Visual tracking for underwater sea cucumber via correlation filters

Honglei Wei, Xiangzhi Kong, Xianyi Zhai, Qiang Tong, Guibing Pang*

(*School of Mechanical Engineering and Automation, Dalian Polytechnic University, Dalian 116038, Liaoning, China*)

Abstract: One of the essential techniques for using underwater robots to fish sea cucumbers is that the robots must track sea cucumbers using computer vision technology. Tracking underwater targets is a challenging task due to suspension, water absorption, and light scattering. This study proposed a simple but effective algorithm for sea cucumber tracking based on Kernelized Correlation Filters (KCF) framework. This method tracked the head and tail of the sea cucumber respectively and calculated the scale change according to the distance between the head and tail. The KCF method was improved on three strategies. First of all, the target was searched at the predicted position to improve accuracy. Secondly, an adaptive learning rate updating method based on the detection score of each frame was proposed. Finally, the adaptive size of the histogram of the oriented gradient (HOG) feature was used to balance the accuracy and efficiency. Experimental results showed that the algorithm had good tracking performance.

Keywords: visual tracking, correlation filters, kernelized correlation filters, sea cucumber, scale estimation, underwater

DOI: 10.25165/j.ijabe.20231603.4503

Citation: Wei H L, Kong X Z, Zhai X Y, Tong Q, Pang G B. Visual tracking for underwater sea cucumber via correlation filters. Int J Agric & Biol Eng, 2023; 16(3): 247–253.

1 Introduction

Sea cucumber is a kind of aquatic product with high nutritional value and economic value. Sea cucumbers live on the bottom of the sea and are mainly fished by diving. The traditional method of fishing is low in yield, high in cost, and seriously harmful to divers' health. Intelligent underwater robots have been widely used in search and rescue, salvage, and other marine activities. The most viable alternative is to capture sea cucumbers using an autonomous underwater vehicle (AUV), such as the underwater robot shown in Figure 1. One of the critical technologies for the AUV is to accurately track sea cucumbers, which is the target tracking subject of computer vision.

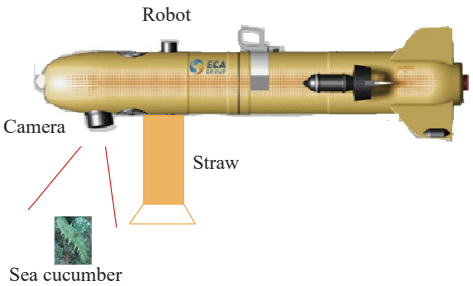


Figure 1 Schematic of fishing sea cucumber using a robot

The goal of a visual tracking algorithm is to train a classifier to distinguish between objects and environments. There are many research achievements in visual tracking, which can be divided into discriminative methods or generative methods. Discriminative methods model target tracking as a classification problem[1-11]. The generative techniques take target tracking as a template matching problem[12-15].

The process starts with visual tracking to determine the location of the sea cucumber and suck it into the storage tank through a straw.

In the comprehensive evaluation of online target tracking based on benchmark[16], those trackers[1, 5-10] based on correlation filtering show good performance in estimating target translation at the fastest speed. Compared with the model-based method, these tracking algorithms based on correlation filters calculate spatial correlation in the form of wise product in the Fourier domain to obtain higher tracking speed. Since they take into account environmental information, they can provide better tracking results. Bolme et al[6]. initially proposed the minimum output sum of squared error (MOSSE) filter for tracking, which could process hundreds of frames per second (FPS) due to the efficiency of the correlation filter. Henriques et al.[7] proposed a circulant structure tracker (CSK), which used a circulant matrix to obtain larger training samples. Kernel Correlation Filter (KCF) presented by Henriques et al.[1] was adopted to improve the CSK method by utilizing the features of HOG. Liu et al.[8] conducted tracking in multiple parts based on the KCF method. In the study of Xia et al.[17-19], an improved algorithm based on Unscented Rauch-Tung-Striebel Smoother had been added to the Kernel Correlation Filter algorithm, and the sparse representation method has been introduced into the training process to heighten the stability of the proposed object tracking algorithm. Guo et al.[20] based on the original histogram of oriented gradient features, integrated the hue, saturation, value, and grayscale information to construct a new descriptor to represent the target appearance. Yan et al.[21] used the gray area growth method to detect the candidate small target region and segment the final small target by the threshold value. Du et al.[22] proposed a kernel-correlation filtered target tracking algorithm that introduces a target block model and the algorithm is more robust in dealing with lighting changes, scale changes, occlusions, and

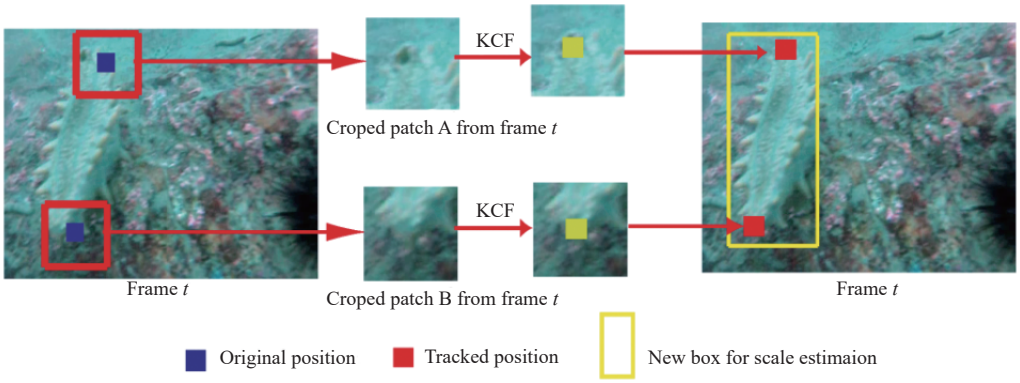
Received date: 2018-07-08 **Accepted date:** 2021-07-30

