MDPI

*Article*

# Tracking-by-Detection Algorithm for Underwater Target Based on Improved Multi-Kernel Correlation Filter

**Wenrong Yue** [1,2]**, Feng Xu** [1] **and Juan Yang** [1,]***

1    Ocean Acoustic Technology Laboratory, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China; xf@mail.ioa.ac.cn (F.X.)
2    University of Chinese Academy of Sciences, Beijing 100049, China
*    Correspondence: yangjuan@mail.ioa.ac.cn; Tel.: +86-15910836396

**Abstract:** Joint detection and tracking of weak underwater targets are challenging problems whose complexity is intensified when the target is disturbed by reverberation. In the low signal-to-reverberation ratio (SRR) environment, the traditional detection and tracking methods perform poorly in tracking robustness because they only consider the target motion characteristics. Recently, the kernel correlation filter (KCF) based on target features has received lots of attention and gained great success in visual tracking. We propose an improved multi-kernel correlation filter (IMKCF) tracking-by-detection algorithm by introducing the KCF into the field of underwater weak target detection and tracking. It is composed of the tracking-by-detection, the adaptive reliability check, and the re-detection modules. Specifically, the tracking-by-detection part is built on the multi-kernel correlation filter (MKCF), and it uses multi-frame data weighted averaging to update. The reliability check helps keep the tracker from corruption. The re-detection module, integrated with a Kalman filter, identifies target positions when the tracking is unreliable. Finally, the experimental data processing and analysis show that the proposed method outperforms the single-kernel methods and some traditional tracking methods.

**Keywords:** active sonar; low signal-to-reverberation ratio; underwater target; tracking by detection; kernel correlation filter

## 1. Introduction

Underwater target detection and tracking in active sonar systems has always been a hot topic in underwater applications. The conventional approach to detect and track underwater targets involves threshold detection, followed by data association and filtering tracking [1–4]. However, practical sonar systems often encounter strong reverberation interference. In this low signal-to-reverberation ratio (SRR) environment, only setting a lower threshold can ensure that the target is not missed, but it also causes a lot of false alarms [5,6]. The higher false alarm rate adversely affects target associations, thereby increasing the risk of tracker drift during the tracking process.

In order to solve the problem of weak target detection and tracking in low SRR, the methods commonly used at present can be categorized into three groups. The first group involves traditional data association methods, such as joint probabilistic data association (JPDA) [7,8] and multiple hypothesis tracking (MHT) [9,10]. However, these approaches suffer from high computational costs when confronted with a high false alarm rate. The second group focuses on methods based on random finite set (RFS) [11,12], which eliminate the need for data association. These methods employ filtering techniques based on the motion characteristics of the target. A crucial aspect of accurate tracking filtering is establishing an appropriate motion model that aligns with the target's motion type. The filtering algorithms with multiple models (MMs) and the jump Markov system (JMS) have been shown to be effective approaches for maneuvering target tracking [13–15]. In addition,

Yue et al. established a multi-directional motion model set according to the motion characteristics of the diver target [16]. However, due to the diverse underwater target motion types, it is challenging to apply a single modeling method to all underwater targets of interest. Furthermore, in environments with reverberation interference, the aforementioned tracking methods face difficulties in accurately determining the target's position solely based on trajectory information. The third group involves deep learning methods, such as Convolutional Neural Network (CNN) [17], Recurrent Neural Network (RNN) [18], and Siamese Network [19]. These approaches typically require a substantial number of samples for model training. However, in practical applications, it may not be feasible to collect a sufficient amount of sample data.

Nevertheless, the kernel correlation filter (KCF) algorithm, proposed by Joao F. Henriques et al., presents a promising solution for tracking multiple target types without the need for predefined target motion models [20,21]. Presently, the KCF algorithm is primarily used in the visual tracking field [22], and there has been no prior instance of its application in underwater target tracking domestically or internationally. Hence, the primary contribution of this study lies in the application of the KCF algorithm to resolve the detection and tracking challenges associated with underwater weak targets.

The effectiveness of single-feature-based tracking is limited due to the absence of prior knowledge about the target in the model-free kernelized correlation filter (KCF) algorithm [23–26]. To enhance robustness in tracking, researchers have explored multi-feature fusion tracking methods [27–30], which leverage a shared kernel function with multiple complementary features. However, these methods face challenges in achieving the optimal solution because different features may require distinct kernel functions. To adaptively use multiple complementary features, Tang et al. introduced multi-kernel learning (MKL) into the correlation filtering algorithm to dynamically update multiple nonlinear kernels, namely the multi-kernel correlation filter (MKCF) [31]. However, the MKCF algorithm only utilizes adjacent frame information for filter updates, which could lead to model update errors in the presence of reverberation occlusion. The second contribution of this paper is to utilize weighted information from historical samples to adaptively solve the parameters of multiple nonlinear kernels and make full use of multiple complementary features to enhance the robustness and tracking accuracy of the long-term tracking process.

In scenarios with low SRR, the involvement of non-target information in training frames may result in error propagation during the model update phase, increasing the risk of drift. Thus, it becomes crucial to assess the reliability of tracking results and identify a more dependable result when the tracking outcome is unreliable.

In terms of assessing the reliability of tracking results, Bolme [32] computed the peak-to-sidelobe ratio (PSR) score of the relevant response, comparing it with a fixed threshold to determine reliability. However, this method exhibited limited effectiveness in complex environments. Wang et al. (their tracker is abbreviated as RRLT) [33] proposed a more effective reliability criterion for evaluating the confidence of the current tracking result. This criterion adaptively updates the mean value of multi-frame PSR scores as a threshold, thereby improving the accuracy of evaluation results in complex environments. Regarding tracking methods, the long-term correlation tracker (LCT) employs an online random fern classifier to generate potential target locations [34], while Wang [33] utilizes a particle filter to generate numerous candidate target positions around the previous frame's target position. Nonetheless, neither LCT nor the random RRLT tracker can successfully re-detect a target that has been obscured for an extended period [35]. The third contribution of this paper entails a real-time assessment of target tracking result reliability and proposes an effective re-detection module. The reliability check module adopts the approach outlined by Wang et al. [33] to evaluate the reliability of both detection and tracking results obtained using the MKCF. When the tracking result is deemed unreliable, the re-detection module utilizes the historical reliability tracking result to drive the Kalman filter, predicting the target candidate position. Subsequently, several candidate positions are generated around

this predicted position, following a Gaussian distribution. Finally, a stringent replacement criterion is applied to determine the final tracking result.

In summary, this paper presents an improved multi-kernel correlation filter (IMKCF) algorithm for robust detection and tracking of weak underwater targets. A novel adaptation of the KCF algorithm from visual tracking to the domain of underwater multi-motion weak target detection and tracking is proposed. To address the issue of limited robustness in single-feature tracking, the weighted information from historical samples is utilized to adaptively resolve the coefficients of multiple nonlinear kernels. The MKCF algorithm is analyzed from a maximum likelihood perspective to determine the target position based on the maximum likelihood criterion. Real-time estimation of the target tracking result reliability is performed, and an effective re-detection module is introduced. The efficacy of the algorithm is validated through the analysis of sea trial data.

The rest of this article is organized as follows: Section 2 introduces the target model and sonar measurement model. Section 3 reviews the KCF algorithm. Section 4 introduces the framework of the IMKCF tracking-by-detection algorithm and introduces the components of each module in detail. Section 5 analyzes the performance of the algorithm by processing experimental data. Section 6 summarizes the work of this paper.

## 2. Model Establishment

### 2.1. Target Model

Given the position of the target $x^k$ at time $k$ denoted by $x_{p_k}$ and $y_{p_k}$, and the corresponding velocities $v_{x_k}$ and $v_{y_k}$, the state of the target can be represented by $x^k = \left[ x_{p_k}, y_{p_k}, v_{x_k}, v_{y_k} \right]$. The evolution of $x^k$ is formulated as a first-order Markov process,

$$x^{k|k-1} = p\left( x^k \middle| x^{k-1} \right) \tag{1}$$

where the $p$ is the a priori probability density function. The specific form of $p$ is determined by the target model.

