Object tracking algorithm for unmanned surface vehicle based on improved mean-shift method

Zuquan Xiang¹,², Tao Tao¹,², Lifei Song¹,², Zaopeng Dong¹,², Yunsheng Mao¹,², Shixin Chu¹,² and Hanfang Wang³

Abstract
The unmanned surface vehicle has the characteristics of high maneuverability and flexibility. Object detection and tracking skills are required to improve the ability of unmanned surface vehicle to avoid collisions and detect targets on the surface of the water. Mean-shift algorithm is a classic target tracking algorithm, but it may fail when pixel interference and occlusion occur. This article proposes a tracking algorithm for unmanned surface vehicle based on an improved mean-shift optimization algorithm. The method uses the self-organizing feature map spatial topology to reduce the interference of the background pixels on the target object and predicts the center position of the object when the target is heavily occluded according to the extended Kalman filter. First, a self-organizing feature map model is built to classify pixels in a rectangular frame and the background pixels are extracted. Then, the method optimizes the extended Kalman filter solution process to complete the prediction and correction of the target center position and introduces a similarity function to determine the target occlusion. Finally, numerical analyses based on a ship model sailing experiment are performed with the help of OpenCV library. The experimental results validated that the proposed method significantly reduces the cumulative error in the tracking process and effectively predicts the position of the target between continuous frames when temporary occlusion occurs. The research can be used for target detection and autonomous navigation of unmanned surface vehicle.

Keywords
Unmanned surface vehicle, object tracking, mean-shift, self-organizing feature map, extended Kalman filter, pixel interference object occlusion, obstacle avoidance

Date received: 26 January 2020; accepted: 16 April 2020

Introduction
Environmental awareness is essential for unmanned surface vehicle (USV) to acquire the autonomous ability in changing environments. Target detection and object tracking skills are its foundation. Due to the interference factors such as background interference, object occlusion, and noise clutter, object tracking on water is a challenging task. Object tracking plays a vital role in the computer vision field. The tracking can be divided into region

¹ Key Laboratory of High Performance Ship Technology, Wuhan University of Technology, Ministry of Education, Wuhan, People’s Republic of China
² Transportation School, Wuhan University of Technology, Wuhan, People’s Republic of China
³ Wuhan Vocational College of Software and Engineering, Wuhan, People’s Republic of China

Corresponding authors:
Yunsheng Mao and Lifei Song, Key Laboratory of High Performance Ship Technology, Wuhan University of Technology, Ministry of Education, Wuhan 430063, People’s Republic of China.
Emails: ysmao@163.com; songlifei@whut.edu.cn
tracking, feature tracking, model tracking, and deformation template tracking. In 2000, to track objects, Comaniciu et al. applied the mean-shift algorithm, which is simple and practical, and fully effective when the target moves slowly. However, the target may lose when the method is applied on water due to the environmental interference and the cumulative error of the algorithm.

In order to promote the deficiency of the mean-shift method, several improved algorithms have been suggested. Collins introduced a method based on the difference of Gaussian mean-shift kernel to effectively track one object through the scale space. The tracking effect was satisfactory but the time complexity was high when the object moves at low speed. Danelljan et al. incorporated a factorized convolution operator in the discriminative correlation filter, the operations of reducing the number of model parameters and training the compact model generated by the sample distribution could also significantly reduce the complexity of memory and time, but the target will easily be lost if it was severely occluded. Ignoring the effects of sea clutter interference, Mohanty et al. proposed a method of determining the background pixel position based on the maximum gradient value in the column direction after the image information was proposed and averaged in the row direction. The algorithm had extreme requirements on the environment, and it was liable to fail when there was a large background interference. Dong and Desouza proposed a new adaptive learning algorithm that uses multiple feature subspaces to handle background pixel mutations caused by light changes, but its accuracy was not very penetrating. For the occlusion problem in multiple environments, Possegger et al. enhanced the model’s recognition accuracy and matching efficiency by adding basic color features to the target appearance model, accurately distinguished the target, and improved the robustness of the algorithm. Nevertheless, the algorithm was difficult to implement when objects were occluded. The scale and orientation adaptive mean-shift tracking (SOAMST) algorithm is another improved algorithm based on mean-shift, which enhances the adaptive ability of mean-shift when the target is deformed. When the value in the weighted image is small, the size of the candidate object area will be close to that of the actual target, the scale and size of the target in motion can be obtained by calculating the gray matrix related to the weight image. SOAMST is a relatively new algorithm at present.