Biographies: Honglei Wei, PhD, Associate Professor, research interest: machine vision technology, Email: weihl2005@163.com; **Xiangzhi Kong**, Postgraduate, research interest: image processing, Email: 963827458@qq.com; **Xianyi Zhai**, Postgraduate, research interest: object detection, Email: zhaixianyi9421@foxmail.com; **Qiang Tong**, PhD, Lecturer, research interest: image processing, Email: tongqiang@dlpu.deu.cn.

***Corresponding author: Guibing Pang**, PhD, Professor, research interest: mechanical manufacturing technology and measuring instrument technology. School of Mechanical Engineering and Automation, Dalian Polytechnic University, 1st Qinggongyuan, Ganjingzi, Dalian 116034, Liaoning, China. Tel: +86-411-86324497, Email: pangguibingsx@163.com.

background noises.

Due to the low contrast and poor quality of underwater images caused by underwater light, suspended matter, light absorption, and scattering, it is more difficult for robots to capture sea cucumbers than target tracking on land. In addition, the relative position between the robot and the sea cucumber constantly changes during the capture process, resulting in constant changes in scale. Some approaches methods[10,18] use the scale pyramid method to calculate the scale change, estimating the scale change reasonably. Still, the calculation is considerable, and it is not suitable for real-time application. This study proposed a simple and effective tracking method based on the KCF framework[1], which has high computational efficiency and robustness to scale changes. The flow chart of the proposed method is shown in Figure 2. The main contributions of this study can be summarized as three strategies. First of all, the method tracks the head and tail of the sea cucumber, respectively, calculates the change in the distance between the two ends of the sea cucumber, and estimates the scale change. Secondly, the KCF method[1] is improved by tracking the image blocks cropped on the predicted position. According to the detection score, a way of adjusting the learning rate is proposed to update the filter template of each frame.



Note: Kernelized Correlation Filters.

Figure 2 Flowchart of the proposed scale estimation method.

Furthermore, adaptive HOG size is adopted to achieve the balance of computational accuracy and efficiency. An experimental study was carried out on a typical sea cucumber video data set. Experimental results show that the algorithm has good performance.

The method of this study tracks two parts located at two ends of sea cucumber (A and B) separately by KCF tracker[1] and estimates the scale according to the distance between two positions.

2 Materials and methods

2.1 Kernelized correlation filter

KCF tracker[1] trains a filter w by minimizing the squared error over samples x_i and their regression targets y_i.

$$\min_w \sum_i \langle(\varphi(x_i), w) - y_i\rangle + \lambda\|w\|^2 \quad (1)$$

where, λ is a regularization parameter; the training examples (x_i, y_i) are cyclic shifts of the basic sample (x, y); $\varphi(x_i)$ represents mapping x_i to kernel space.

The solution w of Equation (1) can be expressed as:

$$w = \sum_i \alpha_i \varphi(x_i) \quad (2)$$

According to Equation (2), the optimization variable changes from w to α_i. The objective function can be minimized if α_i is defined as:

$$\hat{\alpha}^* = \frac{\hat{y}}{\hat{\kappa}^{xx'} + \lambda} \quad (3)$$

where, $\hat{\kappa}^{xx'}$ is the Gaussian kernel of dot-product $\varphi(x)\varphi(x_i)$, and the hat $\hat{}$ denotes the Discrete Fourier Transformation (DFT) of a vector.

During the tracking process, a patch was cropped in the new frame to calculate the kernel $\hat{\kappa}^{xz}$, and then the correlation response mapping $f(z)$ was generated by convolving the filter α with the kernel, as shown in the following equation:

$$f(z) = F^{-1}(\hat{\kappa}^{xz} \circ \hat{\alpha}) \quad (4)$$

where, $\circ$ is the element-wise product; F^{-1} denotes the inverse of Fourier transforms.

In the response map $f(z)$, the location with the maximum value was the target location.

2.2 Method for sea cucumber tracking

This section researches the strategies for sea cucumber tracking, including improved translation estimation of KCF[1] and scale estimation by tracking head and tail.

2.2.1 Translation estimation

In order to reduce computation and improve the robustness, three methods were proposed, including adaptive HOG algorithm, predictive location detection algorithm, and adaptive learning rate algorithm.

1) Adaptive HOG

In the traditional KCF method[1], the size of HOG was fixed to 4, which led to the slow tracking speed of large objects. The size of the HOG proportional was set to the size of the target, as shown in Equation (5):

$$r = \frac{\text{size}_{\text{target}}}{m} \quad (5)$$

where, r is the adaptive HOG size; $\text{size}_{\text{target}}$ is the smaller one of the length and the width; m is the fixed value (set as 25 in this study). If r is less than 2, set r to 2.