### 2.2. Measurement Model

The algorithm employs raw sonar data measurements in range-azimuth format. When the influence of noise is disregarded, the correlation between the range, angle of the sonar, and position of the target is established.

$$\begin{cases} r_k = \sqrt{x_{p_k}^2 + y_{p_k}^2} \\ \theta_k = \arctan\left( \frac{x_{p_k}}{y_{p_k}} \right). \end{cases} \tag{2}$$

At time $k$, the resolution of the measurement area is $N_r \times N_b$. With the sonar position serving as the reference point or origin, the measurement area is characterized by a distance range $[R_{min}, R_{max}]$, which is discretized into $N_r$ distance units, and an azimuth range $[\theta_{min}, \theta_{max}]$, which is discretized into $N_b$ azimuth units.

$$N_r = \frac{2(R_{max} - R_{min})}{c} \times F_s, \tag{3}$$

where $F_s$ is the sampling frequency, and $N_b$ can be determined by the azimuth resolution unit $\Delta\theta$. The element $z_k(x, y)$ of the $x$th azimuth and $y$th distance cell in the measurement $z_k$ is the signal echo intensity. $z_k(x, y)$ can be modeled as

$$z_k(x, y) = \begin{cases} a_k h(x, y; x_k) + w_k & \text{if target is in } (x, y) \\ w_k & \text{otherwise} \end{cases}, \tag{4}$$

where $a_k$ is the peak amplitude of the target, $h$ is the point spread function, and $w_k$ is the measured noise and reverberation of the sonar system at moment $k$.

### 3. Preliminaries of the KCF Algorithm

The KCF algorithm involves three steps: training, detection, and updating. In the training step, it aims to optimize the correlation filter parameter using the training feature–label pairs $\{x_i, y_i\}_{i=1}^m$. It maps the input $x$ to a new space $\varphi(x)$ with higher dimensions and puts the $\varphi(x)$ into the optimization process. The kernel function $\kappa$ and the objective of optimization are as follows.

$$\kappa(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle. \tag{5}$$

$$\min_w \sum_{i=1}^m \left( y_i - \sum_{j=1}^m a_j \kappa(x_i, x_j) \right)^2 + \lambda \|w\|^2, \tag{6}$$

where $w$ is defined as

$$w = \sum_{i=1}^m a_i \varphi(x_i). \tag{7}$$

The solution of (6) is given by employing the circulant structure for fast training and testing,

$$\alpha = F^{-1}\left( \frac{F(\mathbf{Y})}{F^*(\mathbf{K}_{xx}) + \lambda} \right), \tag{8}$$

where the $*$ indicates the complex conjugate, and $F$ and $F^{-1}$ denote the Fourier transform and the inverse, respectively. $\mathbf{K}_{xx}$ denotes kernel matrix.

In the detection step, we can compute the probability of a new input $z$ being from the target feature.

$$\mathbf{Y}' = F^{-1}(F * (\mathbf{K}_{xz}) F(\alpha)). \tag{9}$$

In the updating step, the reference feature $x$ and the correlation filter parameters are calculated as follows

$$x_t = z_t \times \eta + (1 - \eta) \times x_{t-1}, \tag{10}$$

$$\alpha_t = \alpha_{z_t} \times \eta + (1 - \eta) \times \alpha_{t-1}, \tag{11}$$

where $\eta$ is the learning rate.

As above, the KCF can be performed on general machines due to its high computational efficiency. However, the performance of KCF is related to the features extracted. Making full use of multiple complementary features can improve tracking accuracy and robustness. Therefore, in our proposed algorithm, to avoid the interference of different features in the single kernel, we use MKCF to assign a kernel for each feature.

### 4. Improved MKCF Tracking-by-Detection

This study aims to tackle the challenge of detecting and tracking weak targets with various motion types in shallow sea environments. We propose an IMKCF tracking-by-detection algorithm, which consists of three components: the MKCF tracking-by-detection, the adaptive reliability check, and the re-detection modules. The overall framework of the proposed approach is depicted in Figure 1.
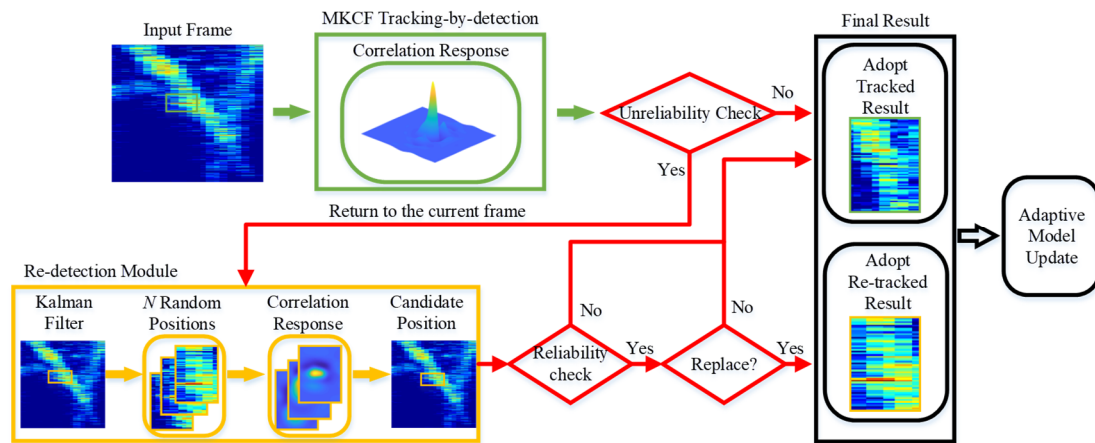
**Figure 1.** The IMKCF tracking-by-detection algorithm framework diagram.

### 4.1. The MKCF Tracking-by-Detection

Studies have shown that the incorporation of multiple kernels can improve the discriminatory ability of classifiers in comparison to a single-kernel approach [36]. A prevalent approach is to employ a basis kernel, $k_m(m = 1, \cdots, M)$, and then view $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{b}^{\mathrm{T}} \mathbf{k}(\mathbf{x}_i, \mathbf{x}_j)$ as a composition of the basis kernels, where $\mathbf{k}(\mathbf{x}_i, \mathbf{x}_j) = (k_1(\mathbf{x}_i, \mathbf{x}_j), \cdots, k_M(\mathbf{x}_i, \mathbf{x}_j))^{\mathrm{T}}$, $\mathbf{b} = (b_1, \cdots, b_m)^{\mathrm{T}}$, $\sum_{m=1}^{M} b_m = 1$, and $b_m \geq 0$. Therefore $\mathbf{K} = \sum_{m=1}^{M} b_m \mathbf{K}_m$, $\mathbf{K}_m$ is the base core of group $m$, whose elements are $k_{ij}^m = k_m(\mathbf{x}_i, \mathbf{x}_j)$. The optimization problem is used to minimize the loss,

$$\min_{\boldsymbol{\alpha}, \mathbf{b}} \frac{1}{2} \left\| \mathbf{y} - \sum_{m=1}^{M} b_m \mathbf{K}_m \boldsymbol{\alpha} \right\|_2^2 + \frac{\lambda}{2} \boldsymbol{\alpha}^{\mathrm{T}} \sum_{m=1}^{M} b_m \mathbf{K}_m \boldsymbol{\alpha} = \min_{\boldsymbol{\alpha}, \mathbf{b}} F(\boldsymbol{\alpha}, \mathbf{b})$$
$$\text{s.t.} \sum_{m=1}^{M} b_m = 1, b_m \geq 0, m = 1, \cdots, M \qquad . \tag{12}$$

The optimal solution can be expressed as Equation (13), and the * indicates the optimal.

$$f^*(\mathbf{x}) = \sum_{i=0}^{l-1} \alpha_i \mathbf{b}^{\mathrm{T}} k(\mathbf{x}_i, \mathbf{x}). \tag{13}$$

The diagram of the MKCF tracking-by-detection algorithm is presented in Figure 2.