To overcome the shortcomings of mean-shift algorithm tracking, many novel methods use Kalman filter (KF) to predict and correct target position based on the framework of mean-shift algorithm. Li constructed a KF system model and used the initial value of the mean-shift algorithm at the target center of the KF prediction. The results of the tracking algorithm were used as adaptive KF measurement values. The KF estimation parameters were fitted by the Bhattacharyya coefficient. The algorithm was practical, but it still failed when object occlusion occurs. Gu used a Gaussian mixture weighted KF in the tracking process to enhance the performance of the algorithm and avoid unnecessary maneuver detection, but the algorithm lacked stability. The KF can effectively improve the tracking accuracy, but it requires the dynamic characteristics of the system to be linear in basic parameters, which is a strict condition.

In addition, with the development of neural networks, they are commonly used in image segmentation, feature point extraction, and other technologies. Self-organizing feature map (SOFM) is a competitive neural network, with a two-layer neural network consisting of an input layer and an output layer. In the network, the output layer is spatial topological structure. SOFM has been widely used in image processing and classification owning to the unique structure. Its outstanding performance in image classification can effectively address the problem of pixel interference in tracking algorithms, and its nearest neighbor rule can effectively complete the pixel classification.

As for the problems that background pixel interference and target loss while occlusion occurred, this article proposes an improved mean-shift optimization algorithm based on SOFM and extended Kalman filter (EKF), which can improve the tracking accuracy while ensuring the anti-occlusion ability. This algorithm classifies the images in the rectangular frame according to the spatial topology of the SOFM, uses the nearest neighbor to select the most likely target image, removes the off-center image, and reduces the influence of the background pixels. Then, the EKF is differentiated and integrated into the mean-shift algorithm. After detecting the occlusion of the object, the new center position of the target is obtained based on the dynamic prediction ability of the EKF. The rest of the article is arranged as follows: the second section introduces the basic principles of mean-shift algorithm, the third section introduces SOFM-based target detection algorithms, the fourth section introduces the target positioning process based on improved EKF, the fifth section describes the experimental results, and the sixth section summarizes the conclusions.

**Mean-shift tracking algorithm**

Mean-shift tracking algorithm is a nonparametric feature space analysis method built on kernel density estimation. As color is the most widely used feature in computer vision problems, mean-shift uses the color histogram as its matching feature for tracking objects. The target model is constructed based on the color characteristics of the current frame. If the target still exists in the next frame area during the tracking process, it will be defined as the region proposal. Traditionally, mean-shift tracking algorithm calculates the color histogram of the target model and the candidate model separately and compares the similarity of the two models according to the Bhattacharyya coefficient to complete the feature module matching. During the
process of object tracking, the initial object is selected manually in the initial frame, and a rectangular window with a tracked object is drawn by manual annotation.\(^{14,15}\)

**Target model**

During the selection of rectangular box in the initial frame, it is assumed that \(n\) pixels are contained, every pixel position is represented by \(\{z_i\}_{i=1,2,...,n}\), and the color space of the selected area is evenly divided to find a feature histogram composed of \(m\) equal intervals. The probability density \(q_u(u = 1,2,\ldots,m)\) of the target model is expressed as follows

\[
q_u = C \sum_{i=1}^{n} K(||z_i^*||^2) \delta(b(z_i) - u) \tag{1}
\]

where \(z_i^*\) indicates the normalized pixel position with object center as origin, calculated by the following equation

\[
z_i^* = \left(\frac{(x_i - x_0)^2 + (y_i - y_0)^2}{x_0^2 + y_0^2}\right)^{0.5} \tag{2}
\]

\((x_0, y_0)\) is the target center coordinate, \(K\) is the Epanechnikov kernel function matrix, \(b(z_i)\) is the histogram index function, which is the red, green, blue (RGB) value corresponding to each pixel in the target frame, \(u\) is a certain eigenvalue (\(u = 1,\ldots,m\)), the function \(\delta\) is used to determine whether the gray value at the pixel \(z_i\) in the target area belongs to the \(u\)th cell in the histogram, if yes, it is 1, otherwise it is 0. \(C\) is the normalization coefficient, which is obtained from the following equation

\[
C = \frac{1}{\sum_{i=1}^{n} K(||z_i^*||^2)} \tag{3}
\]

**Candidate model**

At the \(t\)th frame, the target center position \(f_0\) of the \((t-1)\)th frame is used as the center of the search window to obtain the center position coordinate \(f\) of the candidate target, and the region proposal histogram of the current frame is calculated. The pixels in this area are represented by \(\{z_i\}_{i=1,2,...,n}\). The probability density of the candidate model can be solved by the following equation

\[
p_u(f) = C \sum_{i=1}^{n} K\left(||\frac{f - z_i}{h}||^2\right) \delta(b(z_i) - u) \tag{4}
\]

where \(h\) is the size of the kernel window and determines the weight distribution.