2) Location prediction

In the KCF method[1], the cropped patch used for tracking needs to be processed by the cosine window to process the wrapped-around edges, causing the center to be enhanced and the rest to be weakened. The further away the target is from the original location, the more difficult it is to find the target, which usually happens when the target moves fast. To solve this problem, the position of the current frame was predicted according to the tracking position of the last three frames, and then the filter was performed on the

clipped image block on the predicted position to improve the tracking stability. Equation (6) can calculate the predicted position.

$$p^t = p^{t-1} + \gamma\left(p^{t-1} - p^{t-2}\right) \tag{6}$$

where, γ is the step coefficient, which can be calculated by Equation (7).

$$\gamma = \begin{cases} \dfrac{\left\|p^t - p^{t-1}\right\|}{\left\|p^{t-1} - p^{t-2}\right\|} & \left\|p^{t-1} - p^{t-2}\right\| >= 5 \\ 1 & \left\|p^{t-1} - p^{t-2}\right\| < 5 \end{cases} \tag{7}$$

In Equations (6) and (7), p^{t+1} is the position of the prediction point, and p^t, p^{t-1}, p^{t-2} are the tracking positions of t, $t-1$, $t-2$ frames, respectively.

When the target changes direction or speed rapidly, the prediction Equation (6) may give the wrong position. To solve this problem, the distance between the predicted location and the tracked location was checked.

If the distance was less than the threshold (20 pixels in the experiment), the tracked result would have been accepted. Otherwise, the tracked location patch is cropped and traced again.

3) Model Update

In the KCF method[1], the model is updated at a fixed learning rate and is not adjusted according to the specific situation. The learning rate was adjusted according to the maximum value of the response mapping. That is, the frames with higher detection scores will be learned more. Therefore, the update scheme is defined as:

$$\begin{cases} H^t = \rho H^t + (1-\rho) H^{t-1} \\ x^t = \rho x^t + (1-\rho) x^{t-1} \end{cases} \tag{8}$$

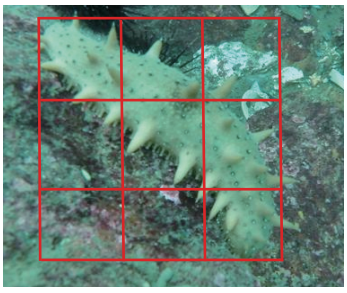
where, the learning rate ρ is calculated as

$$\rho = \eta \max\left(f(z)\right) \tag{9}$$

The η is a constant parameter; $f(z)$ is the response mapping.

2.2.2 Scale estimation

In the process of using underwater robots to catch sea cucumbers, the scale changes constantly due to the changing distance between the robot and the sea cucumber. Scale estimation not only makes the tracking process more robust but also can be used to estimate the relative distance between the underwater robot and the target. In this section, a scale estimation method was developed, and the flowchart is shown in Figure 3.



Note: The red squares indicate the divided blocks.

Figure 3 Searching for the location of two ends from the image patches in the first frame.

1) Selection of tracking points

The proposed method was realized by tracking the head and tail of the sea cucumber, respectively. The first step was to select the block in the head and tail of the target in the first frame that was most suitable for tracking.

The target image block x is divided into nine equal parts x_{ij} in the first frame, as shown in Figure 3, where $i, j \in (1,2,3)$. The part x_{22} was used to calculate filter $\hat{\alpha}$, The image block x_{ij} and center block x_{22} were used to calculate the kernel $\hat{\kappa}^{x_{22}x_{ij}}$, and Equation 4 was used to calculate the response map $f(z_{ij})$. Let s_{ij} denotes the maximum value of response map $f(z_{ij})$, and l_{ij} denotes the corresponding location. The pairs (l_{11}, l_{33}), (l_{12}, l_{32}), (l_{13}, l_{31}), (l_{21}, l_{23}) were checked and the pair with the maximum value of the sum of two s_{ij} were selected as the initial tracking points.

The target object was split into nine smaller image patches, and each window shares one-third of the size of the target. The ends of the sea cucumber were image patches with an appearance most similar to the center patch.

2) Scale calculation

The new position $p_A^t\left(x_A^t, y_A^t\right)$ of the head and the new position $p_B^t\left(x_B^t, y_B^t\right)$ of the tail could be tracked in frame t based on the KCF method[1] starting from the two starting positions selected in the first frame.

Unlike the KCF method, the image patch was not cropped directly on p_A^{t-1} and p_B^{t-1} but the image patch was cut in the predicted positions $\bar{p}_A^t$ and $\bar{p}_B^t$ calculated by Equation (6), and the Equation (10) can calculate the scale.

$$s^t = \frac{\left\|p_A^t - p_B^t\right\|}{\left\|p_A^{t-1} - p_B^{t-1}\right\|} s^{t-1} \tag{10}$$

3) Scale Verification

It was necessary to verify the results to avoid tracking failure caused by scale estimation error. As shown in Figure 4, the algorithm in this study tracks the two endpoints of sea cucumber, as shown in Figure 4a. In order to compare with ground truth, the central point of the target must be calculated, which requires two steps: scaling and rotation, as shown in Figure 4b, and the target position p_G in frame t can be calculated by the following equation:

$$p_G^t = \lambda R\left(p_B^t - p_A^t\right) + p_A^t \tag{11}$$

where, λ is the scale conversion parameter, which can be calculated by the following equation:

$$\lambda = \frac{\left\|p_G^1 - p_A^1\right\|}{\left\|p_B^1 - p_A^1\right\|} \tag{12}$$

where, the p_G^1 is the block center provided by the ground truth in the first frame; R is the rotation matrix, and it takes the form as follows:

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{13}$$

To compute the rotation matrix $\boldsymbol{R}$, Equation (13) was rewritten in a detailed form:

$$\begin{cases} x_1 = x_0\cos\theta - y_0\sin\theta \\ y_1 = y_0\cos\theta + x_0\sin\theta \end{cases} \tag{14}$$

where, $[x_0, y_0]^T = \lambda\left(p_B^1 - p_A^1\right)$, and $[x_1, y_1]^T = p_G^1 - p_A^1$. The $\cos\theta$ and $\sin\theta$ can be solved by the following equation:

$$\begin{cases} \cos\theta = \dfrac{x_0x_1 + y_0y_1}{x_0x_0 + y_0y_0} \\ \sin\theta = \dfrac{x_0y_1 - x_1y_0}{x_0x_0 + y_0y_0} \end{cases} \tag{15}$$

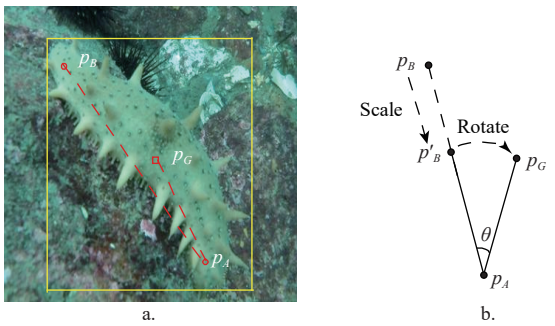
According to Equation (13) and Equation (15), the rotation matrix $\boldsymbol{R}$ can be calculated. Eventually, the p_G^t in frame t was calculated by Equation (11).

If p_G^t satisfies condition Equation (16), the scale s^t is calculated by Equation (10). Otherwise, the calculation of t is not reliable, so it

stays the same $s^t = s^{t-1}$.

$$\left\| p_G^t - p^t \right\|_2 < \varepsilon \tag{16}$$

where, $\|\cdot\|_2$ is to compute the 2-norm; p_G^t is the target center calculated by Equation (11) according to p_A^t and p_B^t, and p^t is the target center obtained by tracking the sea cucumber according to Equation (6).



a. b.

Figure 4 Illustration of the method for computing the center of the target

2.2.3 Framework of method

A brief outline of our method is given in Algorithm 1.

Algorithm 1. Proposed method: iteration at frame t. Input the image I_t, previous target position p^{t-1} and scale s^{t-1}, the model α^{t-1} and x^{t-1}.

First make the Translation estimation: Extract samples from I_t at p^{t-1} and s^{t-1}, compute the correlation score using Equation (4), and set p^t to the target position that maximizes translation estimation.

Secondly, make a scale estimation: If t=1, select tracking points p_A^1 and p_B^1 by the method introduced in Section 2.2.2, otherwise predict location $\bar{p}_A^t$ and $\bar{p}_B^t$ using Equation (10).

Then extract the sample I_A^t and I_B^t from I_t at $\bar{p}_A^t$, $\bar{p}_B^t$, s^{t-1}, and compute the correlation score y_A^t and y_B^t using (4). Set p_A^t and p_B^t to the target position that maximizes scale estimation, calculate the s^t using Equation (11) and verify the s^t using Equation (17). And then made the model update: Extract samples I_t, I_A^t and I_B^t at p^t, p_A^t, p_B^t and s^t, compute the filter using Equation (3), and update the model using Equation (8).

Finally, output the estimated target position p^t and scale s^t, the model α^t and x^t.

3 Experiments and results

3.1 Experimental setup and methods

The algorithms used in the experiment were all implemented in MATLAB on a PC with an Intel i7 2.67 GHz CPU. Center location error (CLE), overlap success plot (OP), and distance precision plot (DP) were used as evaluation criteria. CLE is the average distance from the ground truth to the location being tracked. The success plot showed the percentage of frames with overlapping rates $S > t_0$ among all thresholds $t_0 \in [0, 1]$. The overlap ratio can be calculated by $S = \frac{Area(B_T \cap B_G)}{Area(B_T \cup B_G)}$, where, B_T is the boundary box given by the tracking algorithm, and B_G is the actual ground boundary box. The distance precision plot indicates the percentage of frames that satisfy the condition $S<n$ and the distance of the tracking position from the ground truth within the whole threshold range $n \in [0, 100]$.

When the robot grabs the underwater sea cucumber, the distance and relative angle of the robot approaching the target sea cucumber will change significantly, so the experiment mainly tests the robustness of visual tracking with large-scale change and rotation change. Five sea cucumber videos with large-scale changes and five sea cucumber videos with significant rotation changes were collected for the robustness test of the visual tracking algorithm. The specific information on videos is listed in Table 1.

Table 1 Videos information for the experiment

No.	Video	Frame number	Image size	Attribute
1	S_1	1100	640×360	Scale change
2	S_2	1800	640×360	Scale change
3	S_3	1600	640×360	Scale change
4	S_4	900	640×360	Scale change
5	S_5	900	640×360	Scale change
6	S_6	1100	640×360	Scale change
7	R_1	1600	640×360	Rotation change
8	R_2	1400	640×360	Rotation change
9	R_3	1000	640×360	Rotation change
10	R_4	1100	640×360	Rotation change
11	R_5	1000	640×360	Rotation change
12	R_6	1000	640×360	Rotation change

Note: In Table 1, S_n represents significant scale changes of the tracking target in the video n; R_n represents significant rotation changes of the tracking target in the video n.