In order to achieve localization robustness, the MKCF tracking-by-detection algorithm uses the weighted average of historical samples to update the training coefficients $\boldsymbol{\alpha}$ and $\boldsymbol{b}$. The optimization function is represented by

$$F_p(\boldsymbol{\alpha}_p, \mathbf{b}_p) \equiv \frac{1}{2} \sum_{j=1}^{p} \sum_{m=1}^{M} \beta_m^j u_{F(\boldsymbol{\alpha}, \mathbf{b})}^{j,m}, \tag{14}$$

where $\beta^j$ is the weight of the sample optimization function of the $j$th frame, $u_{F(\boldsymbol{\alpha}, \mathbf{b})}^{j,m} = \left\| \mathbf{y}_c - b_{m,p} \mathbf{K}_m^j \boldsymbol{\alpha}_p \right\|_2^2 + \lambda b_{m,p} \boldsymbol{\alpha}_p^{\mathrm{T}} \mathbf{K}_m^j \boldsymbol{\alpha}_p, j = 2, \cdots, p, \beta_m^1 = (1 - \gamma_m)^{p-1}, \beta_m^j = \gamma_m (1 - \gamma_m)^{p-j}$, $\boldsymbol{\alpha}_p = (\alpha_{0,p}, \cdots, \alpha_{l-1,p})^T$, $\mathbf{b}_p = (b_{1,p}, \cdots, b_{M,p})^T$, $\sum_{m=1}^{M} b_{m,p} = 1$, $p$ is the number of historical

frames, $\gamma_m \in (0, 1)$ is the learning rate, and $\mathbf{K}_m^j$ is the Gram matrix of kernel $m$. The new optimization problem is

$$\min_{\boldsymbol{\alpha}_p, b_p} F_p(\boldsymbol{\alpha}_p, \mathbf{b}_p)$$
$$\text{s.t. } \sum_{m=1}^{M} d_{m,p} = 1$$
$$d_{m,p} \geq 0, m = 1, \cdots, M. \tag{15}$$

First, given $\mathbf{b}_p$ to solve $\boldsymbol{\alpha}_p$, the above optimization problem becomes an unconstrained optimization problem. Let $\nabla_{\boldsymbol{\alpha}_p} F_p(\boldsymbol{\alpha}_p, \mathbf{b}_p)$, then,

$$\boldsymbol{\alpha}_p = \left( \sum_{j=1}^{p} \sum_{m=1}^{M} \beta_m^j \left( \left( b_{m,p} \mathbf{K}_m^j \right)^2 + \lambda b_{m,p} \mathbf{K}_m^j \right) \right)^{-1} \cdot \sum_{j=1}^{p} \sum_{m=1}^{M} \beta_m^j b_{m,p} \mathbf{K}_m^j \mathbf{y}_c. \tag{16}$$

The efficient evaluation can be achieved through the utilization of FFT.

$$A_p \equiv F(\boldsymbol{\alpha}_p) = \frac{\sum\limits_{j=1}^{p} \sum\limits_{m=1}^{M} \beta_m^j F\left( b_{m,p} \mathbf{k}_m^j \right) \odot F(\mathbf{y}_c)}{\sum\limits_{j=1}^{p} \sum\limits_{m=1}^{M} \beta_m^j F\left( b_{m,p} \mathbf{k}_m^j \right) \odot \left( F\left( b_{m,p} \mathbf{k}_m^j \right) + \lambda \right)}. \tag{17}$$

Set

$$A_p = \frac{A_p^N}{A_p^D} = \frac{\sum\limits_{m=1}^{M} A_{m,p}^N}{\sum\limits_{m=1}^{M} A_{m,p}^D}. \tag{18}$$

when $p = 1$,

$$A_{m,1}^N = F\left( b_{m,1} \mathbf{k}_m^1 \right) \odot F(\mathbf{y}_c)$$
$$A_{m,1}^D = F\left( b_{m,1} \mathbf{k}_m^1 \right) \odot \left( F\left( b_{m,1} \mathbf{k}_m^1 \right) + \lambda \right). \tag{19}$$

when $p > 1$,

$$A_{m,p}^N = (1 - \gamma_m) A_{m,p-1}^N + \gamma_m F\left( b_{m,p} \mathbf{k}_m^p \right) \odot F(\mathbf{y}_c)$$
$$A_{m,p}^D = (1 - \gamma_m) A_{m,p-1}^D + \gamma_m F\left( b_{m,p} \mathbf{k}_m^p \right) \odot \left( F\left( b_{m,p} \mathbf{k}_m^p \right) + \lambda \right). \tag{20}$$

The optimal solution $\alpha_p^*$ can be attained by means of the aforementioned iteration. Subsequently, when presented with the task of solving $\mathbf{b}_p$ given $\boldsymbol{\alpha}_p$, the optimization problem outlined earlier transforms into a constrained optimization problem. To address this issue, we initially posit it as an unconstrained optimization problem and subsequently demonstrate that the resulting solution $\mathbf{b}_p$ conforms to the prescribed constraint conditions. Let $\nabla_{\mathbf{b}_p} F_p(\boldsymbol{\alpha}_p, \mathbf{b}_p)$, then,

$$b_{m,p} = \frac{\sum\limits_{j=1}^{p} \beta_m^j \left( \mathbf{K}_m^j \boldsymbol{\alpha}_p \right)^{\mathrm{T}} (2\mathbf{y}_c - \lambda \boldsymbol{\alpha}_p)}{2 \sum\limits_{j=1}^{p} \beta_m^j \left( \mathbf{K}_m^j \boldsymbol{\alpha}_p \right)^{\mathrm{T}} \left( \mathbf{K}_m^j \boldsymbol{\alpha}_p \right)}, \tag{21}$$

where $m = 1, \cdots, M$, let

$$b_{m,p} = \frac{b_{m,p}^N}{b_{m,p}^D}. \tag{22}$$

when $p = 1$,

$$
\begin{aligned}
b_{m,p}^N &= (1 - \gamma_m)b_{m,p-1}^N + \gamma_m \left(\mathbf{K}_m^p \boldsymbol{\alpha}_p\right)^{\mathrm{T}} (2\mathbf{y}_c - \lambda \boldsymbol{\alpha}_p) \\
b_{m,p}^D &= (1 - \gamma_m)b_{m,p-1}^D + 2\gamma_m \left(\mathbf{K}_m^p \boldsymbol{\alpha}_p\right)^{\mathrm{T}} \left(\mathbf{K}_m^p \boldsymbol{\alpha}_p\right) \quad .
\end{aligned}
\tag{23}
$$

when $p > 1$,

$$
\begin{aligned}
b_{m,1}^N &= \left(\mathbf{K}_m^1 \boldsymbol{\alpha}_1\right)^{\mathrm{T}} (2\mathbf{y}_c - \lambda \boldsymbol{\alpha}_1) \\
b_{m,1}^D &= 2\left(\mathbf{K}_m^1 \boldsymbol{\alpha}_1\right)^{\mathrm{T}} \left(\mathbf{K}_m^1 \boldsymbol{\alpha}_1\right) \quad .
\end{aligned}
\tag{24}
$$

The FFT method enables the rapid calculation of $\mathbf{K}_m^p \boldsymbol{\alpha}_p$ as $F^{-1}\left(F^*\left(\mathbf{k}_m^p\right) \odot F(\alpha_p)\right) = F^{-1}\left(F^*\left(\mathbf{k}_m^p\right) \odot A_p\right)$. Subsequently, the optimal solution can be attained through the aforementioned iteration. Finally, the solution is verified to conform to the prescribed constraint conditions.

