remote sensing

MDPI

Article

A Robust Underwater Multiclass Fish-School Tracking Algorithm

Tao Liu [1], Shuangyan He [1,2], Haoyang Liu [1], Yanzhen Gu [1,2,*] and Peiliang Li [1,2]

[1] Ocean College, Zhejiang University, Zhoushan 316021, China
[2] Hainan Institute, Zhejiang University, Sanya 572025, China
* Correspondence: guyanzhen@zju.edu.cn

Abstract: State-of-the-art multiple-object tracking methods are frequently applied to people or vehicle tracking, but rarely involve underwater-object tracking. Compared with the processing in non-underwater photos or videos, underwater fish tracking is challenging due to variations in light conditions, water turbidity levels, shape deformations, and the similar appearances of fish. This article proposes a robust underwater fish-school tracking algorithm (FSTA). The FSTA is based on the tracking-by-detection paradigm. To solve the problem of low recognition accuracy in an underwater environment, we add an amendment detection module that uses prior knowledge to modify the detection result. Second, we introduce an underwater data association algorithm for aquatic non-rigid organisms that recombines representation and location information to refine the data matching process and improve the tracking results. The Resnet50-IBN network is used as a re-identification network to track fish. We introduce a triplet loss function based on a centroid to train the feature extraction network. The multiple-object tracking accuracy (MOTA) of the FSTA is 79.1% on the underwater dataset, which shows that it can achieve state-of-the-art performance in a complex real-world marine environment.

Keywords: underwater fish school tracking; annotated underwater fish school dataset; high-performance tracking and detection algorithm; deep learning

Citation: Liu, T.; He, S.; Liu, H.; Gu, Y.; Li, P. A Robust Underwater Multiclass Fish-School Tracking Algorithm. *Remote Sens.* **2022**, *14*, 4106. https://doi.org/10.3390/rs14164106

Academic Editors: Pawel Rotter, Wojciech Chmiel and Sławomir Mikrut

Received: 8 July 2022
Accepted: 19 August 2022
Published: 21 August 2022

1. Introduction

According to the State of World Fisheries and Aquaculture report [1], aquatic product output is expected to increase from 179 million tons in 2018 to 204 million tons in 2030. With the increase in population, the demand for seafood and damage to the ocean are increasing, which leads to the continuous decline of marine fishery resources. An effective way to alleviate this problem is to raise the proportion of marine pastures in the supply of aquatic products [2], which can ensure the steady and sustainable growth of aquatic resources. Therefore, it is necessary to manage marine pastures scientifically and effectively. In recent years, the rapid development of submarine cable online observation systems has provided a way to systematically manage marine pastures. The quantity and behavior of marine organisms can be monitored and tracked using cameras and sensors in an observation system, promoting scientific fishery management and sustainable fish production [3–5]. For example, we can obtain species diversity and richness through video detection and tracking, which can be applied to disease identification, hypoxia stress identification, etc. In addition, we can analyze the coupling relationship between environmental factors and marine organism populations [6].

Fish detection and behavior analyses have been conducted in ocean observation, aquaculture, and biological research. Compared with the traditional method, fish-tracking methods based on computer vision are real-time and automatic, and the expected behaviors of fish are unaffected [7]. However, underwater observation equipment is usually expensive, and difficult to deploy and maintain. Moreover, it is challenging to continuously access underwater videos in real time in the ocean. Research on underwater fish school detection and tracking is primarily performed in following three ways.

(1) Most studies are conducted using laboratory video observations [8–12], and tanks are used to breed fish. They artificially change environmental variables to study underwater fish tracking algorithms and behaviors. Observing and researching a fish school in a flume is convenient. Still, it cannot simulate a complex natural marine or lake environment, such as with underwater color distortion and wave slamming. So, it is challenging to transfer laboratory algorithms to a natural underwater environment.

(2) Some studies use online open datasets, such as ImageNet [13] and Fish4-Knowledge [14]. These collections consist of discontinuous images extracted from different underwater videos, and limited fish species, behaviors, and environmental water conditions are included. Thus, these online open datasets are usually insufficient for a specific underwater fish-school detection and tracking study.

(3) Some studies use the underwater videos of cameras equipped on an underwater observation platform, such as cabled seafloor observatory systems [15]. Underwater observation videos can be continuously transmitted to a land observation center in real time by using a submarine cable. Many data are available, and online detection and tracking algorithms can be applied to them. The videos used in this study are from an underwater observation platform.

A fish resource statistical technology usually consists of two key steps: first, the accurate underwater detection of fish, and then associating the detected fish in every frame with the tracklets. Current research on technical solutions can be generally grouped into three categories [16].

The first approach uses traditional computer graphics algorithms, such as a background subtraction algorithm [17] or a Gaussian mixture model [18]. Each fish in the frames is matched in the video to obtain the trajectories. Matching methods are usually prediction models, such as the Kalman filter [19] and particle filter [20]. Recently, FlowTrack [21], which is used for small-scale tracking, introduced an optical flow method by calculating the motion vectors of each pixel between frames that can apply to fish targets [22]. However, it is complex and time-consuming, and its tracking performance relies heavily on the detection results. It showed poor performance in a turbid water environment due to noise such as water plants and rocks.

The second approach is a one-stage deep learning scheme that combines detection and tracking models into a unified framework, and can avoid the overall loss caused by submodel errors. The joint detection and embedding (JDE) model [23] is the first algorithm based on this idea; later on, its successors, FairMOT [24] and RMCF [25], were also developed. However, these algorithms treat object identity as a classification task. In particular, all object instances of the same identity in a training set are treated as one class. The number of objects grows over time; thus, an increasing number of categories need to be classified, which deteriorates the algorithm's performance. Moreover, it is difficult to associate the trajectories of those objects re-entering the camera view.

The third approach is the two-stage deep learning algorithm following the tracking-by-detection paradigm. It generally detects objects by first using a neural network and then associating objects using a filtering method or a reidentification algorithm. Detection algorithms include YOLOx [26], Faster-RCNN [27], and DETR [28], while filtering algorithms are similar to those of the first approach [19,20]. The reidentification algorithm uses a neural network to extract the representation features of objects to further calculate the similarity between objects. The technology based on deep learning can deal with massive data, and is widely applied in detecting and tracking scenes. It was also used in underwater fish detection and tracking, and studies showed that detection accuracy was significantly improved [29–31]. This study also uses this approach.

We propose a robust underwater fish-school tracking algorithm (FSTA) in this study. The main contributions and innovations of the FSTA are as follows:

1. We propose an amendment detection module. Prior tracking knowledge is introduced to amend the detection model. The image quality of underwater videos suffers from color distortions, deformations, low resolution, and contrast, and the fish features

vary over time. It also leads to inconsistent detection results because of the sudden low confidence scores during detection. Therefore, in this study, tracklets are used as prior knowledge to amend the detection results to improve the performance of the underwater detection model.

2. We propose a new data association algorithm scheme by recombining representation and location information. In this algorithm, the weight parameters of the position and apparent information are dynamic. The loss value can be adjusted according to the lost situation to improve the matching performance. We recovered objects from the low-score detection box while filtering out the background.
3. The centroid triplet loss function was introduced to train the feature extraction network of Restnet50-IBN. This significantly improved the performance of the tracking algorithm. Its MOTA is 86.7, which achieved SOTA performance. We also released the labeled underwater datasets, including detection and tracking datasets.

The structure of this article is as follows. Section 2 reviews the existing methods of fish detection and tracking. The method proposed in this paper is described in Section 3. Section 4 reports and discusses our experimental results. The conclusions and discussion are given in Section 5.