3.2 Features and parameters

The standard deviation σ of the expected relevant output is set to 1/16 of the target size, the learning rate η is set to 0.025, the regularization parameter λ is set to 0.01, and the overlap rate ρ is set to 1.5 for translation estimation and 1.0 for scale estimation.

For the translation and scale filter in the proposed method, the adaptive cell size described in Section 2.2.1 was used to extract HOG[6] features. Each feature channel in the extracted samples of both translation filters and scale filters is multiplied by a cosine window

3.3 Comparison with baseline tracker

The experiments are implemented on the tracking objects with significant rotation change and scale changes. In the experimental results, the baseline corresponds to KCF trackers[1]. The Update is the algorithm that adds the model update algorithm given in Section 2.2.1 to the Baseline, and Predict is the algorithm that adds the location prediction method given in Section 2.2.1 to the Baseline. Scale is the algorithm that adds the scale estimation method given in Section 2.2.2 to the Baseline, and Joint is the algorithm that adds the methods of Updating, predicting, and scaling to the Baseline.

1) Scale variation

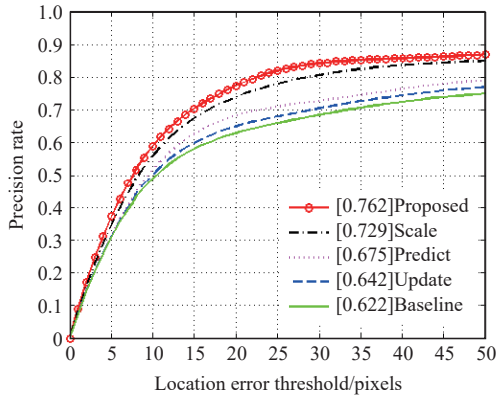
The experimental studies on six videos (S_1-S_6) are conducted and compared the results of proposed trackers with those of baseline trackers. Table 2 lists that the baseline trackers obtained a mean DP of 62.2%, and the update, predict, scale achieves a mean DP of 64.5%, 67.5%, and 72.9%, respectively. The best results were achieved by the Joint method with a significant gain of 14% over

Table 2 Comparison of experimental results between the proposed algorithm and the baseline on videos with significant scale variation

Algorithm	DP	OP	CLE/pixels	FPS/frames·s^{-1}
Baseline	0.622	0.508	18.2	**36.9**
Update	0.645	0.538	16.9	36.6
Predict	0.675	0.557	15.2	36.1
Scale	0.729	0.805	13.2	24.7
Joint	**0.762**	**0.854**	**11.1**	24.2

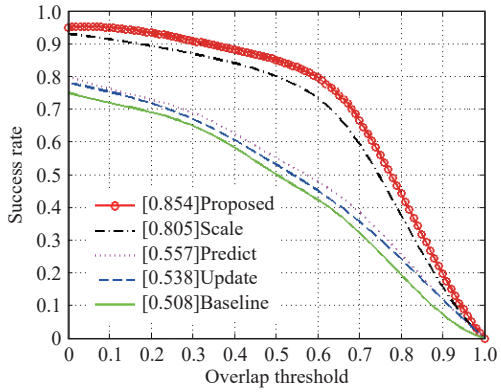
Note: DP: Distance Precision Plot; OP: Overlap Success Plot; CLE: Center Location Error; FPS: Frames per second. Same below.

the baseline tracker. Similarly, the proposed methods also provide improved performance in mean OP compared to the baseline tracker. Table 2 also shows that the baseline achieves a mean OP of 50.8%. The Update, Predict, Scale method improved the median OP by 3.8%, 4.9%, 29.7%, and 34.6% compared with the baseline tracker. Figures 5 and 6 are the experimental results of the distance accuracy and overlap success rate, which shows that the proposed trackers are effective.



Note: The legend of distance precision contains threshold scores at 20 pixels.

Figure 5 Distance precision plots over all the 7 sequences showing the performance of the proposed methods compared to the baseline



Note: The AUC score is reported in the legend of the overlap precision plot for each tracker.

Figure 6 Success plot over all the 7 sequences showing the performance of the proposed methods compared to the baseline

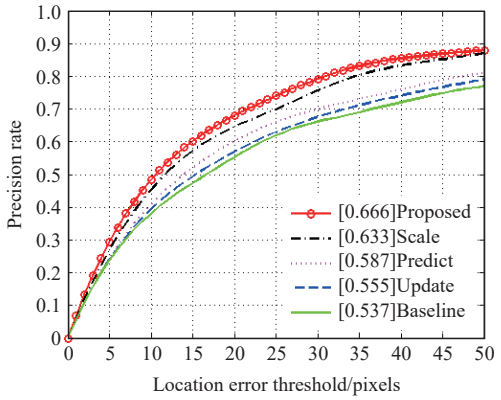
2) Rotation change

The experimental research on videos R6-R10 is carried out and compared its performance with the baseline tracker with a significant target rotation. Table 3 lists the comparison between the method and the baseline. Figures 7 and 8 are the experimental results of the distance accuracy and overlap success rate. The best performance was again achieved using Joint. Compared with the baseline, the mean DP of Joint improved by 12.9% and the mean OP by 17.4%.

