Laboratory of Perceptive Technology and Intelligent System,Hohhot,Inner Mongolia,010080,China
E-mail: fangjd@imut.edu.cn
[*3]Inner Mongolia Key Laboratory of Perceptive Technology and Intelligent System,Hohhot,Inner Mongolia,010080,China

Abstract. Detecting and tracking the ruminating behavior of dairy cows can obtain information on their physiological activities in time, and provide effective data for dairy cow health monitoring. This study is based on the scale-adaptive KCF(kernel correlation filter). Based on the acquisition of the cow's mouth area, the cow rumination activity is tracked in real time, and the rumination curve is drawn to obtain the cow's rumination amount within a certain time range. The experimental results show that the improved scale-adaptive KCF algorithm can accurately analyze the ruminating behavior of dairy cows.

Keywords: Cow Rumination; Target Tracking; Adaptive Scale

1. INTRODUCTION

As a key technology in the current video analysis field, the target tracking method has been gradually applied to various fields in life and production. Using moving target detection and tracking methods to track and monitor the ruminating behavior of dairy cows in real time, collecting and analyzing relevant ruminating information data, can provide effective dairy cow health information for farms and facilitate early treatment of dairy cows in the early stages of illness, which is important to the dairy farming industry the guiding significance.

Target tracking is to determine the position and size change of the object in the subsequent video given the initial position and size of the target object. It is to perform moving target detection, feature extraction, classification recognition, tracking filtering and behavior recognition for video images, so as to obtain accurate information parameters of the target (such as speed, acceleration, position, motion trajectory, etc.), and perform corresponding processing and analysis on it , So as to realize the understanding of the target behavior[1].

Traditional target tracking algorithms include optical flow method, mean shift method, particle filter and methods based on online machine learning. Among them, methods based on online machine learning are divided into generative-based methods and discriminant-based methods[2].The generative-based target tracking method does not need to consider background information. It builds a model to represent the target through learning, and then uses the model to directly match the target category, which has achieved the purpose of tracking. The discriminant-based target tracking method transforms target tracking into a binary classification problem that seeks to track the decision boundary between the target and the background, and maximizes the classification of the target area from the non-target area. Different from the generative method, the discriminative method uses the target foreground and background information at the same time, and through the online update classifier mechanism, the classifier is used to distinguish the foreground and background of the target.

When tracking a cow's ruminant target, since the opening and closing scale of the cow's mouth will change when chewing, and the movement of the cow will also affect the tracking result, choose a method that can adapt to the target scale change and the tracking speed is fast. It is very necessary to obtain the ruminant information of dairy cows. As a kind of discriminant algorithm, the KCF(kernel correlation filter) algorithm transforms the target tracking problem into a binary classification problem of seeking to track the boundary between the target and the background by introducing the concept of correlation filtering in the communication field. It has obvious advantages in the tracking speed and accuracy method[3].

In order to monitor the rumination behavior of dairy cows and obtain ruminating information, this research is based on video analysis and target tracking technology. On the basis of acquiring the mouth area of dairy cows, a kernel-related filtering algorithm based on scale adaptation is selected for target tracking. The DSST(discriminative scale space tracking), which can accurately estimate the target scale in tracking, is combined with the KCF algorithm. The KCF algorithm obtains the position information of the target, and the DSST extracts the characteristics of the target at different scales to determine the current target scale.By continuously updating the position and scale information of the target in each frame of the image, obtaining the characteristics of cow rumination, and finally evaluating the performance of the algorithm.

2. SCALE ADAPTIVE ALGORITHM COMBINING KCF AND DSST ALGORITHMS

2.1. Algorithm Model

In the process of cow rumination, because the mouth is constantly opened and closed during chewing, the scale changes, so by judging the change in the length of the tracking frame during target tracking, the amount of cow rumination can be obtained. The KCF algorithm is suitable for the situation where the scale of the target is fixed during the tracking process. When used to track the cow's mouth, the tracking frame with a fixed size will not be able to determine whether the cow is ruminating, and because the tracking frame cannot be scaled according to the size of the target Changes will cause the tracking frame to drift to a certain extent during the tracking process.