2. Related Work

Several tracking methods have been applied to underwater fish-school tracking in the past decade [30–36]. Chung et al. [32] proposed an automatic fish segmentation algorithm. They first used the object segmentation algorithm to obtain a fish mask, and then combined four cues, namely, vicinity, area, motion direction, and histogram distance, to match the object. In addition, the algorithm modified the Viterbi data association algorithm from a single-target tracking to a multiple-target tracking algorithm. It could effectively divide the fish boundary and overcome poor motion continuity in LFR scenarios. However, this algorithm is time-consuming, and the mismatching phenomenon is serious when an occlusion exists. Palconit et al. [33] used a time-series prediction algorithm to predict the trajectory. They first used the subtraction of two images to obtain the binary image, and then the long short-erm memory (LSTM) or genetic algorithm to predict the fish's position in the next frame. However, most of the images processed by this algorithm were single targets, and the features of the fish were visible. It lacked experimental verification in a complex environment. Liu et al. [25] performed one-stage detection and produced a tracking algorithm. The dataset was collected from a natural marine pasture. They built a parallel two-branch structure in which the detection branch output fish species and position coordinates, and the tracking branch output the number statistics. The algorithm improved the running time and could be deployed in a marine pasture. However, the approach treats object identity embedding as a classification task. As tracking time increases, the number of objects increases, and the algorithm's performance gradually deteriorates, so it is unsuitable for long-term tracking. Sun et al. [34] proposed consistent fish tracking with multiple underwater cameras, introducing a target-background confidence map to build appearance, and using maximal posterior estimation to obtain fish locations. Lastly, centroid coordinate homographic mapping and the speeded-up robust features (SURF) technique [35] were used to capture and match the same fish from the perspectives of multiple cameras with a set of strategies. However, this algorithm could track the same fish from different cameras and cannot process multiple targets. Xu et al. [9] analyzed fish behavior trajectories on the basis of deep learning in an ammonia environment. That research improved the faster R-CNN algorithm [27] to identify fish in a tank, and mapped the behavior trajectory of the fish. After that, they constantly changed the ammonia concentration in the tank and observed behavioral changes in the fish, which would become inactive and gradually die with the increase in ammonia concentration. This approach was used in the laboratory environment, and it is not easily applied in a complex natural marine environment. Wang et al. [15] used the determinant of the Hessian blob detector to extract the head region of each fish, which was the input of the CNN network. So, it obtained the position coordinates from the

CNN network. Lastly, an SVN classifier was used to relink the trajectory across the frames. However, when an occlusion occurs, the fish are easily lost.

In brief, the early tracking algorithm was given priority to computer graphics algorithms, such as the filter algorithm, the optical flow method, and the background modeling algorithm. With the substantial rise of deep learning in recent years, researchers have gradually used computer-vision algorithms to solve tracking problems, and powerful data-fitting capabilities to accelerate computational speed. Therefore, we focus on the accuracy of tracking. Balancing speed and accuracy by improving accuracy without reducing speed is the core idea of the FSTA algorithm proposed in this study.

3. Method

The FSTA algorithm consists of an object detection module, an amendment detection model (ADM), and a data association module. The structure is shown in Figure 1. The detector, such as the YoloX algorithm [26], is used in the object detection module to obtain the objects' classes, bounding boxes, and confidence scores from each frame. Then, the amendment detection model is used to improve underwater detection performance. Lastly, the detection results and current tracklets are inputted into the data association module to obtain trajectories.

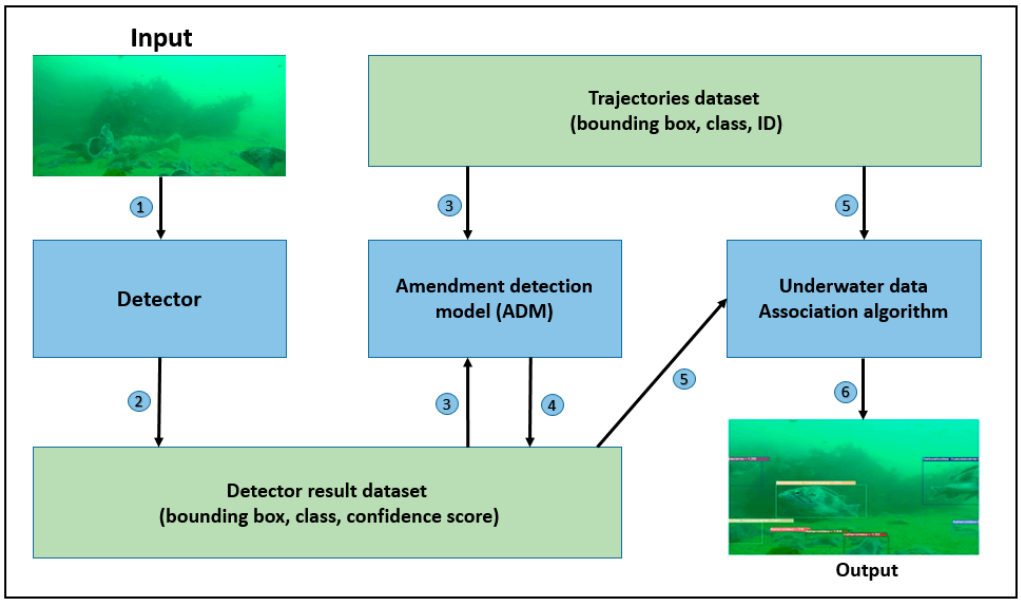


Figure 1. Flow chart of the FSTA. It comprises three modules, a detection result set, and a tracking result set.

3.1. Amendment Detection Model

Underwater videos have problems of color distortion and blur, and fish features change over time. This can lead to inconsistent detection results. Therefore, we used tracklets to amend the detection results and improve the performance of the underwater detection model. The detection result seriously affects the result of the tracking algorithm on the basis of the tracking-by-detection paradigm. Here is an underwater scenario to illustrate the necessity of this module. In Figure 2, black is Fish I, purple is Fish II, and red is Fish III. The first row at the bottom is the confidence score output by the detector model, and the second row is the modified confidence score. Fish II occluded Fish I at Point A, so Fish I was lost at Point A. When Fish I reappeared at Point C, it was partially occluded by Fish III, resulting in low confidence. Then, Fish I was not matched with the existing trajectory in the data-associated algorithm at Point C. Although the confidence score of the fish was low, we can infer that the reappeared fish was Fish I by observing the trajectory at Point C. The idea of this module is to use the current trajectories as prior knowledge. The

fish's confidence improves after the correction at Point C. The object can participate in the subsequent data association algorithm to improve the tracking performance.