**Similarity measure**

Bhattacharyya coefficient is the similarity function, which is defined as

\[
\rho(p,q) = \sum_{u=1}^{m} \sqrt{p_u(f)q_u} \tag{5}
\]

The similarity function \(\rho\) ranges from 0 to 1, the larger value indicates that the two models are more similar.

**Mean-shift iteration**

During the iteration of each frame, the center position of the candidate model is continuously updated until the Bhattacharyya coefficient reaches the maximum value. As for calculating the histogram distribution of the search window under the weight of the kernel function, it uses Hill-climbing algorithm for the density histogram of the data in the current frame and iteratively calculates the data histogram multiple times to make mean-shift find the target area with the greatest similarity along the direction of the data density distribution gradient.

**Target detection based on SOFM**

In most cases, background information will inevitably appear in the rectangular frame that is manually selected. Therefore, when the traditional mean-shift algorithm uses the color histogram of the target as the target’s feature representation, it is easy to cause background interference and target occlusion. The SOFM is introduced into the traditional mean-shift target tracking algorithm to detect target pixels using the time continuity of SOFM and the constraints of the spatial topology.

By setting up SOFM, the clustering of various image slices in the rectangular box is carried out to complete the separation of the background interference pixels. After the image is being divided into several image pieces, these input data are mainly classified into three characteristics, namely, the image film \((p_1)\) which does not contain the target pixels at all, the image chip \((p_2)\) which contains the target pixels, and the image chip \((p_3)\) which contains both the target pixels and the background pixels. The overall process is as follows.

**Argument initialization**

A set of weights is set by random numbers, the weights \(w_k = [w_{k1}(t), w_{k2}(t), w_{k3}(t), \ldots, w_{k4}(t)]\), and they are normalized. Set the learning-rate-time function as \(T_1\), the neighbor-radius-time constant as \(T_2\), and the number of neurons in the competitive layer as \(m\).

**Enter the input of the vector**

The target box image is divided into several image slices, and the image feature vector \(X = (x_1, x_2, x_3, \ldots, x_m)^T\) is entered into the input layer of the network. And the input vector is normalized. In the mapping layer, the Euclidean distance of the weight vector and input vector of various
neurons is calculated, and the two are made internal products. The neuron with the largest inner product is marked as the winning neuron. Since the sample vectors and weights are normalized, the neuron with the smallest Euclidean distance is the one with the largest inner product. The distance between the \( j \)th neuron and the input vector of the mapping layer is as follows

\[
D = \|X - W_j\| = \sqrt{\sum_{i=1}^{m} (x_i(t) - w_{ij}(t))^2}
\]  

(6)

where \( w_{ij} \) is the weight between \( i \)th neurons in the input layer and \( j \)th neurons in the mapping layer.

**Weight learning**

Update weights of the winning neuron topology area with the Kohonen rule

\[
w_{ij}(t+1) = \begin{cases} 
  w_{ij}(t) + \eta(x_i(t) - w_{ij}(t)), & j \in N(t) \\
  w_{ij}(t), & j \notin N(t)
\end{cases}
\]  

(7)

where \( \eta \) is a constant between 0 and 1, gradually reduced to 0 over time; \( N(t) \) is the neighborhood of winning neuron.

**Judgment of algorithm convergence**

The algorithm determines whether the number of iterations has reached the preset maximum. If the maximum number of iterations has not been reached, it proceeds to the second step, otherwise, the algorithm ends.

Image slice is a collection of regional image pixels with structure information, which uses SOFM to classify the image slices in rectangular boxes and maintain its spatial topology. The image slice that is most likely to be targeted is determined using the proximity selection mechanism, and the image piece that deviates from the center of the target will be removed. Target center image slices and images containing targets and targetless images are weighted to determine the center position of the next target frame:

\[
y^0_{i,j+1} = \sum_{i=1}^{n} w_i P_i
\]  

(8)

where \( P_i \) is the center coordinates of the \( i \)th image and \( w_i \) is its weight. The algorithm flow is shown in Figure 1.