Table 3 Comparison of experimental results between the proposed algorithm and the baseline on videos with significant rotation variation

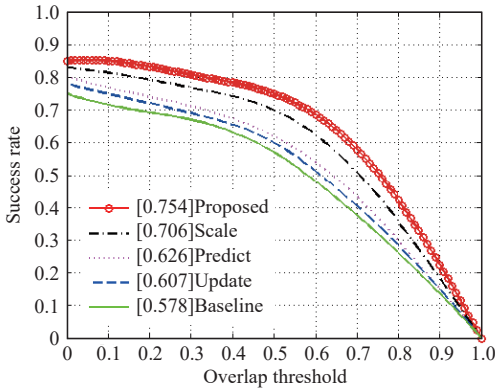
Algorithm	DP	OP	CLE/pixels	FPS/frames·s^{-1}
Baseline	0.537	0.578	23.2	**36.3**
Update	0.555	0.607	20.8	36.4
Predict	0.587	0.626	19.1	30.1
Scale	0.633	0.706	16.7	24.2
Proposed	**0.666**	**0.754**	**14.5**	24.1

In summary, the three proposed methods effectively improve the tracking accuracy, and the Joint of the three methods showed the best effect. As can be seen from the results of frames per second in Tables 2 and 3, the disadvantage of the proposed methods is that they require more computation than the baseline.



Note: The legend of distance precision contains threshold scores at 20 pixels.

Figure 7 Distance precision plots over all the 7 sequences showing the performance of our methods compared to the baseline



Note: The AUC score is reported in the legend of the overlap precision plot for each tracker.

Figure 8 Success plot over all the 7 sequences showing the performance of the proposed methods compared to the baseline

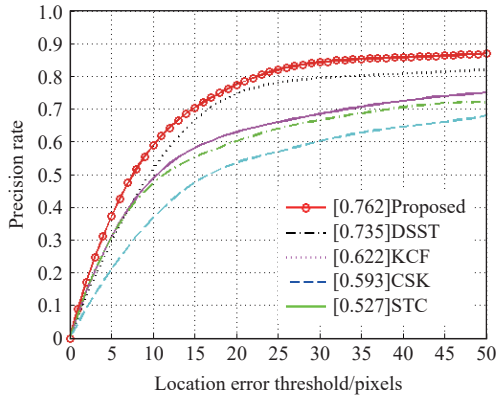
3.4 Robustness comparison with State-of-the-Art trackers

The robustness of the proposed algorithm is compared with four state-of-the-art trackers, including KCF[1], CSK[7], Discriminative Scale Space Tracker (DSST)[18], and Spatio-Temporal Context (STC)[19]. All trackers were tested under the same experimental conditions.

1) Scale variation

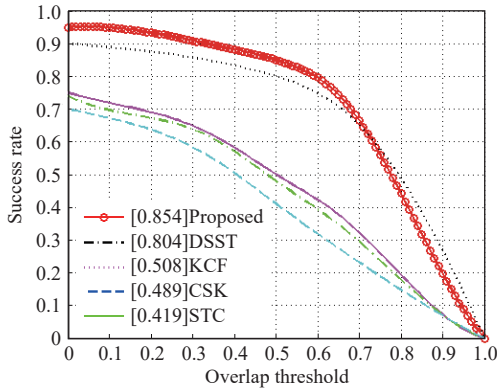
In this part, the experimental research results of the proposed algorithm on videos 1-6 were presented with large target scale variation and compared with the state-of-the-art trackers. Figures 9 and 10 illustrate the variation curves of distance accuracy and overlap success rate under different thresholds. Figure 11 shows a visual comparison of several algorithms on thirteen video sequences.

Table 4 lists that the proposed method achieves a median DP of 76.2%, which is 2.7% higher than the second-highest tracker (DSST), and a median OP of 85.4%, which is 5.0% higher second-highest tracker (DSST). The method has a running speed of 24.5 frames/s, which is relatively slow compared with KCF, CSK, and STC because scale estimation requires a great deal of computation, but the proposed method is much better than the DSST method (running speed was 3.88 frames/s), which also has scale estimation.



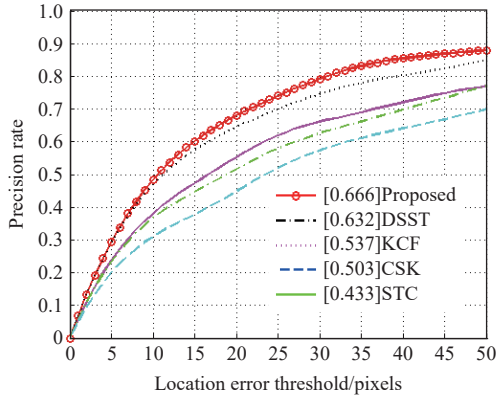
Note: Proposed means the method proposed in this study; DSST: Discriminative Scale Space Tracker; KCF: Kernelized Correlation Filters; CSK: Circulant Structure Tracker; STC: Spatio-Temporal Context. Same below. The legend of distance precision contains threshold scores at 20 pixels.

Figure 9 Distance precision plots show the proposed method's performance compared to several state-of-the-art methods over all seven sequences



Note: The area-under-the-curve (AUC) score for each tracker is reported in the legend.

Figure 10 Success plots showing the performance of our method compared to several state-of-the-art approaches overall 7 sequences



Note: The legend of distance precision contains threshold scores at 20 pixels.