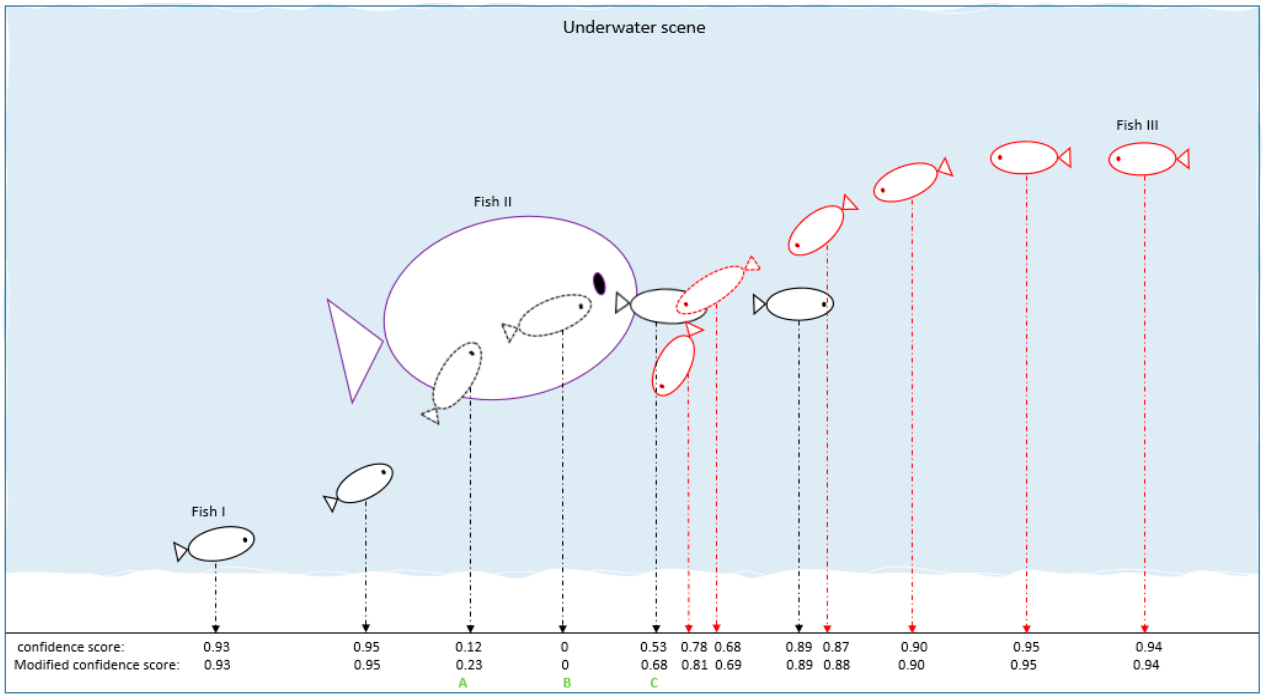


Figure 2. Underwater occlusion scene. Three fish scenarios were simulated, and the confidence scores before and after revision are listed. The A, B and C are three time points.

The pseudocode of the ADM algorithm is shown below.

The input of ADM is a detection set DS along with tracklet set γ and Kalman Filter KF. We also set a parameter σ, which is an adjustable parameter related to the current image size and blur level. The output of ADM is the detection set D of the current frame containing the bounding box and class of the object. Tracklet set γ is empty in the first frame. We merge tracked set τ_{track} and currently lost set γ_{lost} into one set, γ_{pool}. Then, we use Kalman filter KF to predict the new locations of each tracking bounding box. After that, we calculate the distances between each tracking box and each detection box usingIoU distance. (Lines 1 to 8 in Algorithm 1).

Algorithm 1: Pseudocode of ADM

Input: detection set DS, tracklet set γ, adjustment parameter σ, Kalman filter KF.
Output: detection set DS

1. Initialization: $dist \leftarrow 0$
2. $\gamma_{pool} = \gamma_{track} + \gamma_{lost}$
3. **for** t in γ_{pool} **do**
4. $\quad t \leftarrow KF(t)$
5. **end**
6. **for** i, j in DS, γ_{pool} **do**
7. $\quad dist_{i,j} \leftarrow$ *calculate the distance between i and j using IoU distance*
8. **end**
9. **for** i in DS **do**
10. $\quad dist_{i,*} = softmax(dist_{i,*})$
11. $\quad i.score = i.score + (e^{dist_{i,*}} - 1)/\sigma$ /*($i.score$ means the confidence score of detected object i)*/
12. **end**
13. **Return** DS

Lastly, the softmax function normalized the distance between object i and all detection boxes. Then, we modify the object's confidence score with IoU distance, and σ is an adjustable parameter related to the current image size and blur level. The output of ADM is the detection set DS of the current frame containing the bounding box and class of the object (Lines 9 to 13 in Algorithm 1).

3.2. Data Association Algorithm

We introduced the BYTE data association [31] algorithm and improved the algorithm using the re-ID feature extractor network. Since fish are non-rigid organisms, and the motion model was complex, the Kalman filter was mainly modeled with a uniform motion model, so it was not reliable to predict the coordinate position of fish by only using position information. This tracks a false object when occlusion occurs in the video. Therefore, we combined the representation feature of the object and its location information. The tracking loss of the current data association algorithm was high, and we designed a new data association algorithm. The occlusion of the object changed over time; thus, the weights of the representation and location features were adjusted automatically to improve the performance.

The input of the data association algorithm is a video sequence S along with an object detector and Kalman filter KF. We also set five thresholds: ξ_{high}, ξ_{low}, ϵ, $lost_{buffer}$, and $pair_{val}$. γ_{high} and γ_{low} are the detection score thresholds, ϵ is the tracking score threshold, $pair_{val}$ is the match score threshold, and $lost_{buffer}$ is the max lost frame number. The output of the data association algorithm is tracks γ of video sequence S.

Each frame obtains the bounding boxes, confidence scores, and coordinates using the detector. All the detection boxes are divided into two parts, D_{high} and D_{low}, according to their detection score. The boxes whose scores are higher than ξ_{high} are classified as D_{high}, and the boxes whose scores are between ξ_{high} and ξ_{low} are classified as D_{low} (Lines 3 to 13 in Algorithm 2).

Algorithm 2: Pseudocode of data association algorithm

Input: video sequence S, object detector, re-ID feature extractor Ext, Kalman filter KF, detection score threshold ξ_{high}, ξ_{low}, tracking score threshold ϵ, match score threshold $pair_{val}$, lost buffer frame number $lost_{buffer}$.

Output: Tracks γ_{track} of the video.

1. Initialization: $\gamma_{track} \leftarrow \varnothing$
2. **for** frame f_k in S **do**
3. $D_k \leftarrow Det(f_k)$
4. $D_{high} \leftarrow \varnothing$
5. $D_{low} \leftarrow \varnothing$
6. **for** d in D_k **do**
7. **if** $d.score > \xi_{high}$ **then**
8. $D_{high} \leftarrow D_{high} \cup \{d\}$
9. **end**
10. **else if** $d.score > \xi_{low}$ and $d.score < \xi_{high}$ **then**
11. $D_{low} \leftarrow D_{low} \cup \{d\}$
12. **end**
13. **end**
14. $\gamma_{pool} = \gamma_{track} + \gamma_{lost}$
15. **for** t in γ_{pool} **do**
16. $t \leftarrow kf(t), Ext(t)$
17. **end**
18. /*high score association*/

19. $M_{pair} \leftarrow$ *associate* γ_{pool} *and* D_{high} *using IoU distance*
20. $D_{remain} \leftarrow$ *remaining objects boxes from* D_{high}
21. $\gamma_{remain} \leftarrow$ *remaining objects boxes from* γ_{pool}
22. **for** m in M_{pair} **do**
23. **if** $m.score < pair_{val}$ **then**
24. $\gamma_{remain} \leftarrow m$
25. /*low score association*/
26. $D_{low} = D_{low} \cup D_{remain}$
27. $cost1 \leftarrow$ *calculate cost between* γ_{remain} *and* D_{low} *using IoU distance*
28. $cost2 \leftarrow$ *calculate cost between* γ_{remain} *and* D_{low} *using* $Re-ID$ *distance*
29. $cost = 7 \times cost2 \times min(1, e^{\frac{log2 \times \gamma_{remain}.lostlen}{lost_{buffer}}} - 1) + cost1 \times max(0, 2 - e^{\frac{log2 \times \gamma_{remain}.lostlen}{lost_{buffer}}})$

/ * ($\gamma_{remain}.lostlen$ means the lost frame number of the object in the remaining track set)*/

30. Associate γ_{remain} and D_{low} *using cost*
31. $\gamma_{re-remain} \leftarrow$ *remaining tracks from* γ_{remain}
32. /* delete unmatched tracks*/
33. $\gamma_{track} \leftarrow \gamma_{pool} \backslash \gamma_{re-remain}$
34. **for** d in D_{remain} *do*
35. **if** $d.score > \epsilon$ **then**
36. $\gamma_{track} \leftarrow \gamma_{track} \cup \{d\}$
37. **end**
38. **end**
39. **end**
40. **Return** γ_{track}

Tracked set γ_{track} and lost set γ_{cost} are merged into one set, γ_{pool}. Then, a Kalman filter (KF) is used to predict the new coordinates of each track, as performed in ADM. We also use a re-ID network to obtain the representation feature of each object in track set γ_{pool} (Lines 14 to 17 in Algorithm 2).