The kernel function employed in this algorithm utilizes a Gaussian kernel, and the elements within the kernel matrix can be computed using the following formula,

$$
k_m\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-\frac{1}{\sigma_k^2}\left(\|\mathbf{x}_i - \mathbf{x}_j\|^2\right)\right).
\tag{25}
$$

The exponent within the exponential function $\exp(\cdot)$ used in this study is determined by the negative normalized Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. The higher the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ in Euclidean space, the greater the value of $k_m\left(\mathbf{x}_i, \mathbf{x}_j\right)$. This function is commonly known as the likelihood function in filter-based tracking [37]. It enables the calculation of a value that represents the likelihood of the measured real target given any measurement. Based on the maximum likelihood estimation criterion, the target position is estimated by determining the peak position $y_p$ of the correlation response $\mathbf{y}$.
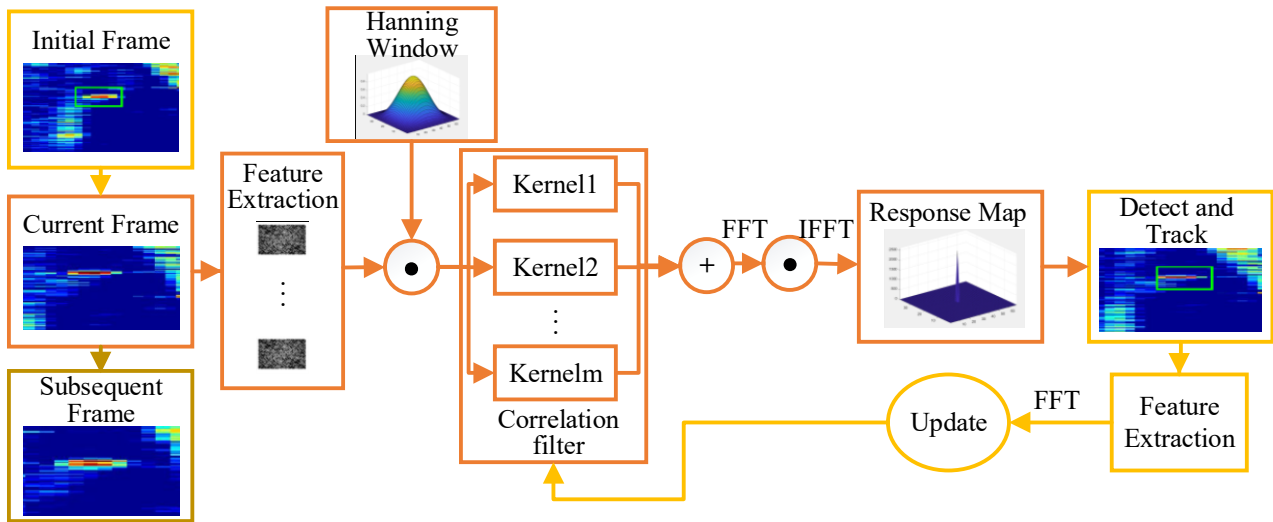


**Figure 2.** The diagram of the MKCF tracking-by-detection algorithm.

*4.2. The Adaptive Reliability Check*

In the correlation filter response map, a single peak is observed, and the sharpness of the peak corresponds to the reliability of the tracking result. The work conducted by Bolme [34] has proposed the idea that the peak-to-sidelobe ratio (PSR) possesses the potential to serve as an indicator of the sharpness of the response peak. The PSR is defined as

$$
S_p = \frac{\max\left(R_p\right) - \mu_p}{\sigma_p},
\tag{26}
$$

where $R_p$ represents the response map calculated by the correlation filter at frame $p$, and $\mu_p$ and $\sigma_p$ are the mean and standard deviation of $R_p$, respectively. In cases where the tracking result is unreliable, as exemplified in Figure 3, the response map may exhibit multiple peaks with low values, resulting in a significant decrease in the PSR. Therefore, the PSR can serve as an indicator of tracking result quality to a certain degree.
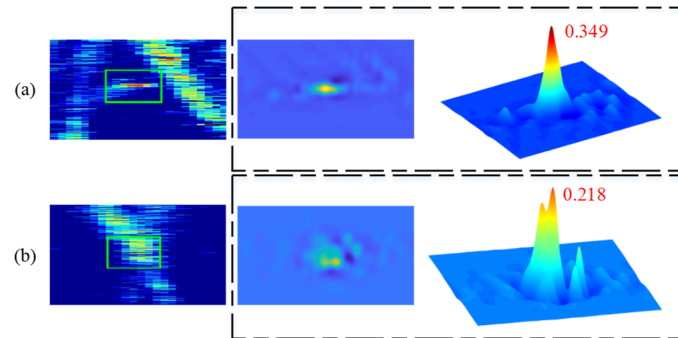


**Figure 3.** From left to right: input image, correlation response map (with a 2D visualization), and correlation response map (with a 3D visualization). (**a**) The target is in the target box. (**b**) The non-target is in the target box.

It is not suitable to pre-define a constant threshold to judge the reliability of the current tracking because the PSR fluctuates between different values due to the uncertainty factor in different scenarios. In order to mitigate the impact of fluctuations in the PSR, we utilize the historical frames to compute the average score to determine the reliability of the tracking results. We combine the PSR values of the historical frames to $C = \{S_2, \cdots, S_{p-1}\}$ with the mean of $M$. Furthermore, we introduce a small coefficient $\tau_1$, whereby the PSR $S_i$ of the $i$th frame is stored in $C$ if $S_i < \tau_1 \cdot M$ and discarded otherwise. Finally, the evaluation criteria of the MKCF algorithm adaptively changes from frame to frame as the average of the multi-frame PSR is computed.

We check the reliability of the tracking result in each frame. The tracking result is deemed unreliable if $S_i < \tau_1 \cdot M$ ("Unreliability Check" in Figure 1). On the other hand, the tracking result is likely to be reliable if it satisfies $S_i > \tau_2 \cdot M$ ("Reliability Check" in Figure 1) and the coefficient $\tau_2$ is higher than $\tau_1$. Once the initial tracking result is determined to be unreliable, the re-detection module is initiated.

### 4.3. The Re-Detection Module

This section provides an introduction to the re-detection module, which plays a crucial role in generating candidate target locations and determining whether to substitute the initial tracking result with the optimal candidate target location. A key component of this module is the implementation of the Kalman filter, which utilizes reliable tracking results from the current frame for filter updates. Assuming that the motion between adjacent frames adheres to a linear Gaussian distribution as prior knowledge, the motion model in the Kalman filter can be established as a uniform linear motion model.

In cases where the track result is deemed unreliable, the target positions from the last reliable tracking results are saved and used to drive a Kalman filter, providing an estimated position $position_p$ for the current target position with the variance in $\psi\left(position_p\right)$. Subsequently, $N$ random locations $g_j (j = 1, \cdots, N)$ are generated around the location $position_p$, following a Gaussian distribution with the mean $position_p$ and the variance $\psi\left(position_p\right)$. With $g_j$ as the center, $N$ target candidate bounding boxes $B_j$ are generated. For the obtained $B_j$, their response maps $R_p^j$ are generated by Equation (6) and the maximum values of these maps, $q_j = \max_u R_p^j(u)$, are also computed. Suppose $q_j^*$ is the maximum among $\{q_1, \ldots, q_N\}$. The best candidate bounding box of the target $B_j^*$ and the best candidate

location of the target $g_j^*$ are determined accordingly. Finally, the decision of whether the best candidate location $g_j^*$ replaces the initial tracking result is determined through the following two steps.

(1) If it does not meet the reliability check $S_i > \tau_2 \cdot M$, the initial tracking result is used;

(2) If it meets the reliability check $S_i > \tau_2 \cdot M$, we compute the correlation response at the initial tracking location, and the highest response value is recorded as $k_p$. If (27) is met, the initial tracking result is replaced with $g_j^*$.

$$q_j^* \geq \gamma \times k_p, \tag{27}$$

where $\gamma$ is the penalty parameter, and the * indicates the optimal. If the above equation is not satisfied, the initial tracking result is not replaced.