Refer to the paper [4][5], combining the DSST algorithm with the KCF algorithm. When extracting target features, the KCF tracker is fused with the DSST algorithm scale filter, and the target scale is adaptively improved.

In the first frame of the video image, select the target area to be detected, and enlarge the selected target area by 2.5 times as the candidate tracking area. Use the sample cyclic shift method to generate the training sample set for the candidate area, and obtain the position filter according to the sample set training Device. Extract the HOG features of the candidate samples, establish an observation model, add a cosine window to reduce the boundary effect, and perform fast Fourier transform at the same time to ensure the speed of the algorithm. Multiply the result obtained in the above steps with the correlation filter and then perform the inverse Fourier transform to obtain the distribution map of the target output, and the maximum response position is the target position.

When estimating the scale of the target, take the middle position of the target as the center to intercept training images of different scales, and use a scale filter with a size of $M \times N \times S$ to obtain S image blocks of different scales.For each image block, its feature descriptor is obtained, and the scale filter template is obtained by training. When detecting a new frame of image, take the middle position of the target as the center, and perform Fourier transform according to the training process, and the scale corresponding to the maximum value is the final estimated target scale.

Repeat the above steps to get the target position and scale of the next frame until the last frame of the video.The flowchart is shown in Figure 1:

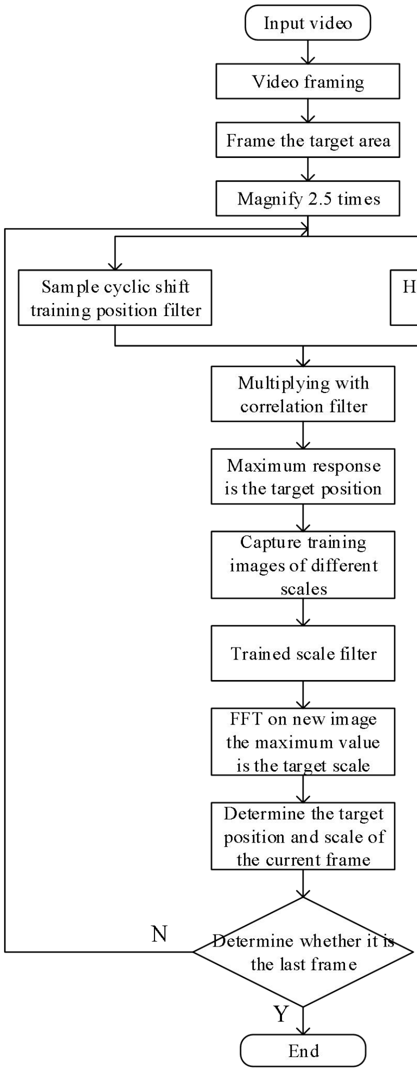

Fig. 1 Model flow chartt

2.2. KCF Tracking Algorithm

The KCF algorithm was first proposed by Henriques et al.[6]. This algorithm is classified as a discriminative target tracking algorithm according to the classification of the observation model. By designing a filter template, the next frame of image is convolved with the filter template, and the area with the largest response is the location of the prediction target[7]. The KCF algorithm first selects the target area to be detected in the first frame of image, and initializes the filter. In the subsequent image and video, the sample cyclic shift is used to replace the sampling window for modeling, and the target position is cyclically shifted to obtain the candidate sample, that is, the circulant matrix. Using the circulant matrix can speed up the running speed of the algorithm. Then extract the target feature from the next frame of target image, build the observation model, and get the classifier. The target model filter and candidate samples obtained from the first frame image are Fourier transformed to obtain the target prediction position, and the observation model is continuously updated to track the video. The algorithm flow is shown in Fig.2:

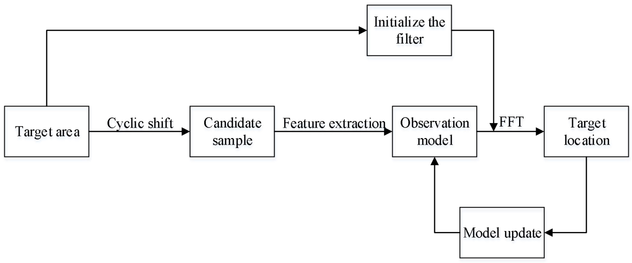

Fig. 2 KCF algorithm flow chart

The sample cyclic shift is to multiply the permutation matrix P and the vector x to perform cyclic shift. $x = [x_1, x_2 \cdots, x_{n-1}, x_n]$:

$$P = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ & & \vdots & & \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{1}$$

The vector after shift is $Px = [x_n, x_1, x_2 \cdots, x_n]^T$.