Next, we match detection bounding boxes D_{high} with the objects in track set γ_{pool}. The IoU is used to compute the similarity of all possible pairs, and we use a binary graph maximal matching algorithm, such as the Hungarian algorithm [36], to accomplish their matching. Then, all the matched pairs are filtered by the Euclidean distance of their appearance feature, whose distance over $pair_{val}$ remains. Unlike tracking methods, only using predicted position information is easily affected by the distance of the boxes. This situation affects lost-target tracking and the matching false detection box on the track. The proposed method adds the apparent feature scores to discriminate between the different objects. At the end of this stage, only the unmatched detection objects in D_{remain} and the unmatched track set objects in γ_{remain} remain (Lines 17 to 24 in Algorithm 2).

In the second stage of association, unmatched first-stage high-score detection set D_{remain} is added to low-score detection set D_{low}. We associate objects between low detection set D_{low} and remaining track set γ_{remain}. We calculate the cost between γ_{remain} and D_{low} by the IoU distance and the re-ID feature distance. This cost change over lost time. The Hungarian algorithm is used to finish the matching on the basis of the cost (Lines 25 to 31 in Algorithm 2).

All remaining unmatched tracklets $\gamma_{re-remain}$ after the second stage of the association are removed from tracked tracklets γ_{track} if they are in γ_{track}, and added into lost tracklets γ_{lost}. For simplicity, the procedure of tracking rebirth [21,30,37] is not shown in Algorithm 2. Only if the target is not tracked for over 30 frames, i.e., the tracklet is added into γ_{lost} for over 30 frames, it is deleted from lost tracklet γ_{lost}, which means that the target left the camera forever (Lines 32 to 33 in Algorithm 2).

Lastly, the detection boxes that satisfy the following conditions are used to initialize new tracklets (Lines 33 to 39 in Algorithm 1). (1) They are from D_{remain}, which remains in the first association stage and is still not matched in the second association stage. (2) The detection scores are higher than ϵ, where $\epsilon > \xi_{high}$. (3) The boxes are detected over two consecutive frames.

3.3. Re-ID Network

The performance of the re-ID network affects the results of the subsequent data-matching algorithm in multiple-target tracking experiments. So, it is necessary to obtain accurate feature extraction results from the re-ID network. The fish swim in all directions in the extraction process, and their characteristics vary a lot. In view of these facts, we adopte the heuristic dynamic feature fusion strategy and integrate the fish features extracted from multiple viewing angles to improve the feature representation ability of the network. A re-identification neural network is used to obtain the representation feature of the object. Thus, reducing the feature distances of the same object while enlarging the feature distances of different objects is a re-identification task. When occlusion and high turbidity occur, a common re-identification neural network has a poor effect and cannot enlarge the feature distance of different objects. Most research uses triplet loss to obtain more stable features. Furthermore, the triplet loss function is improved to achieve better performance. We improve the loss function by replacing instance-based loss with object-based loss. The feature of the object is gradually robust with the increase in the number of successfully matched frames, so the object could be rematched according to the local feature when occlusion occurrs.

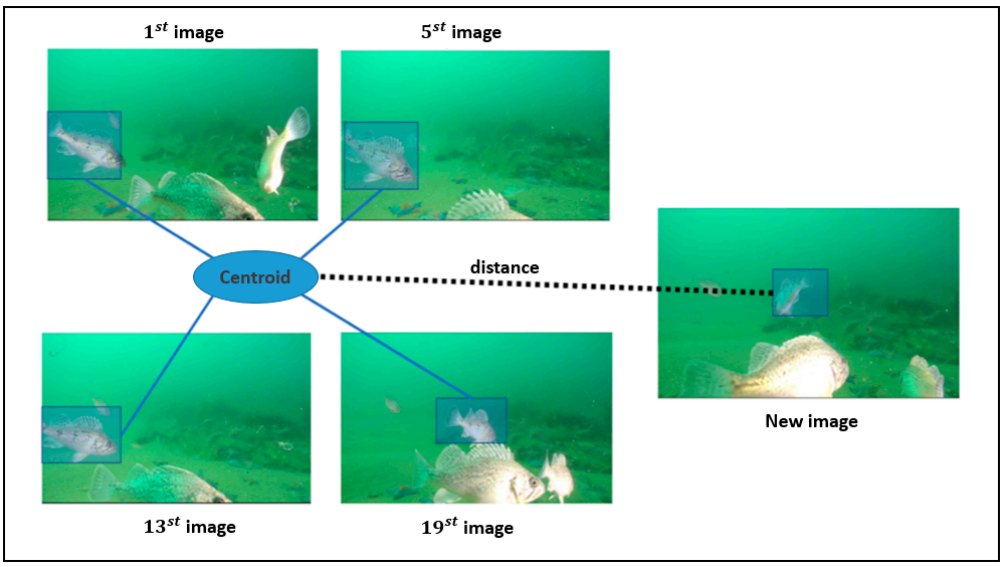


Figure 3. Centroid is calculated as the mean of all samples belonging to each class. When an object is matched, its feature is added to the centroid calculation.

Instead of using the triple loss function, the centroid triplet loss function is performed to train the re-identification network. The centroid triplet loss function is formulated in Equation (1), where $[z]+ = max(z, 0)$, f denotes the re-identification network, and A is an anchor instance. Centroid triplet loss (CTL) computes the distance among an anchor instance, a positive class centroid C_P and a negative class centroid C_N.

$$L_{triplet} = [\|f(A) - C_P\|_2^2 - \|f(A) - C_N\|_2^2 + \alpha]_+ \quad (1)$$

The centroid is calculated as the mean feature of the same object in the video frames, as shown in Figure 3. Thus, each object is represented by a single embedding along its lifetime, reducing retrieval time and storage requirements. The backbone of the re-ID network is Resnet50-IBN. Most CNNs use instance normalization (IN) or batch normalization alone. High-level visual tasks such as recognition use BN as a critical component to improve learning ability, while low-level visual tasks such as image style transformation use IN to remove the changing parts of pictures. IN eliminates the difference in individual appearance, but simultaneously reduces useful information. Resnet50-IBN reasonably combines IN and BN, which improves learning and generalization abilities.

4. Experiments and Results

To demonstrate the perfromance of the proposed fish tracking algorithm, we first elaborate on the dataset and implementation details in Section 4.1. We compare the FSTA with other related methods and evaluate different settings to justify our design choices in Section 4.2. Then, we show that our correlation tracker outperformed the state-of-the-art methods on five MOT benchmarks in Section 4.3. Lastly, we visualize the tracking trajectories in Section 4.4.

4.1. Dataset and Implementation Details

Dataset. Our neural network consist of the detection and re-identification networks. We used 5300 images as the training dataset of the detection network. The images were all extracted from the observation video of a marine pasture over one year. LabelImage software labelled the image according to the VOC format [38]. Second, 6318 images were used as the training dataset of the re-identification network; the format of the dataset was similar to that of DeepFashion [39]. The dataset is available at https://drive.google.com/file/d/15BDaciElZRaAIZgDRFjl1iq_sVseDRqP/view?usp=sharing (accessed on 18 August 2022).

Implementation details. The training and reference of the algorithm were carried out on a server configured with Intel(R) Xeon(R) Gold 6342 CPU × two and NVIDIA A100 40 GB × 4 in this experiment. We used PyTorch to build the backbone network and ran it on the Ubuntu 18.04 LTS operating system.