Figure 11 Distance precision plots show our method's performance compared to several state-of-the-art approaches over all 7 sequences

2) Rotation changing

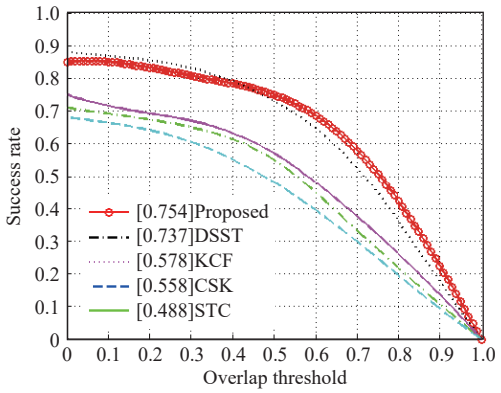
In this part, the experimental research results of the proposed algorithm on videos 7-12 with large target rotation variation are presented and compared with the state-of-the-art trackers. Figures 11 and 12 illustrate the variation curves of distance accuracy and overlap success rate under different thresholds.

Table 5 lists that the proposed method achieves a median DP of 66.6%, which is 3.4% higher than the second-highest tracker (DSST), and a median OP of 75.4%, which is 1.7% higher second-highest tracker (DSST). For the same reason as scale estimation, the proposed method run at 25.9 frames per second, which is slower than KCF, CSK, and STC without scale estimation. However, the proposed method is much better than the DSST method with scale estimation (running speed is 3.01 frames/s).

Table 4 Comparison of experimental results between the proposed algorithm and the state-of-the-art trackers on videos with significant scale variation

Algorithm	DP	OP	CLE/pixels	FPS/fames·s^{-1}
Proposed	**0.762**	**0.854**	**14.5**	24.5
DSST	0.735	0.804	15.3	3.88
KCF	0.622	0.508	20.8	36.9
CSK	0.593	0.489	21.1	54.7
STC	0.527	0.419	23.4	37.8

Note: Proposed means the method proposed in this study; DSST: Discriminative Scale Space Tracker; KCF: Kernelized Correlation Filters; CSK: Circulant Structure Tracker; STC: Spatio-Temporal Context. Same below.



Note: The area-under-the-curve (AUC) score for each tracker is reported in the legend.

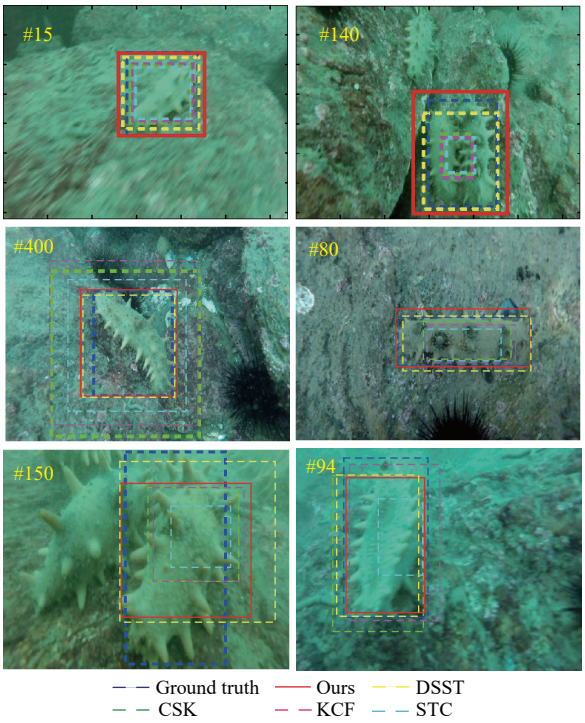
Figure 12 Success plots showing the performance of our method compared to several state-of-the-art approaches over all 7 sequences

Table 5 Comparison of experimental results between the proposed algorithm and the state-of-the-art trackers on videos with significant rotation variation

Algorithm	DP	OP	CLE/pixels	FPS/fames·s^{-1}
Proposed	**0.666**	**0.754**	**17.2**	25.9
DSST	0.632	0.737	18.1	3.01
KCF	0.537	0.578	23.2	39.2
CSK	0.503	0.558	24.5	**61.1**
STC	0.433	0.488	28.7	39.5

3) Evaluation of experimental results

The above experimental results show that the algorithm can accurately and effectively estimate the scale and location of challenging sequences. Figure 13 shows a visual comparison of several algorithms on 13 video sequences, where for clarity, only representative frames of seven sequences are given. The reasons can be summarized as follows. First of all, the algorithm used the tracking results of the critical components of the target to estimate the scale change of the target, which is more robust and effective. Second, on the new frame, the new tracking is started from the predicted position rather than the traced position on the last frame, which is very effective for fast-moving targets. Finally, the adaptive



Note: Ours: The proposed method in this study; DSST: Discriminative Scale Space Tracker; CSK: Circulant Structure Tracker; KCF: Kernelized Correlation Filters; STC: Spatio-Temporal Context (STC).

Figure 13 Qualitative evaluation of the Ground Truth, Proposed algorithm, DSST, KCF, CSK, and STC methods

HOG size is adopted to reduce the computation, which is more efficient than the fixed size methods.

In conclusion, the proposed method improves performance over standard trackers in sea cucumber harvesting, which shows that accurate scale estimation is crucial for the robustness of the tracker. For the current best scale-adaptive tracker DSST, the experimental results clearly show that the method provides a significant speed gain while maintaining competitive performance.