**Object localization based on EKF**

Object tracking is a dynamically changing process, whose eigenvalues will change differently over time. When the state at \( (t+1) \)th frame is not a linear function of the state at \( t \)th frame, we need to use the local characteristics of the nonlinear function to locally linearize the correlation process and evolve the KF into EKF, which can achieve the optimal estimation of the target center position.

The dynamic characteristics of the object are selected through EKF as input parameters, and the eigenvalue at the next moment is determined according to the change of the input.

The system state equation is as follows

\[
X_t = f(X_{t-1}, u_t) + q_1
\]  

(9)

The system observation equation is as follows

\[
Z_t = h(X_t, u_t) + q_2
\]  

(10)

State transition matrix is

\[
F_t = \frac{\partial f(X)}{\partial X} \bigg| X = X_{t-1}
\]

and observation matrix is

\[
H_t = \frac{\partial h(X)}{\partial X} \bigg| X = X_{t-1}
\]

where \( X_t \) is the system random noise vector, \( Z_t \) is the system random observation vector, \( f(\cdot) \) is the system prediction function, \( h(\cdot) \) is the system observation function, \( u_t \) is set to 0 generally, and \( q_1 \) is the estimation error and \( q_2 \) is the observation error, both of which are zero-mean vectors with Gaussian distribution. The covariance matrices of system state noise and observation noise are \( Q \) and \( R \), respectively.

EKF predicts the state of the next moment according to the state of the previous moment, then combines the difference between the observed value and the predicted value to modify the predicted value by the filter gain matrix. In a word, the essence of EKF is to give the dynamic system state value at the current moment through the optimal estimated value at the previous moment and the actual observed value. The EKF process is shown in Figure 2.
During the updated process, EKF needs to solve the Jacobian matrix at each moment, which increases the time complexity of the algorithm. This article uses the differential linearized EKF method.\textsuperscript{18} We solve the Taylor series expansion of the function \(y = f(x)\) in the neighborhood of \(\bar{x}\), and keep only the first term of the expansion

\[
y = f(\bar{x}) + \frac{\partial f}{\partial x} |_{x=\bar{x}} (x - \bar{x})
\]  

(11)

In high-dimensional nonlinear systems, the form of the Jacobian matrix is generally used to express the derivative of the function \(f(x)\). This article uses the mean value of state transitions to predict time constant. The object area state vector is

\[
X = [x, y, v_x, v_y, w_x, h_t, \Delta w_t, \Delta h_t]
\]

During mean-shift algorithm object tracking, an initial position needs to be set before each iteration calculation. Considering the dynamic system estimation capability of EKF, the starting position of the iteration can be provided by EKF prediction, which reduces the running time of mean-shift algorithm and improves the system processing capability. Due to the temporal continuity of the video image, the position and speed of the target between continuous frames will not change unexpectedly,\textsuperscript{19} so the movement of objects between two continuous frames can be regarded as a linear equation. This article uses the mean value of state transitions to predict time. The state transition matrix is

\[
F_{t|t-1} = \begin{bmatrix}
1 & 0 & dt & 0 \\
0 & 1 & 0 & dt \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  

(14)

The observation function is defined as

\[
Z_t = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{bmatrix}
\]  

(15)

The state noise matrix is

\[
Q_t = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  

(16)

When the target object is not blocked, the tracking steps are as follows:

1. Using the optimal estimation of the object center position of the previous frame \(X_{t-1|t-1}\) to solve the prediction of the position \(X_{t|t}\) of the next frame.
2. Iteratively calculating the object center position in the current frame as the EKF observation according to the position of the previous frame \(Z_t\).
3. Using \(Z_t\) to modify the predicted value, then calculating the nearest estimate position \(x_{t|t}\) for the current frame.
4. Continue iterating with \(x_{t|t}\) as the starting point until the target object is tracked, it is regarded as the center position of the target in the current frame, and the estimation ends.

