Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

Research paper

USV-Tracker: A novel USV tracking system for surface investigation with limited resources

Tao Huang [a,c,1], Yiheng Xue [c,d,1], Zhenfeng Xue [b,c,*], Zheng Zhang [c,d], Zhonghua Miao [b], Yong Liu [a]

[a] *School of Control Science and Engineering, Zhejiang University, No. 38 Zheda Road, Hangzhou, 310027, China*
[b] *School of Mechatronic Engineering and Automation, Shanghai University, No. 99 Shangda Road, Shanghai, 200444, China*
[c] *Research Center for Intelligent Perception and Control, Huzhou Institute of Zhejiang University, No. 819 Xisaishan Road, Huzhou, 313098, China*
[d] *School of Advanced Technology, Xi'an Jiaotong-Liverpool University, No. 111 Ren'ai Road, Suzhou, 215123, China*

ARTICLE INFO

Keywords:
Unmanned surface vehicle
Object tracking
Motion planning
Kalman filtering

ABSTRACT

This paper introduces USV-Tracker, a novel tracking system for Unmanned Surface Vehicles (USVs) tailored for practical applications such as surface investigation and target tracking. The system tackles three pivotal challenges: perception robustness, tracking concealment, and planning efficiency. The contributions of this work are manifold: (1) A multi-sensor fusion framework utilizing an Extended Kalman Filter (EKF) to enhance target detection and positioning accuracy, integrating data from cameras, LiDAR, GPS, and IMU sensors. (2) A two-stage path planning algorithm that generates occlusion avoidance trajectories and employs a virtual elastic force constraint to maintain appropriate relative positioning. In dense obstacle environments, the algorithm tends to get closer to the target and incorporates FOV orientation constraints to ensure stable perception. (3) A visibility-aware control strategy that ensures continuous target observability through EKF-based trajectory prediction. Simulations in Gazebo and corresponding physical experiments validate the system's effectiveness and robustness, demonstrating its applicability in real-world scenarios. The computational workload is managed on a constrained on-board computer, underscoring the system's practicality.

1. Introduction

In recent years, the widespread application of drones (Muchiri and Kimathi, 2022; Dissanayaka et al., 2023) and unmanned vehicles (Szrek et al., 2020; Abd Rahman et al., 2022) across various industries has spurred rapid advancements in USV technologies (Chen et al., 2021; Huang et al., 2023a). USV tracking systems, used for applications such as environmental monitoring and surface investigation, aim to enable USVs to follow targets stably. However, most existing research remains theoretical and is not widely adopted in practice. Previous works (Sinisterra et al., 2017; Yu et al., 2019) have primarily focused on advanced control theory, striving for positional consistency with the target. Real-world applications involve greater complexity, including occlusion and collision with obstacles, temporary target loss, and the need for computational timeliness.

Three critical challenges must be addressed for practical USV tracking systems: perception robustness, target tracking concealment, and the tight coupling between perception and planning. Perception robustness requires maintaining consistent performance across diverse environments by leveraging the strengths of multiple sensors. The fusion of monocular cameras and multi-beam LiDAR enhances perceptual robustness, with cameras providing precise small object identification and LiDAR offering reliable distance estimation and local mapping. Additionally, the EKF and trajectory prediction algorithms further refine the accuracy of target position detection and forecast future positions. Target tracking concealment involves integrating the predicted target trajectory into the planning algorithm. An efficient elastic planning algorithm generates flexible and occlusion-avoidance trajectories while maintaining an optimal tracking distance and ensuring the target remains within the field of view (FOV). Elastic planning methods, adapted from drones (Han et al., 2021; Ji et al., 2022) to maritime environments, incorporate the three degrees of freedom of the hull into kinematic planning. A virtual force mechanism ensures that the USV maintains a suitable position and orientation relative to the target. These algorithms are optimized for real-time operation on a compact on-board computer, ensuring continuous and stable tracking. Tight coupling between perception and planning framework is essential. Accurate target trajectory information provided by perception is crucial for effective planning. The planning process considers the conditions necessary for successful perception, such as keeping the target within

* Corresponding author at: School of Mechatronic Engineering and Automation, Shanghai University, No. 99 Shangda Road, Shanghai, 200444, China.
E-mail address: zfxue0903@shu.edu.cn (Z. Xue).

[1] Equal contribution.

https://doi.org/10.1016/j.oceaneng.2024.119196
Received 31 May 2024; Received in revised form 19 August 2024; Accepted 4 September 2024