## 5. Experimental Results

In this section, we apply the proposed method on two test scenarios and compare it with the traditional tracking methods and the original KCF algorithms. All the algorithms are implemented in MATLAB 2018b, utilizing an Intel i5-6200U CPU with a main frequency of 2.3 GHz and 8 GB of memory. We make a Table 1 to summarize the abbreviations of various algorithms.

**Table 1.** The abbreviations of various algorithms.

| Algorithm | Abbreviation |
|---|---|
| Multiple Hypothesis Tracking | MHT |
| Joint Probabilistic Data Association | JPDA |
| Probability Hypothesis Density | PHD |
| Multi-Feature Kernel Correlation Filter | MF-KCF |
| Multi-Kernel Correlation Filter | MKCF |
| Improved Multi-Feature Kernel Correlation Filter | IMF-KCF |
| Improved Multi-Kernel Correlation Filter | IMKCF |

### 5.1. Evaluation Metrics

Evaluation metrics of performance are discussed below.

(1) Root-mean-square error (RMSE) and precision: the average distance error between the estimated position and the actual position, defined as (28). Given a threshold $M$, the centroid position is properly estimated if its RMSE is less than $M$. The RMSE accuracy is defined as the percentage of the total number of frames for which the location is correctly estimated.

$$\text{RMSE} = \sqrt{\sum_{i=1}^{m} \left[ \left( \hat{x}_k^i - x_k \right)^2 + \left( \hat{y}_k^i - y_k \right)^2 \right] / m}, \tag{28}$$

where $m$ is the number of Monte Carlo experiments.

(2) Intersection over union (IOU) and precision: IOU is defined as (29). A larger IOU value indicates a more accurate estimation of the target. Given a threshold $N$, the target box is considered correctly estimated if its IOU is greater than $N$. The IOU accuracy is defined as the percentage of the total number of frames for which the target box is correctly estimated.

$$\text{IOU} = \frac{E \cap G}{E \cup G}, \tag{29}$$

where $E$ denotes the area of the estimated target and $G$ denotes the area of the real target. Operator $\cap$ represents intersection and $\cup$ means the union.

(3) Frames Per Second (FPS): The number of frames processed per second, the greater the FPS; the higher the efficiency of the algorithm.

*5.2. Test Scenarios and Parameter Settings*

5.2.1. Test Scenarios

In order to verify the effect of the proposed algorithm on tracking targets exhibiting different types of motion, we have designed two scenarios. Notably, both the ball and the diver have equivalent target strength, with the difference lying in their respective motion types. The experiment used a GPS device to record the actual motion trajectory of the target. The GPS device is soft-connected to the target and floats on the ocean directly above the target.

(a) Maneuvering target: As shown in Figure 4a, the small boat drags the ball to perform a turning motion. Figure 5a is the sound speed profile of the experiment. The actual trajectory of the target is plotted in Figure 5b, represented by the red line. Subsequent to the 70th frame, the SRR is lower than 0 dB for the majority of frames within this specific scene.
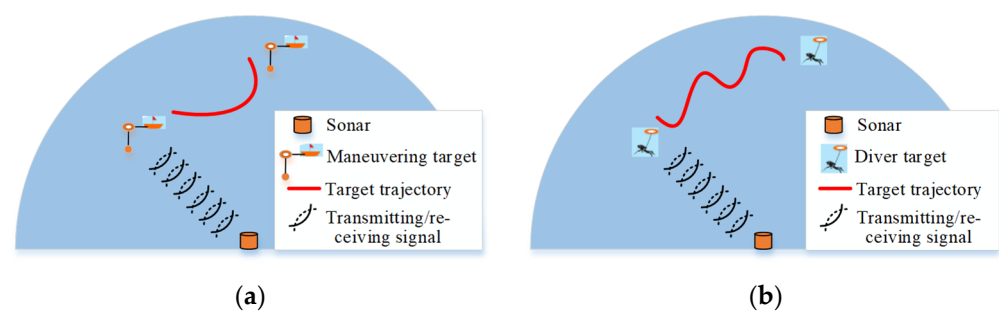


**Figure 4.** The test scenarios. (**a**) Maneuvering target. (**b**) Diver target.
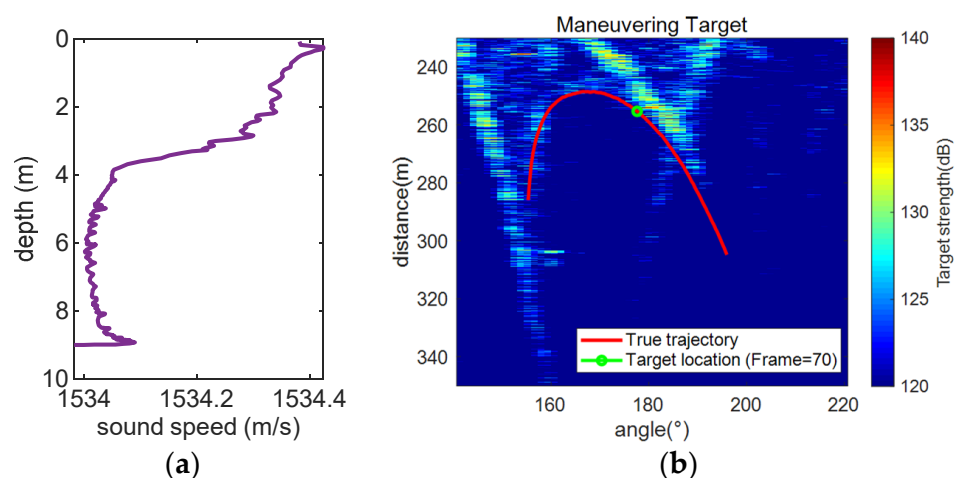


**Figure 5.** (**a**) Sound speed profile. (**b**) The actual trajectory of maneuvering target in acoustic image.

(b) Diver target: As represented in Figure 4b, the closed diver moves in a Z-shaped manner. The sound speed profile is shown in Figure 6a. The actual trajectory of the target is plotted in Figure 6b, represented by the red line. In this scene, the SRR surpasses 0 dB for the majority of frames, indicating a higher quality data set.
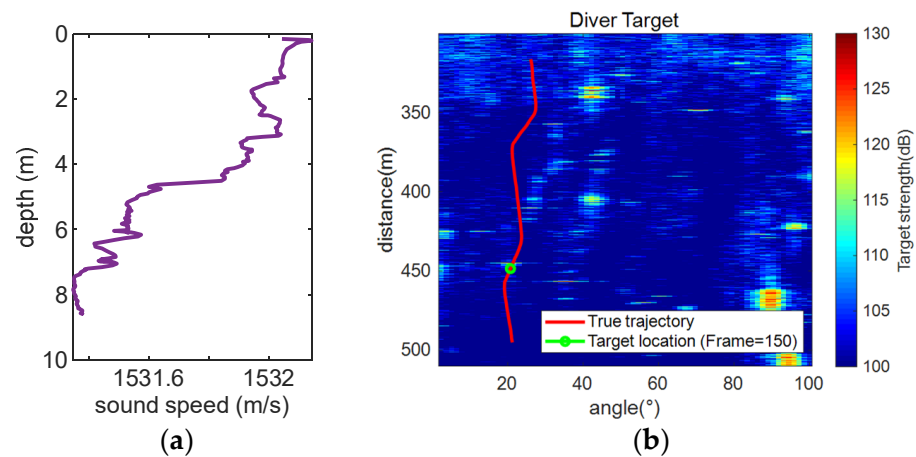
**Figure 6.** (**a**) Sound speed profile. (**b**) The actual trajectory of diver target in acoustic image.