4 Discussion

In this study, a target tracking method based on an improved kernelized correlation filters algorithm was proposed to capture sea cucumbers under underwater conditions. The proposed scale estimation method tracked two parts of the target simultaneously and calculated the scale change effectively according to these two parts. This study also improved the method in two aspects. One was to track the image patch of the predicted position to improve accuracy. The other was to update the filter and model according to the maximum value of the response graph, and the adaptive HOG size was adopted to reduce the computation of scale estimation. The proposed scale estimation method is designed for underwater sea cucumber tracking, but it is universal and can be used in any underwater tracking framework, such as fish tracking, diver tracking, underwater vehicle tracking, and so on.

Acknowledgements

This work was financially supported by the Basic Research Project of Higher Education Institutions of Liaoning Province (Grant No. 20210126; No. 20210135).

[References]

[1] Henriques J F, Caseiro R, Martins P, Batista J. High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014; 37(3): 583–596.

[2] Babenko B, Yang M H, Belongie S. Robust object tracking with online multiple instance learning. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2011; 33(8): 1619–1632.

[3] Helmut G. Real-time tracking via on-line boosting. In: Proc. of British Machine Vision Conference, 2006; pp.47–56.

[4] Hare S, Saffari A, Torr P H S. Struck: Structured output tracking with kernels. In: IEEE International Conference on Computer Vision. Barcelona, Spain: IEEE, 2012; pp.263–270. doi: 10.1109/ICCV.20211.6126251.

[5] Yao R, Shi Q, Shen C, Zhang Y, van den Hengel A. Robust tracking with weighted online structured learning. Computer Vision – ECCV 2012. Springer, 2012; pp.158–172. doi: 10.1007/978-3-642-33712-3_12.

[6] Bolme D S, Beveridge J R, Draper B A, Lui Y M. Visual object tracking using adaptive correlation filters. In: 2010 IEEE Computer Vision and Pattern Recognition. San Francisco, CA, USA: IEEE, 2010; pp.2544–2550. doi: 10.1109/CVPR.2010.5539960.

[7] Henriques J, Rui C, Martins P, Batista J. Exploiting the circulant structure of tracking-by-detection with kernels. In: European Conference on Computer Vision-ECCV 2012. Springer-Verlag, 2012; pp.702–715. doi: 10.1007/978-3-642-33765-9_50.

[8] Liu T, Wang G, Yang Q. Real-time part-based visual tracking via adaptive correlation filters. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, MA, USA: IEEE, 2015; pp.4902–4912. doi: 10.1109/CVPR.2015.7299124.

[9] Yao R, Xia S X, Shen F M, Zhou Y, Niu Q. Exploiting spatial structure from parts for adaptive kernelized correlation filter tracker. IEEE Signal Processing Letters, 2016; 23(5): 658–662.

[10] Li Y, Zhang Y F, Xu Y L, Wang J B, Miao Z. Robust scale adaptive kernel correlation filter tracker with hierarchical convolutional features. IEEE Signal Processing Letters, 2016; 23(8): 1136–1140.

[11] Ma C, Xu Y, Ni B B, Yang X K. When correlation filters meet convolutional neural networks for visual tracking. IEEE Signal Processing Letters, 2016; 23(10): 1454–1458.

[12] Mei X, Ling H. Robust visual tracking and vehicle classification via sparse representation. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2011; 33(11): 2259–2272.

[13] Li H X, Shen C H, Shi Q F. Real-time visual tracking using compressive sensing. In: 2011 IEEE Computer Vision and Pattern Recognition. Colorado Springs, CO, USA: IEEE, 2011; pp.1305-1312. doi: 10.1109/CVPR.2011.5995483.

[14] Wang S, Lu H, Yang F, Yang M H. Superpixel tracking. In: 2011 International Conference on Computer Vision. Barcelona, Spain: IEEE, 2011; pp.1323–1330. doi: 10.1109/ICCV.2011.6126385.

[15] Ross D A, Lim J W, Lin R S, Yang M H. Incremental learning for robust visual tracking. International Journal of Computer Vision, 2008; 77(1-3): 125–141.

[16] Wu Y, Lim J W, Yang M H. Online object tracking: A benchmark. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition. Portland, OR, USA: IEEE, 2013; pp.2411–2418. doi: 10.1109/CVPR.2013.312.

[17] Xia R L, Chen Y T, Ren B B. Improved anti-occlusion object tracking algorithm using Unscented Rauch-Tung-Striebel smoother and kernel correlation filter. Journal of King Saud University-Computer and Information Sciences, 2022; 34(8): 6008–6018.

[18] Danelljan M, Häger G, Khan F. Accurate scale estimation for robust visual tracking. In: British Machine Vision Conference. 2014; 65p.

[19] Zhang K, Zhang L, Liu Q, Zhang D, Yang M H. Fast visual tracking via dense spatio-temporal context learning. In: Computer Vision–ECCV 2014. Cham: Springer, 2014; pp.127–141. doi: 10.1007/978-3-319-10602-1_9.

[20] Guo D Q, Zhang G X, Neri F, Peng S, Yang Q, Liu P. An adaptive kernelized correlation filters with multiple features in the tracking application. Journal of Visual Communication and Image Representation, 2022; 84: 103484.

[21] Yan P M, Yao S B, Zhu Q Y, Zhang T, Cui W N. Real-time detection and tracking of infrared small targets based on grid fast density peaks searching and improved KCF. Infrared Physics & Technology, 2022; 123: 104181.

[22] Du F, Wang W L, Zhang Z. Target tracking algorithm for pedestrians movement based on kernel-correlation filtering. Enterprise Information Systems, 2022; 16(10-11): 1500–1514.