When extracting target features, KCF uses Histogram of Oriented Gradient (HOG features) to extract target features. This method composes features by calculating and counting the gradient histograms of local areas of the image[8]. First, the image to be detected is divided into connected regions, namely cells, and then the gradient or edge direction histogram of each pixel in the cell is collected, and finally the histograms are composed to obtain a feature descriptor. At the same time, by calculating the density of each histogram in a larger range, and normalizing the cells in the range according to the obtained density, the influence of the image on the tracking effect due to changes in illumination can be reduced. The steps of HOG feature extraction are:

(1) Grayscale the image to be detected.
(2) Gamma correction method is used to normalize the image, adjust the image contrast, thereby reducing the impact of image shadow and light changes, and reducing noise.
(3) Calculate the gradient of each pixel of the detection image, obtain the contour information of the target, and further reduce the influence of light. The calculation formula of the gradient value of each pixel is as formula (2):

$$\begin{cases} G_x(x,y) = H(x+1,y) - H(x-1,y) \\ G_y(x,y) = H(x,y+1) - H(x,y-1) \end{cases} \tag{2}$$

(4) Divide the detection image into cells, and count the gradient histogram of each cell.
(5) Construct several cells into a large area, that is, a block, and the feature composed of all cells in the block is the HOG feature of the block.
(6) The HOG feature of all blocks is connected to the HOG feature of the image.

The construction of the observation model uses the ridge regression problem, and the optimal solution with the least loss can be obtained through ridge regression[9]. Suppose the training sample set is (x_i, y_i), the sample x_i is the column vector, y_i is the regression label, and the linear regression function is:

$$f(x_i) = \omega^T x_i \tag{3}$$

Among them,ω is the weight coefficient of the column vector, and the vector is written in matrix form, and the minimum average error between the training sample and the regression label is:

$$min\|X\omega - y\|^2 + \lambda\|\omega\|^2 \tag{4}$$

Among them, λ is a regularization parameter that controls overfitting.

$X = [x_1, x_2, \cdots, x_n]^T$ each row represents a sample, $y = [y_1, y_2, \cdots, y_n]^T$ is a column vector, and each element represents a sample label. The least squares method in equation (4) is solved as:

$$\omega = \left(X^T X + \lambda I\right)^{-1} X^T \tag{5}$$

KCF differs from other related filtering algorithms in that it introduces a kernel function to solve the non-linear problem in target tracking[10]. The kernel function maps the input space to the high-dimensional feature space through the nonlinear function φ(x), so that the obtained solution is the optimal solution. Kernel function $k(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle$, which means the inner product of samples x_i and x_j, and the expression is:

$$K_{ij} = k(x_i, x_j) = \varphi(x_i)^T \varphi(x_j) \tag{6}$$

The regression function form of high-dimensional space can be expressed as:

$$f(x_j) = \sum_{i=1}^{m} \alpha_i \langle \varphi(x_i)^T \varphi(x_j) \rangle = K\alpha \tag{7}$$

Taking the partial derivative of α, we can get:

$$\alpha = (K + \lambda I)^{-1} y \tag{8}$$

Where K is the kernel correlation coefficient obtained by cyclic shift of the training sample, which can make the linear problem be mapped to the nonlinear kernel space. For the Fourier transform on both sides of equation (8), we get:

$$\alpha = \left(\frac{1}{k^{xx} + \lambda}\right)^* \odot \mathrm{y} \tag{9}$$

k^{xx} represents the first row of the circulant matrix K, which can simplify the calculation process, k^{xx} is the Gaussian kernel, and the expression is:

$$k^{xx} = \exp\left(-\frac{1}{\sigma^2}\left(\|x\|^2 + \|x'\|^2 - 2F^{-1}\left(x^* \odot x'\right)\right)\right) \tag{10}$$

σ is the time domain bandwidth, the larger the σ, the thicker the Gaussian graph[10].