5.2.2. Parameter Settings

In this paper, we use the histogram of oriented gradient (HOG) and invariant moment features. The HOG feature has nine gradient orientations, and the cell size is $4 \times 4$. In the KCF, it uses a single Gaussian kernel with parameter $\sigma = 0.5$ and the learning factor $\eta = 0.01$. It is worth noting that the multi-feature KCF (MF-KCF) trains the tracker based on the above features' fusion. The MKCF uses two Gaussian kernels with parameters $\sigma_1 = 0.3$ and $\sigma_2 = 0.3$ and learning factors $\eta = 0.0175$ and $\eta = 0.018$, respectively. Both methods employ a regularization parameter of $\lambda = 10^{-4}$ [38]. In the adaptive reliability check module, the coefficients $\tau_1$ and $\tau_2$ are 0.75 and 0.9, respectively.

To address the issue of high-frequency noise caused by abrupt edges in samples after the cyclic shift, this study uses the Hanning window when processing sample features. The application of the Hanning window aids in smoothing boundaries and minimizing interference from background information, consequently enhancing overall tracking accuracy. Furthermore, due to the potential decline in detection and tracking performance associated with an excessive search area in the KCF, it becomes necessary to restrict the size of the search area. Hence, the search area is limited to 2.5 times the size of the target box [39].

### 5.3. Data Processing and Analysis

5.3.1. Comparison with Traditional Tracking Algorithms

In this section, a comparison is made between the proposed algorithm and three commonly used algorithms in underwater target detection and tracking: MHT, JPDA, and PHD. To minimize target loss during thresholding, a relatively low threshold is adopted in the data preprocessing stage. It should be noted that this approach can lead to an increased probability of false alarms, resulting in an increased number of false targets, thereby adding difficulties of data association and computational costs.

The outcomes of data processing for maneuvering and diver targets are depicted in Figures 7 and 8. Analysis of these figures reveals that the proposed IMKCF algorithm consistently maintains a relatively low RMSE across the majority of frames in both maneuvering and diver target tracking processes, when compared to the other three algorithms. This superior performance can mainly be attributed to two factors. Firstly, our response map, which is based on target features, effectively identifies target areas. Secondly, the re-detection module with a Kalman filter estimates a reliable target location when the tracking result is deemed unreliable.
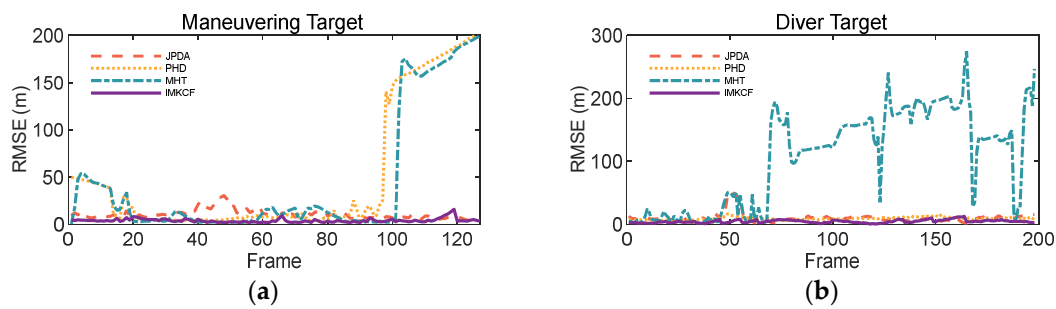
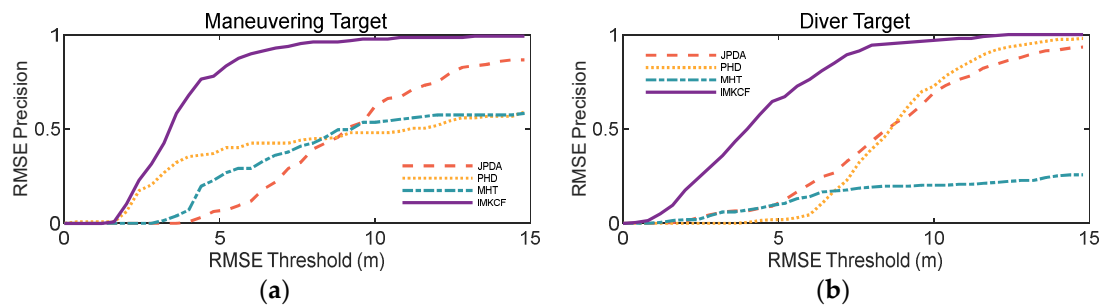**Figure 7.** (**a**) RMSE with maneuvering target. (**b**) RMSE with diver target.



**Figure 8.** (**a**) RMSE precision with maneuvering target. (**b**) RMSE precision with diver target.

As shown in Figure 7, in the tracking of maneuvering targets, the three traditional tracking algorithms perform poorly, particularly the PHD and MHT algorithms. Both exhibit significant tracking drift at approximately 100 frames. The JPDA and PHD algorithms demonstrate satisfactory performance in tracking the diver target, while the MHT algorithm experiences tracking drift at about 70 frames. The JPDA algorithm excels in cluttered environments as it does not require prior information about the target and clutter, allowing for successful target tracking. Nevertheless, because this study focuses on a single-target tracking scenario, the JPDA algorithm performs relatively well. In contrast, the MHT algorithm necessitates prior information about the target and clutter, and the computational complexity increases exponentially with the clutter density. To enhance computational efficiency, we set a smaller value for N-scan pruning, but this compromises the tracking performance of the algorithm. The PHD algorithm, which is based on RFS theory, avoids the intricate correlation process associated with traditional methods and exhibits high computational efficiency. In the data processing of the two scenarios, we assume a uniform linear motion model for both the MHT and PHD algorithms. However, this motion constraint is not robust in low SRR environments, leading to the failure of the PHD algorithm in tracking maneuvering targets. Our proposed method surpasses these algorithms by utilizing multiple features and incorporating reliability estimation to identify a reliable re-detected target for self-correction.

Figure 8a,b display the percentage of frames within a given RMSE threshold for the error between the estimated position and the true position of the maneuvering target and the diver target, respectively. Analysis of these figures demonstrates that the IMKCF algorithm outperforms the other three tested algorithms in terms of RMSE precision. Specifically, when the RMSE threshold is set at 5, the IMKCF algorithm achieves an accuracy rate close to 80%, while the other algorithms only have approximately 50% accuracy when the RMSE threshold is 10.

### 5.3.2. Comparison with Original KCF Algorithms

In this section, a comparative analysis is conducted between the proposed algorithm and three original KCF algorithms, namely, MF-KCF, improved MF-KCF (IMF-KCF), and MKCF. Figure 9 illustrates the target position in the final frame and tracking outcomes

for all four algorithms. It can be observed from the figure that both the MF-KCF and MKCF algorithms fail to track the maneuvering target. Conversely, the IMF-KCF and IMKCF algorithms, incorporating adaptive reliability checks and a re-detection module, successfully track the target. At approximately the 70th frame, target tracking is interfered with by reverberation, and the training samples are contaminated, resulting in error propagation during model training and subsequent tracker drift. The IMKCF algorithm checks real-time reliability on target tracking results and re-detects the target position when deemed unreliable, preventing frame drift. The versatility of the re-detection module is demonstrated by its successful implementation in both algorithms.
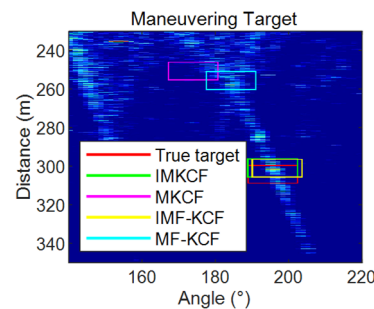