We adopted YoloX as the detector and trained it on the input with a resolution of 1920 × 1080. We used random flip, random scaling (between 0.5 to 0.9), cropping, and data mosaic as data augmentation, and Adam to optimize the overall objective. The learning rate was initialized as 1×10^{-4} and then decayed to 1×10^{-6} in the last 20 epochs. We trained with a batch size of 32 (on 4 GPUS) for 120 epochs. When using the Adam optimizer, some networks may fall off a cliff and then fix at a value from which they are no longer able to recover. By reducing the learning rate to some threshold, such as 0.0001, unstable cases can be solved; the step size is then reduced to 0.000001 to ensure that the model gradually converges.

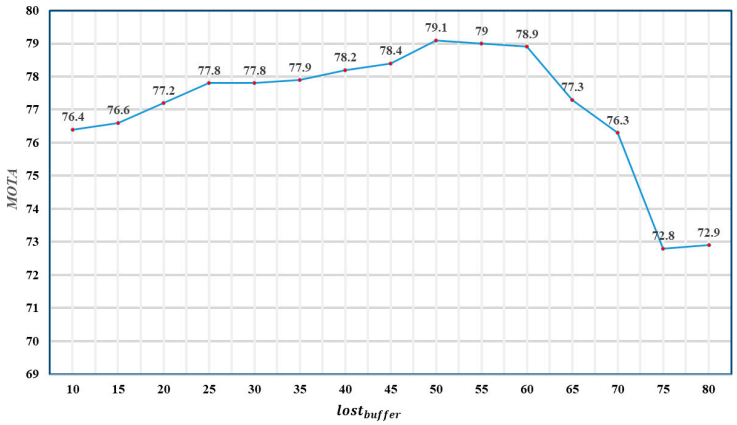
The backbone network of the re-identification network is Resnet50-IBN; the implementation and hyperparameters mostly follow Pan et al. and Wieczorek et al. [40,41]. The backbone network was pretrained on ImageNet, and then we fine-tuned the network in our dataset. We used $stride = 1$ for the last convolutional layer and modified it to a 512-dimensional embedding size. The loss functions were similar to those in He et al. and Wieczorek et al. [42,43], and we used triplet loss, center loss, and classification loss. The center loss was weighted by a factor of 5×10^{-4}, and all other losses were assigned a weight of 1. The learning rate was initialized as 5×10^{-4} and then decayed to 1×10^{-5} in the last 25 epochs. The models were trained for 150 epochs because the loss function curve converged at the 150th epoch on the basis of our dataset. Note that, this value is related to the size of the dataset. Different datasets have different convergence speed levels. Usually, the larger the dataset is, the fewer the training epochs are needed.

In the data association algorithm, detection score threshold τ_{high} was set to 0.7, and τ_{low} was set to 0.2. The tracking score threshold ϵ was set to 0.6. The matching score threshold $pair_{val}$ was set to 0.5. The maximal lost frame $lost_{buffer}$ was set to 50.

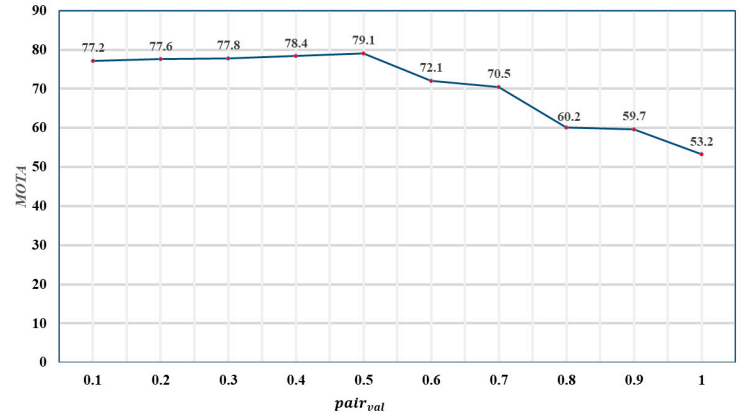
4.2. Hyperparameter Setting and Network Experimentaion

We first tested the hyperparameter setting. As Figure 4a shows, when $lost_{buffer} < 50$, MOTA does not decrease with the increase in $lost_{buffer}$. When $lost_{buffer} = 50$, the maximal MOTA reaches 79.1. Thus, $lost_{buffer}$ was set to 50 in this study. This value is related to the video's frame rate and the fish's swimming speed. The frame rate of the video that we tested is 30. If the frame rate of the video is not 30, additional adjustment of this hyperparameter would be needed. With a fixed $lost_{buffer}$, we searched for the proper value of $pair_{val}$. This value was used to control the matching accuracy; we only used the position

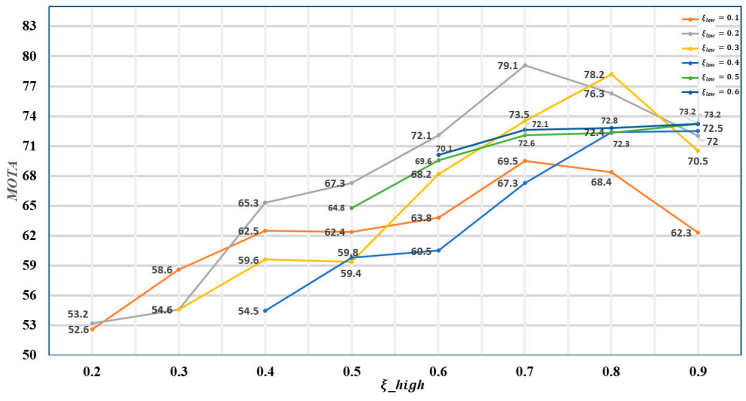
prediction algorithm in the first association. When this value was too large, most successful matches were not filtered, so many wrong matches were considered to be correct, and the MOTA was significantly reduced. Figure 4b shows that, when $pair_{val}$ is 0.5, the result is the best. ξ_{high} and ξ_{low} controll the confidence score range of the two-stage data association algorithm and, to some extent, determine the reliability of the Kalman filter. In Figure 4c, we present the tracking performance under different ξ_{high} and ξ_{low}. The result is better when the difference between the two values is around 0.5, among which ξ_{high}= 0.7 and ξ_{low}= 0.2 have the best result. When $\xi_{high} = \xi_{low}$, it becomes a one-stage data association algorithm using only position prediction.



(**a**)



(**b**)



(**c**)

Figure 4. Tracking performance of FSTA with different hyperparameters. (**a**) Tracking performance with different $lost_{buffer}$; (**b**) tracking performance with different $pair_{val}$; (**c**) tracking performance with different ξ_{high} under different ξ_{low}.

Table 1 presents the achieved performance using different backbone models of reidentification networks. The top MOTA and the highest IDF1 are obtained with ResNet50-IBN. ResNet50-IBN means that we added IBN-Net into ResNet50. IBN-Net carefully integrated instance normalization (IN) and batch normalization (BN) as building blocks, and could be wrapped into ResNet50 networks to improve its performance without increasing computational cost. IN learns invariant features to appearance changes, such as colors, styles, and virtuality or reality, by delving into IN and BN. At the same time, BN is essential for preserving content-related information. This backbone is considered to run all the following experiments.

Table 1. Comparison of different backbone networks on the validation set. The best results are shown in bold.

Model	MOTA↑	IDF1↑	ID Sw.↓	FPS↑
ResNet50	76.5	76.4	350	**30**
DenseNet	77.9	78.2	278	29.4
Darknet-53	78.1	78.6	230	29.3
ResNet50-IBN	**79.1**	**80.3**	**210**	29

Table 2 shows the obtained performance by considering different input image resolutions. Better performance could be achieved with high image resolutions, but at a higher computational cost.

Table 2. Comparison of different resolutions on the validation set. The best results are shown in bold.