In order to adapt to changes in the appearance of the tracking target, the model needs to be continuously updated. The bilinear difference method is added to the update of the target model, the filter coefficient α and the target observation model x are:

$$\alpha_t = (1-\mu)\alpha_{t\text{-}1} + \mu\alpha_t \tag{11}$$

$$x_t = (1-\mu)x_{t\text{-}1} + \mu x_t \tag{12}$$

2.3. SCALE ADAPTIVE KCF ALGORITHM

The key to scale adaptation of the DSST algorithm lies in the construction of a three-dimensional filter[5], that is, a one-dimensional scale filter + a two-dimensional position filter. Two mutually independent filters meet the requirements of combining scale estimation with the KCF algorithm.

The DSST algorithm first uses a position filter to determine the position information, and then uses a scale filter to obtain the scale information of the target on the basis of obtaining the target position. The position filter has a total of 28-dimensional features, including one-dimensional gray-scale features and 27-dimensional fhog features[5]. The fhog feature removes the block on the basis of the HOG feature, and directly performs the four-dimension feature in each cell. Row and column accumulation and extraction[11]. In estimating the position of the target, first calibrate the position frame of the target, extract the characteristic information of the target position frame, select the three-dimensional filter constructed by the three-dimensional Gaussian function to output the response y, and the maximum response value is the target position. The calculation formula is as follows (13):

$$y = F^{-1}\left\{\frac{\sum_{l=1}^{d} \overline{A^l} Z^l}{B+\lambda}\right\} \tag{13}$$

Z is the input feature map, λ is the regular term to prevent the denominator from being 0, A_t and B_t are the numerator A and denominator B obtained by splitting the filter H, and the terms obtained after iterative update:

$$A_t = \eta F_t \cdot G_t^* + (1-\eta)A_{t-1} \tag{14}$$

$$B_t = \eta F_t \cdot G_t^* + (1-\eta)B_{t-1} \tag{15}$$

Where η represents the learning rate.

The scale filter uses a one-dimensional correlation filter to estimate the scale of the target. Obtain the target image from the target position obtained in the previous step, take the midpoint of the target image as the center, and intercept pictures of different scales, so as to achieve scale adaptation in a small range. When the target size is P×R and the scale is S, the target center size obtained by the scale filter is:

$$a^n P \times a^n R \tag{16}$$

a is the scale factor, n is the scale level, and the value range is $n \in \left[\left(-\frac{S-1}{2}\right),......,\left(\frac{S-1}{2}\right)\right]$. For two consecutive frames of images, the change in position is often greater than the change in scale[4]. Therefore, in the target tracking process, the location of the target is determined first, and then the scale information is obtained.

3. EXPERIMENTAL RESULTS AND ANALYSIS

3.1. Experimental Environment and Parameter Settings

Experimental code hardware environment: Intel i7, 3.6 GHz CPU, RAM 16 GB. Software environment: 64-bit Windows 10 operating system, PyCharm Community Edition 2020.3.4, python 3.6.

Parameter settings: The minimum average error regular term λ in the KCF tracker is 0.0001, the target rectangle parameter padding is 2.5, and the Gaussian target bandwidth σ in the kernel function is 0.125. The size of the HOG feature cell array is 4 × 4, the adaptive linear difference factor γ is 0.012, and the Gaussian convolution kernel bandwidth σ is 0.6. The sample number S of the scale filter is 33, and the scale factor a is 1.05.

Three cows ruminating videos are selected for testing. Video 2 and Video 3 are closer to the camera and the target is larger; Video 1 is farther from the camera and the target is smaller. Obtain the height of the tracking frame in the running video, compare the KCF algorithm and the improved scale-adaptive KCF algorithm. During the tracking process, whether the target frame is scaled according to the size of the cow's mouth, and qualitatively analyze the performance of the scale-adaptive KCF algorithm.

Table. 1 Experimental test video.