**Figure 9.** Track results (frame = 127).

To further elucidate tracking performance under low-SRR scenarios, Figure 10 presents the frame-by-frame RMSE and IOU of maneuvering target tracking. Higher IOU values and lower RMSE values signify more accurate tracking outcomes. As shown in Figure 10, both the MF-KCF and MKCF algorithms lose track of the targets around the 70th frame. The IMKCF algorithm outperforms the IMF-KCF algorithm, exhibiting relatively low RMSE and high IOU values across most frames, indicating superior accuracy. This improvement can be attributed to the MKCF tracking-by-detection module, which enables the algorithm to make full use of the complementary features and improve tracking accuracy.
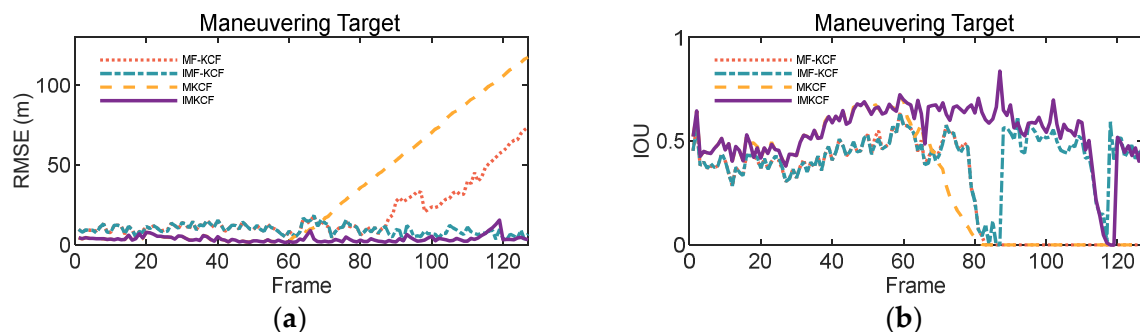


**Figure 10.** (**a**) RMSE. (**b**) IOU.

Figure 11a depicts the RMSE precision. The figure reveals that when the RMSE threshold is set at 10, the IMKCF algorithm achieves nearly 100% precision, while the other three algorithms fall below 60%. Additionally, Figure 11b demonstrates the IOU precision. Notably, when the IOU threshold is set at 0.5, the IMKCF algorithm attains nearly 100% precision, whereas the other three algorithms exhibit less than 60%. The results show that, compared with the single-kernel correlation filter, the correlation filter based on multiple kernels has greater advantages in tracking accuracy.
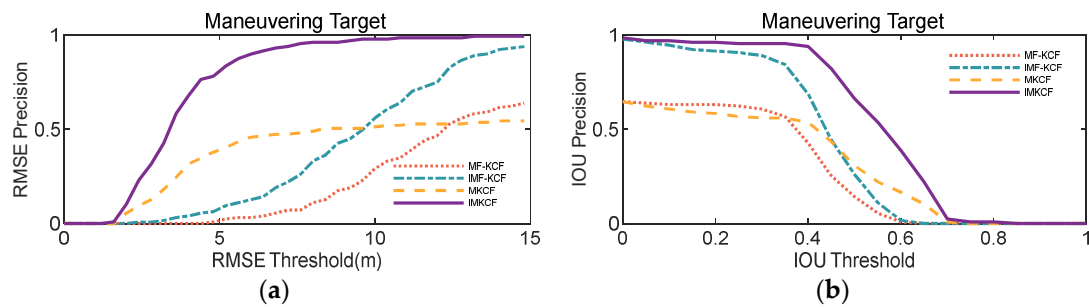
**Figure 11.** (**a**) RMSE precision. (**b**) IOU precision.

Figure 12 illustrates the PSR curve during the tracking process of the MKCF and IMKCF algorithms. It can be observed that the PSR of the MKCF algorithm experiences a significant decline around the 70th frame, whereas the PSR of the IMKCF algorithm fluctuates steadily throughout the tracking process. These findings indicate that the PSR score serves as an indicator of tracking result reliability, and the adaptive reliability check and re-detection modules within the IMKCF algorithm play a vital role in enhancing tracking robustness.
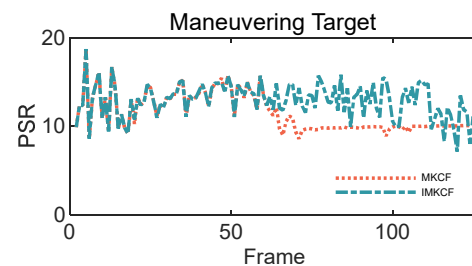


**Figure 12.** The comparison of PSR between MKCF and IMKCF algorithms.

### 5.3.3. Algorithm Efficiency

Table 2 displays the PFS and average RMSE of the above algorithms. The results indicate that the PHD algorithm has the highest FPS and computational efficiency but exhibits the poorest tracking accuracy. In contrast, the KCF algorithms exhibit relatively lower computational efficiency but has higher tracking accuracy compared to the classical filtering algorithm. The analysis suggests that MKCF exhibits a slight improvement in computational efficiency when compared MF-KCF. This implies that the incorporation of an additional kernel in kernel correlation filtering does not result in a significant increment in computational cost. Moreover, the results demonstrate that IMKCF achieves superior tracking accuracy in comparison to both IMF-KCF and MF-KCF. Notably, Table 2 reveals that the inclusion of a re-detection module can lead to an increase in computational cost, as it carries out target re-detection when the tracking results are deemed unreliable.

**Table 2.** FPS and the average of RMSE.

| | MHT | JPDA | PHD | MF-KCF | MKCF | IMF-KCF | IMKCF |
|---|---|---|---|---|---|---|---|
| FPS | 7.2 | 107.1 | 122.3 | 22.3 | 27.2 | 11.3 | 14.2 |
| RMSEaver | 44.82 | 10.26 | 50.83 | 19.94 | 33.52 | 9.6 | 3.86 |

## 6. Conclusions

In this paper, we propose an IMKCF algorithm to solve the challenging problem of detecting and tracking weak targets with varying movements in a complex marine environment. The IMKCF algorithm consists of three modules: the MKCF tracking-by-detection, the adaptive reliability check, and the re-detection modules. The MKCF tracking-

by-detection module employs a multi-frame data weighted average technique to adaptively update the coefficients of multiple kernels, thereby enhancing tracking accuracy. We conduct a comprehensive analysis of the MKCF algorithm using a maximum likelihood perspective and prove that the target location can be precisely determined based on the location of the maximum value of the correlation response. The remaining two modules work collaboratively to improve the robustness of target tracking. In particular, the previous reliable tracking results are utilized to drive a Kalman filter, generating a position estimate when the tracking result is considered unreliable. A decision is then made about whether to replace the original target position with the estimated one.

In data processing, we extracted HOG features and invariant moment features to train the proposed IMKCF algorithm, which has been compared with traditional tracking algorithms and original KCF algorithms. The experimental results demonstrate that our proposed algorithm not only exhibits the capability of effectively tracking underwater targets with diverse motion types but also achieves long-term robust tracking in low-SRR environments. Moreover, the tracking accuracy of our algorithm surpasses that of the single-core correlation filter. Currently, the method proposed in this paper is only suitable for single-target tracking. In future research, we plan to delve deeper into the challenges of multiple weak underwater target tracking. Additionally, we aim to mine more target feature information to enhance the algorithm's robustness.