Input Size	MOTA↑	IDF1↑	ID Sw.↓	FPS↑
256 × 256	76.5	76.4	350	**30**
800 × 533	77.9	78.2	278	29.4
1024 × 683	78.1	78.6	230	29.3
1920 × 1080	**79.1**	**80.3**	**210**	29

4.3. Ablation Study

We tested the effect of the above modules on the results, as shown in Table 3. The baseline model was ResNet50 with a Kalman filter algorithm. Re-ID indicates that we add the representation feature into the data association algorithm in the baseline model. Then, the amendment detection model (ADM) is added into the baseline model. Centroid loss indicates the formula that is used as triplet loss. The baseline performs similarly to SORT, which has the best FPS, but we significantly improve the multiple-object tracking accuracy (MOTA), and the introduced representation features substantially reduce the ID switches.

Table 3. The influence of different modules on algorithm performance. The best results are shown in bold.

Model	MOTA↑	IDF1↑	ID Sw.↓	FPS↑
Baseline	71.6	74.4	405	**32**
+Re-id	75.3	76.2	315	29.5
+ADM	77.9	79.4	294	29
+Centroid loss	**79.1**	**80.3**	**210**	29

4.4. State-of-the-Art Comparisons

We compared FSTA with other popular association methods: SORT [29], DeepSORT [30], JDE [23] and ByteDance [31]. The results are shown in Table 4.

Table 4. Comparison of different data association methods on the validation set. The best results are shown in bold.

Method	w/Re-ID	MOTA↑	IDF1↑	IDs↓	FPS↑
SORT		71.6%	75.4%	342	**30.1**
DeepSORT	✓	75.4%	77.4%	301	19
J.D.E.	✓	76.1%	72.2%	453	30
ByteDance		77.6%	79.3%	249	29.6
FSTA (ours)	✓	**79.1%**	**82.3%**	**160**	29

SORT can be considered as our baseline method because all methods only use a Kalman filter to predict the object's motion. The FSTA improved the MOTA metric of SORT from 71.6% to 79.1% and IDF1 from 75.4% to 82.3%, and decreased the IDs from 342 to 160. This highlights the importance of the representation features, and proves the ability of FSTA to recover object boxes from a low-score one.

DeepSORT uses additional re-ID models to enhance long-range association. We surprisingly observed that the FSTA also had more significant gains than those of DeepSORT, which suggests that the cascade association is not the best algorithm, which could be improved by optimizing the data association structure and re-ID network.

JDE proposed a MOT system that allows for a shared model to learn target detection and appearance embedding. Specifically, it incorporates the appearance embedding model into a single-shot detector, such that the model could simultaneously output detections and the corresponding embeddings. The FSTA improves the MOTA metric of JDE from 76.1% to 79.1% and IDF1 from 72.2% to 82.3%, and decreases IDs from 453 to 160. It uses the idea of classification to fit the target ID, which causes the ID switch to be too large.

ByteDance suggests that a simple Kalman Filter can perform long-range association, and achieve better IDF1 and IDs when the detection boxes are accurate. However, it is difficult for the motion model to refine a lost object in severe occlusion cases. Therefore, representation features behave more reliably.

4.5. Visualization

The specific algorithm effect is shown in Figure 5, in which frames of 10, 15, 40, 70, 100, and 130 s were intercepted from test video data. The mark in the figure is the scientific name and count of each category, respectively. This algorithm can be used as a long-term recognition algorithm for an underwater observation platform, and it not only meets the real-time requirements, but also has a high accuracy rate.

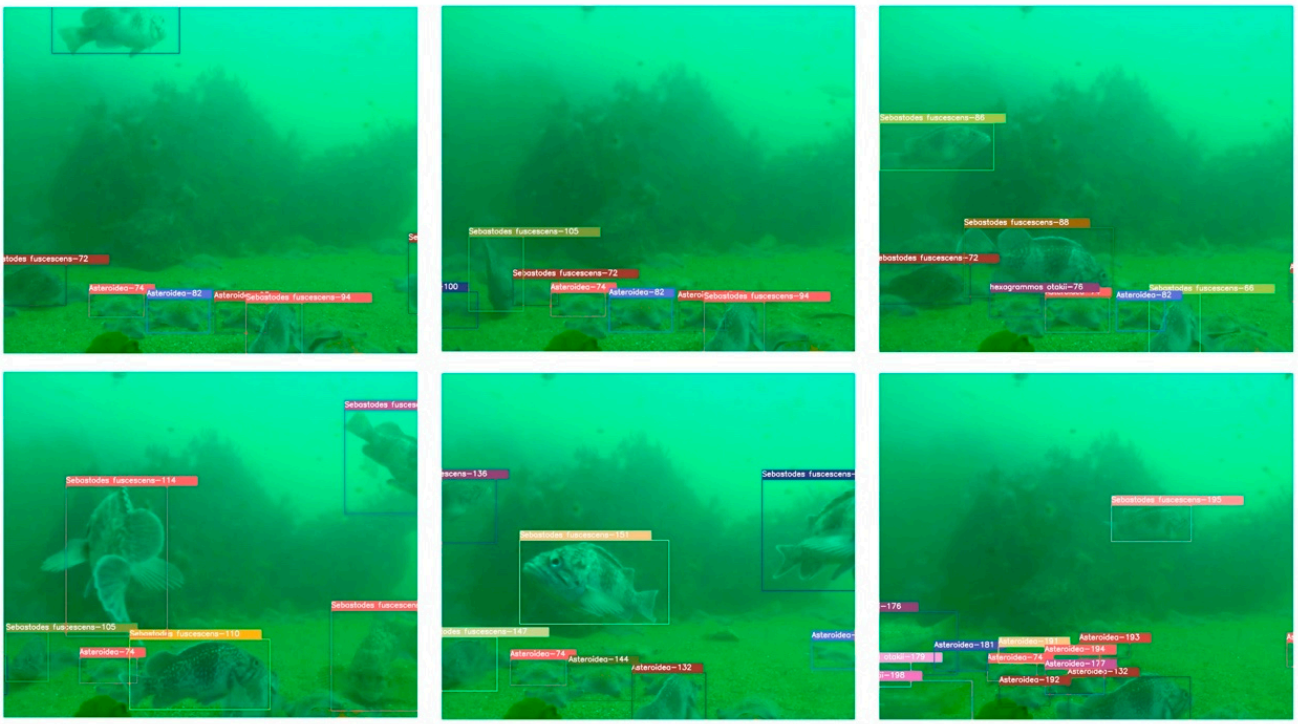


Figure 5. Visualization result of the FSTA. We selected six sequences from the validation set of our dataset and showed the effectiveness of FSTA to handle complex cases such as occlusion and motion blur.

5. Discussion

By adding the robust Re-ID algorithm proposed in this study, the centroid-based matching algorithm improves its accuracy over time because more complete organism characteristics could be learned. We provide a visual comparison between our algorithm and another algorithm that only uses the location information in Figure 6. There were three fish swimming in overlapping occlusion over time. We use a triangle to represent each fish, and the solid purple lines successfully represente the matched object. The dotted line is the unsuccessful match. Algorithm 1 is ours, and Algorithm 2 only uses the Kalman filter to predict the position of the coordinate to match the object. The fish swimming cover problem occurred in the 67th frame, and the algorithm lost the original target, but when Fish III reappeared in the 101st frame, Algorithm 1 could track the target. Algorithm 2 lost the target and gave it a new ID, which added tracking error.

This technology has its shortcomings. The motion model of Kalman filtering is a uniform motion model. When there is a large wave or fish mutation, the model affects the tracking result because fish are nonrigid. The problem can be solved through an improved motion model of the Kalman filter. Still, it is also difficult to build the fish motion model. We also tested the nonlinear particle filter with stronger overfitting ability, but its performance was not stable because it could only obtain local optimal solutions. If the global optimal solution could be obtained, the tracking performance could be effectively improved, and this could be a future breakthrough point.