Video number	Video frames	Video resolution /pixel	Video duration /s
1	381	768×576	12
2	270	720×480	9
3	769	500×310	32

3.2. Target Tracking Experiment Results

Fig. 3-Fig. 5 are screenshots of the video frame and the corresponding tracking frame coordinates of the comparison tracking results of KCF and scale-adaptive KCF:

a. KCF b. Scale adaptive KCF

Fig. 3 Video 1 tracking results and tracking frame coordinates.

a. KCF b. Scale adaptive KCF

Fig. 4 Video 2 tracking results and tracking frame coordinates.

a. KCF b. Scale adaptive KCF

Fig. 5 Video 3 tracking results and tracking frame coordinates.

Table 2 shows the partial tracking results obtained by the KCF and scale-adaptive KCF algorithm during the cow's mouth closing and mouth opening process of the above three video frames. The coordinates of the upper left and lower right corners of the tracking frame, the size of the tracking frame, and the tracking The amount of change in the frame length and width scale:

Table. 2 Tracking frame coordinates and scale

Video No.	Frames	KCF			Scale adaptive KCF		
		Frame coordinates	Frame size (w, h)	Scale change	Frame coordinates	Frame size (w, h)	Scale change
1	14th	(75,120,103,148)	(28,28)	0	(79,118,103,142)	(24,24)	(3,4)
	26th	(74,123,102,151)	(28,28)		(74,120,101,148)	(27,28)	
2	44th	(154,176,231,238)	(77,62)	0	(151,175,223,228)	(72,53)	(4,2)
	55th	(151,171,228,233)	(77,62)		(152,179,228,234)	(76,55)	
3	78th	(202,113,239,153)	(37,40)	0	(207,119,237,150)	(30,31)	(6,7)
	90th	(200,114,237,154)	(37,40)		(202,115,238,153)	(36,38)	

From Figure 2 to Figure 4, it can be seen that the KCF and the scale-adaptive KCF tracking frame have no obvious deviation of the target in the three tracking videos, and the tracking result is basically the same as the actual position of the target. It can be seen from Table 2 that in the three videos of the KCF algorithm, the tracking frame is the same size and the scale is not changed during the cow's mouth closing-mouth opening process; the scale-adaptive KCF algorithm tracking frame can be based on the contour of the cow's mouth when the cow is ruminating. When the cow's mouth is opened, the height of the tracking frame increases, the mouth is closed, and the tracking frame falls back.

3.3. Cows Ruminant Acquisition

In order to verify the effectiveness of the scale-adaptive KCF algorithm for obtaining the amount of rumination of dairy cows, according to the tracking frame coordinates obtained in the previous step, draw the curve of the height of the tracking frame as the rumination curve during the cow rumination tracking process. The amount of rumination in the time frame.

The time it takes for a cow to chew once is about 0.8-1.3 seconds[12], which is about 20-30 frames of images in this experiment. When a cow completes a chewing, the movement state of the mouth is closed mouth — open mouth — closed mouth. When the mouth is opened, the height of the tracking frame reaches the maximum value, and when the mouth is closed, the height of the tracking frame drops. On the curve of the height change of the tracking frame, the maximum value of the curve in the 20-30 frames of images is one chewing of the cow. Therefore, the number of chewing times of cows can be determined as the amount of rumination according to the number of maximum values of the track frame height curve in a certain video frame.

In order to more accurately determine the performance of the scale adaptive algorithm, the false detection rate and frame processing speed are used to evaluate the algorithm:

False detection rate m_r: the absolute value of the difference between the cow's chewing times j_c and the actual chewing times j_r calculated according to the ruminating curve, divided by the actual chewing times, as in formula (17):

$$m_r = |j_c - j_r| / j_r \times 100\% \tag{17}$$

Frame processing speed f_s: the total number of video frames f_n divided by the time t required for the algorithm to process the video, as in formula (18):

$$f_s = f_n / t \tag{18}$$

Fig.6-8 shows the height curve of three video tracking frames:

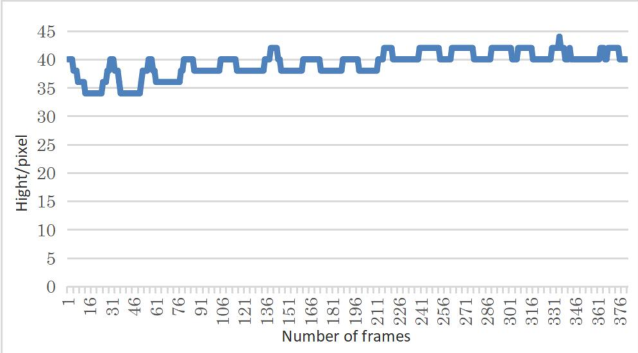

Fig. 6 Video 1 tracking frame height curve

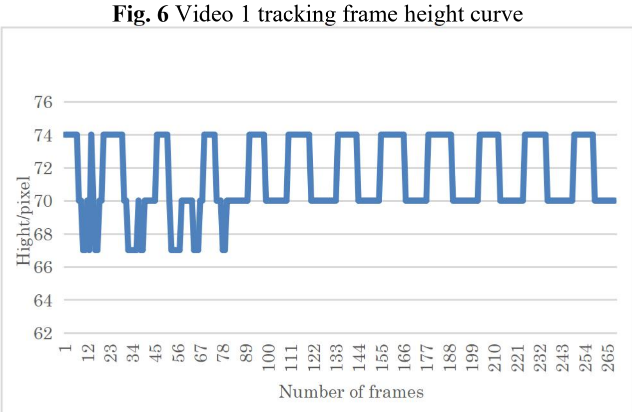

Fig. 7 Video 2 tracking frame height curve

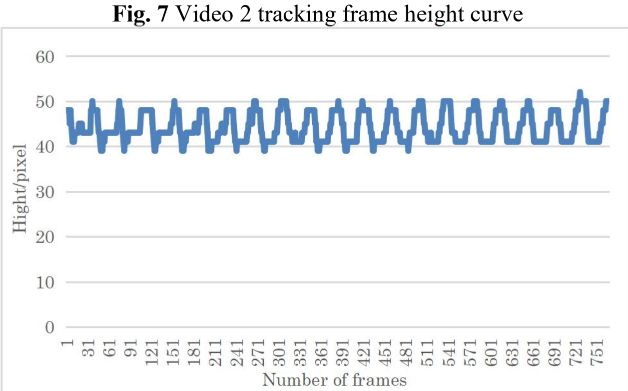

Fig. 8 Video 3 tracking frame height curve

The maximum value of the tracking frame height in Figure 6 is 42 pixels on average, and lasts about 10 frames of images. There are 4 dense undulating line segments near frames 338 and 375, and 4 maximum values appear. According to the analysis of the previous paragraph, the average cow in the video Chew once in 20-30 frames of image, so it can be judged that there is an error in the scale conversion of the tracking frame, and it is considered that there are 2 chewings in this video. The overall curve is relatively complete, and the tracking frame rises and falls with the change of the cow's mouth. From the maximum number of the curve, it can be concluded that the video 1 cow chews 15 times. The curve in Figure7 has multiple maximums and minimums in the first 78 frames. It can be concluded that the scale-adaptive KCF has poor performance in the initial stage of the video, and the error is large. In the 192 frames of the subsequent video, the algorithm works well. The curve is stable, and the tracking frame rises in 7 frames of image on average, and the tracking frame of 12 frames falls back. From the maximum value of the curve, it can be concluded that video 2 cows chewed 15 times. The curve in Figure 8 has a stepwise rise process before the maximum value of the tracking frame height appears. The maximum value is an average of 50 pixels. At the same time, it can be seen from the graph that in the 22nd frame of the video, the maximum value of the tracking frame height corresponds to the maximum value of 50 pixels. Compared with other video frame images, the maximum value is smaller, which reflects that the algorithm has more obvious errors in the early stage of video operation. The whole graph can clearly show that the maximum number is 22, that is, the cow chews 22 times.

Table 3 shows the false detection rate and frame processing speed value of the scale-adaptive KCF algorithm in three test videos:

Table. 3 Cow chewing times, false detection rate and frame processing speed

Video No.	Actual number of chews	Scale adaptive KCF		Frame processing speed/(frame·s-1)
		Number of chews	False detection rate/%	
1	14	15	7.14	50
2	12	14	16.67	54
3	21	22	4.76	56

After calculation, the scale-adaptive KCF algorithm has an average of 50 frames/s, 54 frames/s and 56 frames/s for the three video frames. The tracking speed is negatively correlated with the video resolution. The lower the video resolution, the frame The higher the processing speed. This algorithm has a low false detection rate in video 1 and video 3. Video 2 has a relatively high false detection rate due to the short video duration, and the comprehensive false detection rate of the three videos is 9.52%.