When the object is occluded, the mean-shift algorithm cannot calculate the correct center position, which gives birth to the condition that EKF is not able to obtain the true observation value. A threshold \(G_t\) is presented to determine whether the target is out of occlusion. Similarly, two continuous frames are selected for comparison. The Bhattacharyya coefficient is introduced to determine whether occlusion occurs during object tracking. In this article, \(P_t\) and \(P_{t-2}\) are defined as the Bhattacharyya coefficients of the current frame and the previous frame, respectively, and the threshold \(G_t\) determines the occlusion degree in the \(t\)th frame

\[
f = \begin{cases} 
1, \text{ occlusion} & \frac{P_t}{P_{t-2}} \leq G_t \\
0, \text{ out of occlusion} & \frac{P_t}{P_{t-2}} > G_t 
\end{cases}
\]  

(17)

where \(G_t\) is set to 0.3.

When the object is out of occlusion, the mean-shift algorithm starts to iterate, and the EKF is provided with valid observations from the target’s true center position. Combined with the description in this section, the specific algorithm flowchart is shown in Figure 3.

**Experiment design and result analysis**

The SOFM algorithm can be regarded as a process of classifying image pieces in a rectangular frame. According to the object size in the actual test image, the network dimension is set to \(3 \times 5\), the initial training step covering the input space is 100, the initial neighborhood size is 3, the layer topology function uses the \textit{hextop} function, and the distance function between neurons is \textit{linkdist} function.
Figure 3. Object localization flow based on EKF. EKF: extended Kalman filter.

Figure 4. The experiment result of SOFM. SOFM: self-organizing feature map.
The experiment program is developed with C++ based on Visual Studio 2017 and the OpenCV 3.4.6 computer vision function library was used. As Figure 4 shows, the program running result of a sailing USV video well reflects the space topology structure of the SOFM algorithm. As described in the third section, the testing picture is divided into $3 \times 3$ slices, and they are classified according to the features p1, p2, and p3. Images 2 and 3, which include the object pixel, and images 1 and 4, which do not include the object pixel at all are labeled, respectively. As shown in the classification results, two neighboring image slices will be allocated to two neurons adjacent to each other in space. Using the spatial structure of the SOFM algorithm, the adjacent image slices are analyzed as an entirety to deal with the influence of background interference.

Under the same development environment, an outdoor USV sailing video is used to conduct an object occlusion test experiment. In order to guarantee the reliability of the experiment, the improved algorithm is compared with the SOAMST algorithm. In the comparison experiments, the red rectangular frame is the manually selected rectangular frame in the improved algorithm, while the black rectangular frame is the manually selected rectangular frame in the SOAMST algorithm. In the video, the floating box on the water surface will cause a complete occlusion to the USV, and the background interference and reflection interference challenge the algorithm too. The tracking results are shown in Figure 5. As the first three figures show, in the condition of sailing with slow speed, tracking effects of both two algorithms are ideal, however, the SOAMST algorithm gives rise to some fluctuations while the improved algorithm has less accumulated errors due to the application of SOFM. When the USV is occluded, as showed in the next three pictures, the SOAMST algorithm loses the target, while the improved...
algorithm accurately estimated the target position due to the prediction ability of the EK. After the disappearance of occlusion, the improved algorithm still finds the target correctly.

In Figure 6, the blue continuous curve describes the pixel distance error of the SOAMST algorithm during the experience, and the orange dotted curve represents the error of the optimized algorithm. Due to the cumulative error of the algorithm and the influence of occlusions, the tracking result of SOAMST algorithm becomes worse and worse over time. In addition, the algorithm starts EKF position prediction when the object is occluded, which effectively shortens the overall running time.

Conclusion
This article analyzes the advantages and disadvantages of the mean-shift algorithm and proposes optimization ideas for the cumulative error and background interference problems. First, the spatial topological features of the SOFM classification algorithm is used to eliminate image slices with background interference and to reduce background interference factors. The EKF’s dynamic system estimation capabilities is used to accurately predict the position of the next frame. By incorporating EKF into the traditional mean-shift algorithm, a new target center position can be and efficiently obtained, and the accurate position of the target can be found. The algorithm is proved in the experiments that the background pixel interference can be effectively narrowed. When the occlusion time is short, the algorithm has an excellent anti-occlusion effect. In further research, the similarity function of occlusion determination needs to be optimized to improve the universal applicability of the algorithm.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship and/or publication of this article: The funding of this work was supported by the National Natural Science Foundation of China (NSFC: 51809203 and 51709214) and the Fundamental Research Funds for the Central Universities (WUT: 2019IVB011, 2017IVA008, and 2017IVA006).