Through experimental tests, the recognition accuracy rate dropped by 1.5% to 2.3% in other sea areas. This was due to different sea areas' color cast and contrast differences. So, when the FSTA is applied to different sea areas, the network weight coefficients need to be retrained accordingly. By adding different white balance methods to process the video first, since the parameters of white balance need to be manually adjusted, the most appropriate method should introduce a color complement neural network before image classification. This would be another future breakthrough point.

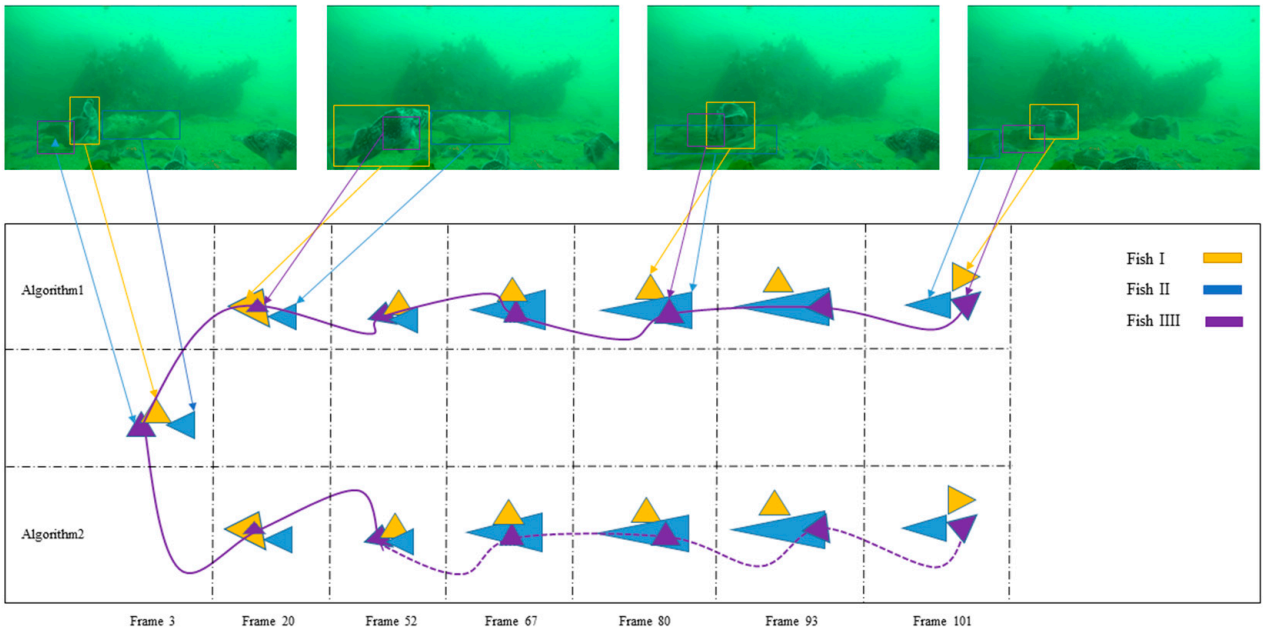


Figure 6. Visualization result of the different algorithms. We selected seven sequences from the validation set of our dataset, and show the effectiveness of FSTA in handling occlusion.

This study is expected to have good prospects. First, our model was stably tested in marine ranches, and could achieve long-term and highly accurate underwater biological tracking. Second, our algorithm could be applied not only to underwater loading equipment, but also to underwater robots, underwater AUVs, and other equipment. The algorithm in this paper has the potential to realize automatic statistics in marine environments, and it can be a useful complement to traditional manual statistical methods. In addition, the algorithm provides a means to analyze underwater biological environments. By tracking the trajectory of fish, fish behavior can be analyzed; current water-quality and environmental changes can be further analyzed to provide an additional verification of observer data. The technology only needs to rely on high-definition underwater cameras without special requirements for the cameras, but there are certain restrictions on the observation and tracking time. For example, the time during which we could track was mainly during the day. Accuracy dropped at night and in particularly murky water, so there are specific light requirements.

6. Conclusions

Tracking methods based on deep learning rarely involve underwater-object tracking. Compared with processing in non-underwater photos or videos, underwater fish tracking is challenging due to varying marine environments and poor-quality underwater imaging. In this paper, we proposed a robust tracker FSTA, and its accuracy achieves 79.1% of MOTA and 82.3% of IDF1 on the basis of our underwater dataset.

The FSTA, as a new paradigm for underwater fish tracking, has the potential to be widely applied to aquatic observation systems. First, the image quality of underwater videos suffers from color distortions, deformations, low resolution, and contrast, and fish features vary over time. This also leads to inconsistent detection results because of the sudden low confidence scores during detection. Therefore, in this study, trajectories have been used as prior knowledge to amend the detection results to improve the performance of the underwater detection model. Second, a new data association algorithm is proposed by recombining representation and location information. In this algorithm, the weight parameters of the position and apparent information are dynamic. The loss value could be adjusted according to the lost situation to improve matching performance. We recovered objects from the low-score detection box while filtering out the background. The algorithm

is robust to occlusion due to dynamic association, which improves retracing lost targets. Lastly, we introduce the centroid triplet loss function to train the RestNet50-IBN feature extraction network. This greatly enhances the performance of the tracking algorithm.

The different hyperparameters affect our algorithm's performance. The effects of the three components of the algorithm in this paper have been tested with ablation experiments. The experimental results show that each element improved the model's performance in a different aspect, and among them, the Re-ID network showed superior performance. Also, the results of our model were compared with the performance of state-of-the-art algorithms, and our algorithm has indices in the leading position for the underwater dataset. In addition, the shortcomings of the model has also been discussed in this paper, and further improvement could be achieved by focusing on the unresolved complex fish aliasing phenomena.

Author Contributions: T.L. conceived the idea and performed the experiment; S.H. and H.L. analyzed the data and investigated method. T.L. wrote the original manuscript. S.H., Y.G. and P.L. reviewed and edited the paper. All authors have read and agreed to the published version of the manuscript.

Funding: The project was funded by the Major Science and Technology Project of Sanya [funding no. SKJC-KJ-2019KY03], the Key Research and Development Plan of Zhejiang Province [funding no. 2020C03012], the Finance Science and Technology Project of Hainan Province [funding no. ZDKJ202019], and the High-Level Personnel of Special Support Program of Zhejiang Province [funding no. 2019R52045].

Data Availability Statement: Our labeled underwater datasets, including detection and tracking datasets, are available at https://drive.google.com/file/d/15BDaciElZRaAIZgDRFjl1iq_sVseDRqP/view?usp=sharing (18 August 2022).

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have to influence the work reported in this paper.