4. CONCLUSION

This research analyzes the principle of the KCF algorithm, applies the KCF algorithm and the scale-adaptive KCF algorithm fused with the DSST algorithm to target tracking on dairy cow ruminating videos, and draws the tracking frame height curve obtained by the scale-adaptive KCF algorithm as the ruminating curve, which verifies The applicability of this method to the monitoring of dairy cow ruminating behavior.

(1) The tracking frame position and scale change obtained by the two algorithms for cow target tracking show that the tracking result is basically the same as the actual target position, and both can achieve automatic target tracking. The KCF algorithm tracking frame cannot change the scale according to the size of the target, and it is more suitable for the situation where the scale of the target is fixed during the tracking process. The scale-adaptive KCF algorithm tracking frame can be adaptively changed according to the target scale to determine whether the cow is ruminating.

(2) It can be seen from the ruminating curve and evaluation index that the scale-adaptive KCF algorithm has an average false detection rate of 9.52% for cow chewing times, an average success rate of 90.48%, and an average tracking speed of 53.3 frames/s, which is suitable for cow rumination. Real-time monitoring, but the algorithm has poor effect in the initial stage of video operation and high error. It should be gradually improved in subsequent research.

REFERENCES:

[1] Yang Kaiyun. Research on Visual Target Tracking Based on Kernel Correlation Filtering Algorithm [D]. Jiangsu University of Science and Technology, 2020.

[2] Lu Huchuan, Li Peixia, Wang Dong. Overview of target tracking algorithms [J]. Pattern Recognition and Artificial Intelligence, 2018, 31(01): 61-76.

[3] Tang Yiming, Liu Yufei, Huang Hong. Overview of visual single-target tracking algorithms[J]. Measurement and Control Technology, 2020, 39(08): 21-34.

[4] Hu Yuncai, Lu Hong, Yang Chen, Hua Xiang, Peng Jun. Scale adaptive tracking algorithm combining DSST and KCF[J]. Computer Engineering and Design, 2019, 40(12): 3563-3568.

[5] Huang Yueping, Li Xiaofeng, Yang Xiaogang, Qi Naixin, Lu Ruitao, Zhang Shengxiu. New development of visual target tracking algorithm based on correlation filtering [J/OL]. System Engineering and Electronic Technology: 1-19 [2021-04-28].

[6] Henriques João F et al. High-Speed Tracking with Kernelized Correlation Filters.[J]. IEEE transactions on pattern analysis and machine intelligence, 2015, 37(3): 583-96.

[7] Li Jun, Dai Tian, Wei Rui. Improved target tracking algorithm with nuclear correlation filter [J]. Sensors and Microsystems, 2021, 40(03): 146-148.

[8] Wang Fan, Shen Shibin, Zhang Yue, Xie Fei, Lu Fei, Liu Yijian. Multi-face feature extraction method based on HOG-SIFT fusion optimization[J]. Journal of Nanjing Normal University (Engineering Technology Edition), 2020, 20(03): 43-49.

[9] Wang Tingting. Research on target tracking algorithm based on nuclear correlation [D]. Xi'an University of Technology, 2020.

[10] M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg. "Accurate scale estimation for robust visual tracking". In Proceedings of the British Machine Vision Conference (BMVC), 2014.

[11] Liu Xiaoyue, Wang Yunming, Ma Weining. Scale-adaptive correlation filter tracking algorithm fusing FHOG and LBP features[J]. Progress in Laser and Optoelectronics, 2020, 57(04): 330-337.

[12] Song Huaibo, Niu Mantang, Ji Cunhui, Li Zhenyu, Zhu Qingmei. Multi-target dairy cow ruminating behavior monitoring based on video analysis[J]. Transactions of the Chinese Society of Agricultural Engineering, 2018, 34(18): 211-218.