ORCID iDs
LiFei Song https://orcid.org/0000-0002-8994-781X
Zaopeng Dong https://orcid.org/0000-0002-8909-8395
Yunsheng Mao https://orcid.org/0000-0002-9266-0369

References
1. Song LF, Chen Z, Dong ZP, et al. Collision avoidance planning for unmanned surface vehicle based on eccentric expansion. Int J Adv Robot Syst 2019; 16. DOI: 10.1177/1729881419851945.
2. Manley JE. Autonomous surface vessels, 15 years of development. In: Proceedings oceans 2008 MTS/IEEE Quebec conference and exhibition, Quebec, 2008, pp. 1–4.
3. Motwani M, Tirpankar N, Motwani R, et al. Towards benchmarking of video motion tracking algorithms. In: 2010 International conference on signal acquisition and processing, ICSAP 2010, Bangalore, India, 9–10 February 2010, pp. 215–219. IEEE Computer Society.
4. Comaniciu D, Ramesh V, and Meer P. Real-time tracking of non-rigid objects using mean shift. IEEE Comput Soc Conf Comput Vis Pattern Recognit 2000; 2(2): 142–149.
5. Whoang I, Kim JH, and Choi KN. Object tracking using maximum colour distance under illumination change. Int J Adv Robot Syst 2019; 9: 212.
6. Angelov P. Autonomous novelty detection and object tracking in video streams. Hoboken: John Wiley & Sons, Ltd, 2012.
7. Collins RT. Mean-shift blob tracking through scale space. In: IEEE computer vision & pattern recognition, Madison, USA, 18–20 June 2003, pp. 234–240. IEEE.
8. Daneljan M, Bhat G, Khan FS, et al. ECO: efficient convolution operators for tracking. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, HI, 21–26 July 2017, pp. 6931–6939. IEEE.
9. Dong Y and Desouza GN. Adaptive learning of multi-subspace for foreground detection under illumination changes. Comput Vision Imag Understand 2011; 115(1): 31–49.
10. Possegger H, Mauthner T, and Bischof H. In defense of color-based model-free tracking. Comput Vision Imag Understand 2015; 2131–2120.
11. Li XH. Object tracking using an adaptive Kalman filter combined with mean shift. Opt Eng 2010; 49(2): 020503.
12. Gu GS. Maneuvering target tracking based on class-dependent tracking gates. Opto-Electron Eng 2016; (02).
13. Ristic D M, Pavlovic M, and Reljin I. Image segmentation method based on self-organizing maps and K-means algorithm. In: 9th Symposium on neural network applications in electrical engineering, 2008, NEUREL 2008, Belgrade, Serbia, 25–27 September 2008. IEEE.
14. Khan ZH, Gu IY, and Backhouse AG. Robust visual object tracking using multi-mode anisotropic mean shift and particle filters. IEEE Trans Circuits Syst Video Technol 2011; 21(1): 74–87.
15. Tu JL, Tao H, and Huang TS. Online updating appearance generative mixture model for meanshift tracking. In: Computer vision – ACCV 2006, 7th Asian conference on computer vision, Hyderabad, India, 13–16 January 2006, Proceedings, Part I. Berlin: Springer.
16. Strom J, Jebra T, Basu S, et al. Real time tracking and modeling of faces: an EKF-based analysis by synthesis approach. In: Proceedings of the IEEE international workshop on modelling people, Kerkyra, Greece, 20 September 1999. IEEE.
17. Zhou JC. EKF based object detect and tracking for UAV by using visual-attention-model. In: 2014 International conference on progress in informatics and computing (PIC), Shanghai, China, 16–18 May 2014. IEEE.

18. Zhou WJ and Bo YM. Research of EKF based on differential linearization. Comput Eng Appl 2009; 45(9): 64–66.

19. Danelljan M, Bhat G, Khan FS, et al. Beyond correlation filters: learning continuous convolution operators for visual tracking. In: European conference on computer vision, 16 September 2016, pp. 472–488. Cham: Springer.

20. Kaehler A and Bradski G. Learning OpenCV 3: computer vision in C++ with the OpenCV library. Sebastopol: O'Reilly Media, Inc, 2016.

21. Liang X, Qu X, Hou Y, et al. Distributed coordinated tracking control of multiple unmanned surface vehicles under complex marine environments. Ocean Engineering, 2020; 205(107328): 1–9.