**Author Contributions:** W.Y. conceived the main idea, designed the main algorithm, and wrote the manuscript. The experimental results were analyzed by W.Y., F.X. and J.Y. provided suggestions for the proposed algorithm. All authors have read and agreed to the published version of the manuscript.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The data are not publicly available due to our laboratory policy.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Jeong, T.T. Particle PHD Filter Multiple Target Tracking in Sonar Image. *IEEE Trans. Aerosp. Electron. Syst.* **2007**, *43*, 409–416. [CrossRef]
2. Rodningsby, A.; Bar-Shalom, Y. Tracking of Divers using a Probabilistic Data Association Filter with a Bubble Model. *IEEE Trans. Aerosp. Electron. Syst.* **2009**, *45*, 1181–1193. [CrossRef]
3. Zhang, H.; Tian, M.; Ouyang, Q.; Liu, J.; Shao, G.; Cheng, J. Track Detection of Underwater Moving Targets Based on CFAR. *J. Phys. Conf. Ser.* **2023**, *2486*, 012076. [CrossRef]
4. Zhu, J.; Song, Y.; Jiang, N.; Xie, Z.; Fan, C.; Huang, X. Enhanced Doppler Resolution and Sidelobe Suppression Performance for Golay Complementary Waveforms. *Remote Sens.* **2023**, *15*, 2452. [CrossRef]
5. Zhang, D.; Gao, L.; Sun, D.; Teng, T. Soft-decision Detection of Weak Tonals for Passive Sonar using Track-before-detect Method. *Appl. Acoust.* **2022**, *188*, 108549. [CrossRef]
6. Yi, W.; Fu, L.; García-Fernández, Á.F.; Xu, L.; Kong, L. Particle Filtering based Track-before-detect Method for Passive Array Sonar Systems. *Signal Process.* **2019**, *165*, 303–314. [CrossRef]
7. Vivone, G.; Braca, P. Joint Probabilistic Data Association Tracker for Extended Target Tracking Applied to X-Band Marine Radar Data. *IEEE J. Ocean. Eng.* **2016**, *41*, 1007–1019. [CrossRef]
8. Yang, S.; Thormann, K.; Baum, M. Linear-Time Joint Probabilistic Data Association for Multiple Extended Object Tracking. In Proceedings of the 2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM), Sheffield, UK, 8–11 July 2018; pp. 6–10.
9. Blackman, S.S. Multiple Hypothesis Tracking for Multiple Target Tracking. *IEEE Aerosp. Electron. Syst. Mag.* **2004**, *19*, 5–18. [CrossRef]
10. Li, X.; Zhao, C.; Lu, X.; Wei, W. Underwater Bearings-Only Multitarget Tracking Based on Modified PMHT in Dense-Cluttered Environment. *IEEE Access* **2019**, *7*, 93678–93689. [CrossRef]
11. Zhou, T.; Wang, Y.; Chen, B.; Zhu, J.; Yu, X. Underwater Multitarget Tracking with Sonar Images Using Thresholded Sequential Monte Carlo Probability Hypothesis Density Algorithm. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1506305. [CrossRef]
12. Williams, J.L. Marginal Multi-bernoulli Filters: RFS Derivation of MHT, JIPDA, and Association-based Member. *IEEE Trans. Aerosp. Electron. Syst.* **2015**, *51*, 1664–1687. [CrossRef]
13. Chai, L.; Kong, L.; Li, S.; Yi, W. The Multiple Model Multi-Bernoulli Filter based Track-before-detect using a Likelihood based Adaptive Birth Distribution. *Signal Process.* **2020**, *171*, 107501. [CrossRef]

14. Li, S.; Yi, W.; Kong, L.; Wang, B. Multi-bernoulli Filter based Track-before-detect for Jump Markov Models. In Proceedings of the 2014 IEEE Radar Conference, Cincinnati, OH, USA, 19–23 May 2014; pp. 1257–1261.
15. Liu, Z.-X.; Zhang, Q.-Q.; Li, L.-Q.; Xie, W.-X. Tracking Multiple Maneuvering Targets using a Sequential Multiple Target Bayes Filter with Jump Markov System Models. *Neurocomputing* **2016**, *216*, 183–191. [CrossRef]
16. Yue, W.; Xu, F.; Xiao, X.; Yang, J. Track-before-Detect Algorithm for Underwater Diver Based on Knowledge-Aided Particle Filter. *Sensors* **2022**, *22*, 9649. [CrossRef]
17. Nam, H.; Han, B. Learning multi-domain convolutional neural networks for visual tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4293–4302.
18. Fan, H.; Ling, H. SANet: Structure-aware network for visual tracking. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 2217–2224.
19. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. Fully-convolutional Siamese networks for object tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–10 October 2016; pp. 850–865.
20. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. Exploiting the circulant structure of tracking-by-detection with kernels. In Proceedings of the 12th European Conference on Computer Vision (ECCV), Florence, Italy, 7–13 October 2012; pp. 702–715.
21. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-speed Ttracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. [CrossRef] [PubMed]
22. Shin, J.; Kim, H.; Kim, D.; Paik, J. Fast and Robust Object Tracking Using Tracking Failure Detection in Kernelized Correlation Filter. *Appl. Sci.* **2020**, *10*, 713. [CrossRef]
23. Zhang, L.; Suganthan, P.N. Robust Visual Tracking via Co-trained Kernelized Correlation Filters. *Pattern Recognit.* **2017**, *69*, 82–93. [CrossRef]
24. Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Torr, P.H.S. Staple: Complementary Learners for Real-Time Tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1401–1409.
25. Hao, Z.; Liu, G.; Gao, J.; Zhang, H. Robust Visual Tracking Using Structural Patch Response Map Fusion Based on Complementary Correlation Filter and Color Histogram. *Sensors* **2019**, *19*, 4178. [CrossRef]
26. Sun, X.; Cheung, N.-M.; Yao, H.; Guo, Y. Non-rigid Object Tracking via Deformable Patches using Shape-Preserved KCF and Level Sets. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 5496–5504.
27. Zhou, Y.; Su, H.; Tian, S.; Liu, X.; Suo, J. Multiple Kernelized Correlation Filters based Track-Before-Detect Algorithm for Tracking Weak and Extended Target in Marine Radar Systems. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 3411–3426. [CrossRef]
28. Zhou, Y.; Wang, T.; Hu, R.; Su, H.; Liu, Y.; Liu, X.; Suo, J.; Snoussi, H. Multiple Kernelized Correlation Filters (MKCF) for Extended Object Tracking Using X-Band Marine Radar Data. *IEEE Trans. Signal Process.* **2019**, *67*, 3676–3688. [CrossRef]
29. Zeng, X.; Xu, L.; Cen, Y.; Zhao, R.; Hu, S.; Xiao, G. Visual Tracking Based on Multi-Feature and Fast Scale Adaptive Kernelized Correlation Filter. *IEEE Access* **2019**, *7*, 83209–83228. [CrossRef]
30. Ren, H.; Qiao, J.; Shi, T. Multifeature Fusion Tracking Algorithm Based on Self-Associative Memory Learning Mechanism. *IEEE Access* **2022**, *10*, 100605–100614. [CrossRef]
31. Tang, M.; Feng, J. Multi-kernel Correlation Filter for Visual Tracking. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 3038–3046.
32. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual Object Tracking using Adaptive Correlation Filters. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.
33. Wang, N.; Zhou, W.; Li, H. Reliable Re-Detection for Long-Term Tracking. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *29*, 730–743. [CrossRef]
34. Ma, C.; Yang, X.; Zhang, C.; Yang, M.-H. Long-term Correlation Tracking. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5388–5396.
35. Tang, F.; Ling, Q. Contour-Aware Long-Term Tracking with Reliable Re-Detection. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *30*, 4739–4754. [CrossRef]
36. Varma, M.; Ray, D. Learning the Discriminative Power-Invariance Trade-Off. In Proceedings of the 2007 IEEE 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 14–21 October 2007; pp. 1–8.
37. Esmzad, R.; Mahboobi Esfanjani, R. Modified likelihood probabilistic data association filter for tracking systems with delayed and lost measurements. *Digit. Signal Process.* **2018**, *76*, 66–74. [CrossRef]
38. Tang, M.; Yu, B.; Zhang, F.; Wang, J. High-Speed Tracking with Multi-kernel Correlation Filters. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, CA, USA, 18–23 June 2018; pp. 4874–4883.
39. Danelljan, M.; Khan, F.S.; Felsberg, M.; van de Weijer, J. Adaptive Color Attributes for Real-Time Visual Tracking. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 1090–1097.