References

1. *The State of World Fisheries and Aquaculture*; FAO: Rome, Italy, 2020; ISBN 9789251327739.
2. Lorenzen, K.; Cowx, I.G.; Entsua-Mensah, R.E.M.; Lester, N.P.; Koehn, J.D.; Randall, R.G.; So, N.; Bonar, S.A.; Bunnell, D.B.; Venturelli, P.; et al. Stock assessment in inland fisheries: A foundation for sustainable use and conservation. *Rev. Fish Biol. Fish* **2016**, *26*, 405–440. [CrossRef]
3. Li, D.; Hao, Y.; Duan, Y. Nonintrusive methods for biomass estimation in aquaculture with emphasis on fish: A review. *Rev. Aquac.* **2020**, *12*, 1390–1411. [CrossRef]
4. Melnychuk, M.C.; Peterson, E.; Elliott, M.; Hilborn, R. Fisheries management impacts on target species status. *Proc. Natl. Acad. Sci. USA* **2017**, *114*, 178–183. [CrossRef]
5. Saberioon, M.; Císař, P. Automated within tank fish mass estimation using infrared reflection system. *Comput. Electron. Agric.* **2018**, *150*, 484–492. [CrossRef]
6. Zion, B. The use of computer vision technologies in aquaculture—A review. *Comput. Electron. Agric.* **2012**, *88*, 125–132. [CrossRef]
7. Li, D.; Wang, Z.; Wu, S.; Miao, Z.; Du, L.; Duan, Y. Automatic recognition methods of fish feeding behavior in aquaculture: A review. *Aquaculture* **2020**, *528*, 735508. [CrossRef]
8. Hu, J.; Zhao, D.; Zhang, Y.; Zhou, C.; Chen, W. Real-time nondestructive fish behavior detecting in mixed polyculture system using deep-learning and low-cost devices. *Expert Syst. Appl.* **2021**, *178*, 115051. [CrossRef]
9. Xu, W.; Zhu, Z.; Ge, F.; Han, Z.; Li, J. Analysis of Behavior Trajectory Based on Deep Learning in Ammonia Environment for Fish. *Sensors* **2020**, *20*, 4425. [CrossRef]
10. Yu, X.; Wang, Y.; An, D.; Wei, Y. Identification methodology of special behaviors for fish school based on spatial behavior characteristics. *Comput. Electron. Agric.* **2021**, *185*, 106169. [CrossRef]
11. Barreiros, M.D.O.; Dantas, D.D.O.; Silva, L.C.d.O.; Ribeiro, S.; Barros, A.K. Zebrafish tracking using YOLOv2 and Kalman filter. *Sci. Rep.* **2021**, *11*, 3219. [CrossRef]
12. Villar, S.A.; Madirolas, A.; Cabreira, A.G.; Rozenfeld, A.; Acosta, G.G. ECOPAMPA: A new tool for automatic fish schools detection and assessment from echo data. *Heliyon* **2021**, *7*, e05906. [CrossRef]
13. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.

14. Boom, B.J.; Huang, P.X.; He, J.; Fisher, R.B. Supporting Ground-Truth Annotation of Image Datasets Using Clustering. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Tsukuba, Japan, 11–15 November 2012; pp. 1542–1545.
15. Wang, S.H.; Zhao, J.W.; Chen, Y.Q. Robust tracking of fish schools using CNN for head identification. *Multimed. Tools Appl.* **2017**, *76*, 23679–23697. [CrossRef]
16. Yang, X.; Zhang, S.; Liu, J.; Gao, Q.; Dong, S.; Zhou, C. Deep learning for smart fish farming: Applications, opportunities and challenges. *Rev. Aquac.* **2021**, *13*, 66–90. [CrossRef]
17. Piccardi, M. Background subtraction techniques: A review. In Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583), The Hague, The Netherlands, 10–13 October 2004; Volume 4, pp. 3099–3104.
18. Rasmussen, C.E. The Infinite Gaussian Mixture Model. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2000; Volume 7.
19. Welch, G.; Bishop, G. *An Introduction to the Kalman Filter*; University of North Carolina: Chapel Hill, NC, USA, 2006; p. 16.
20. Gustafsson, F. Particle filter theory and practice with positioning applications. *IEEE Aerosp. Electron. Syst. Mag.* **2010**, *25*, 53–82. [CrossRef]
21. Zhang, J.; Zhou, S.; Chang, X.; Wan, F.; Wang, J.; Wu, Y.; Huang, D. Multiple Object Tracking by Flowing and Fusing. *arXiv* **2020**, arXiv:2001.11180.
22. Mohamed, H.E.-D.; Fadl, A.; Anas, O.; Wageeh, Y.; ElMasry, N.; Nabil, A.; Atia, A. MSR-YOLO: Method to Enhance Fish Detection and Tracking in Fish Farms. *Procedia Comput. Sci.* **2020**, *170*, 539–546. [CrossRef]
23. Wang, Z.; Zheng, L.; Liu, Y.; Li, Y.; Wang, S. Towards Real-Time Multi-Object Tracking. In Proceedings of the Computer Vision—ECCV 2020, Glasgow, UK, 23–28 August 2020; pp. 107–122.
24. Zhang, Y.; Wang, C.; Wang, X.; Zeng, W.; Liu, W. FairMOT: On the Fairness of Detection and Re-Identification in Multiple Object Tracking. *Int. J. Comput. Vis.* **2021**, *129*, 3069–3087. [CrossRef]
25. Liu, T.; Li, P.; Liu, H.; Deng, X.; Liu, H.; Zhai, F. Multi-class fish stock statistics technology based on object classification and tracking algorithm. *Ecol. Inform.* **2021**, *63*, 101240. [CrossRef]
26. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
27. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
28. Zheng, M.; Gao, P.; Zhang, R.; Li, K.; Wang, X.; Li, H.; Dong, H. End-to-End Object Detection with Adaptive Clustering Transformer. *arXiv* **2021**, arXiv:2011.09315.
29. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple Online and Realtime Tracking. *IEEE Int. Conf. Image Processing* **2016**, 3464–3468. [CrossRef]
30. Wojke, N.; Bewley, A.; Paulus, D. Simple Online and Realtime Tracking with a Deep Association Metric. *arXiv* **2017**, arXiv:1703.07402.
31. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. *arXiv* **2021**, arXiv:2110.06864.
32. Chuang, M.-C.; Hwang, J.-N.; Williams, K.; Towler, R. Tracking Live Fish From Low-Contrast and Low-Frame-Rate Stereo Videos. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 167–179. [CrossRef]
33. Palconit, M.G.B.; Almero, V.J.D.; Rosales, M.A.; Sybingco, E.; Bandala, A.A.; Vicerra, R.R.P.; Dadios, E.P. Towards Tracking: Investigation of Genetic Algorithm and LSTM as Fish Trajectory Predictors in Turbid Water. In Proceedings of the 2020 IEEE Region 10 Conference (TENCON), Osaka, Japan, 16–19 November 2020; pp. 744–749.
34. Sun, N.; Nian, R.; He, B.; Yan, T. Consistent fish tracking via multiple underwater cameras. In Proceedings of the OCEANS 2014–TAIPEI., Taipei, Taiwan, 7–10 April 2014; pp. 1–5.
35. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [CrossRef]
36. Kuhn, H.W. The Hungarian method for the assignment problem. *Nav. Res. Logist.* **2005**, *52*, 7–21. [CrossRef]
37. Chen, L.; Ai, H.; Zhuang, Z.; Shang, C. Real-Time Multiple People Tracking with Deeply Learned Candidate Selection and Person Re-Identification. In Proceedings of the 2018 IEEE International Conference on Multimedia and Expo (ICME), San Diego, CA, USA, 23–27 July 2018; pp. 1–6.
38. Everingham, M.; Eslami, S.M.A.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
39. Liu, Z.; Luo, P.; Qiu, S.; Wang, X.; Tang, X. DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1096–1104.
40. Pan, X.; Luo, P.; Shi, J.; Tang, X. Two at Once: Enhancing Learning and Generalization Capacities via IBN-Net. In *Proceedings of the Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 484–500.
41. Wieczorek, M.; Rychalska, B.; Dabrowski, J. On the Unreasonable Effectiveness of Centroids in Image Retrieval. *arXiv* **2021**, arXiv:2104.13643.

42. He, S.; Luo, H.; Chen, W.; Zhang, M.; Zhang, Y.; Wang, F.; Li, H.; Jiang, W. Multi-Domain Learning and Identity Mining for Vehicle Re-Identification. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 2485–2493.
43. Wieczorek, M.; Michalowski, A.; Wroblewska, A.; Dabrowski, J. A Strong Baseline for Fashion Retrieval with Person Re-identification Models. In *Proceedings of the Neural Information Processing*; Yang, H., Pasupa, K., Leung, A.C.-S., Kwok, J.T., Chan, J.H., King, I., Eds.; Springer: Cham, Switzerland, 2020; pp. 294–301.