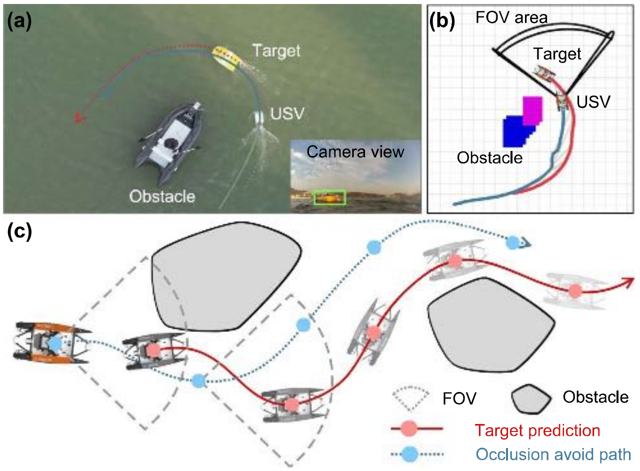


Fig. 1. Overview of the USV-Tracker. The blue line represents the predicted trajectory of the target, while the red line indicates the planned path of the USV, both incorporating strategies for obstacle avoidance and FOV constraints, (a) depicts the actual tracking system in operation blue, (b) shows the obstacle map utilized in the path planning task, (c) illustrates a diagram of a USV dynamically tracking a moving target, adjusting its course and camera FOV to navigate around obstacles and maintain consistent focus on the target.

the camera's FOV. This integration enhances system robustness, enabling the USV to continue tracking even if the target is temporarily lost using the EKF-predicted trajectory to guide the USV.

In this paper, we propose a novel tracking system named USV-Tracker, as depicted in Fig. 1, integrating these components into one unit. The system is validated through simulations and a compact USV prototype with standard sensors. The average target positioning error is sub-meter, significantly smaller than the target's size. The USV system demonstrates stable target tracking over extended periods in simulated and physical environments, underscoring its practical applicability.

The structure of this article is arranged as follows. Section 2 overviews of relevant methods. Section 3 describes the framework and critical challenges of the system. Section 4 details the path planning module. Section 5 presents the simulation and physical experiments. Finally, Section 6 offers the conclusions.

2. Related work

2.1. 3D perception and target tracking algorithm

Significant advancements have been achieved over the past decade in vision-based target detection. Two-stage approaches, known for their high accuracy, and one-stage methods, renowned for their efficiency, have substantially contributed to the development of the field. Recent efforts have concentrated on accelerating detection and enhancing accuracy, with technologies like TensorRT specifically supporting edge computing platforms. Furthermore, adapting Transformer technology from natural language processing has markedly improved detection accuracy on advanced computing platforms (Girshick, 2015; Ren et al., 2015; Redmon et al., 2016; Carion et al., 2020).

Point cloud target detection is an emerging area within 3D computer vision research. Initial methods, such as PointNet (Qi et al., 2017a), effectively handle rotation invariance and disorder in raw data but struggle with integrating regional information. Successive methods, including PointNet++ (Qi et al., 2017b) and PointRCNN (Shi et al., 2019), have addressed these limitations through hierarchical feature extraction and improved precision in object detection. Although voting mechanisms and Transformer models further enhance detection accuracy, they face significant challenges in mobile deployment due to high computational demands (Qi et al., 2019; Liu et al., 2021). Grid-based methods, such as MV3D (Chen et al., 2017) and AVOD (Ku et al., 2018), alongside voxel-based approaches like VoxelNet (Liu et al., 2021) and PV-RCNN (Shi et al., 2020), offer efficient strategies for edge applications. Grid-based techniques project 3D data onto grids for feature extraction, effectively combining multi-view features. Voxel-based methods, on the other hand, transform dense point clouds into a more manageable form through voxelization. While innovative, these methods encounter computational limitations when deployed in edge applications. Achieving a balance between accuracy and computational efficiency remains a crucial challenge for these methods.

Multi-sensor fusion and Bird's Eye View (BEV) representations have become increasingly pivotal in computer vision. Techniques like MV3D (Chen et al., 2017) and Frustum PointNets (Qi et al., 2018) generate 3D object proposals and merge 2D object detection with 3D deep learning, effectively localizing objects within dense point clouds. Despite their ingenuity, these methodologies often struggle with computational efficiency and real-time processing due to the inherent complexity of integrating multiple perspectives.

Recent advancements in BEV fusion are noteworthy. Methods such as Lift, Splat, Shoot (LSS) (Philion and Fidler, 2020), BEVFusion (Liu et al., 2023), and BEVDepth (Li et al., 2023) have significantly advanced the integration efficiency and accuracy of BEV representations. These techniques collectively epitomize the state-of-the-art BEV multi-sensor fusion, each contributing to the substantial progression of the field and establishing a robust foundation for future research.

2.2. USV motion planning and tracking methods

Previous research primarily focuses on studying the guidance control problem of USV individually. Some works (Breivik et al., 2008; Bibuli et al., 2012) adopt a Line of sight (LOS) guidance strategy to build a control closed loop. Chen et al. (2022) proposed a Particle Swarm Optimization controller in USV target tracking to obtain more stable tracking results in practice. Although real-world experiments verify these methods, they rely on known target status and do not consider obstacles within the surroundings. Agrawal and Dolan (2015) introduced the International Regulations for Preventing Collisions at Sea (COLREGS) rules into the path planning of target following to deal with other moving ships on the water surface during the following process. However, this method uses the search-based method to find the shortest path in the four-dimensional discrete space–time, which means a considerable time cost. Švec et al. (2014) uses the Monte Carlo sampling method to predict the target movement, making the planned trajectory more suitable for USV target tracking in the obstacle area. Lin et al. (2023) proposed an adaptive USV interception method based on a backstepping technique, which can complete the interception of a target USV within a limited time. A prevalent issue exists with the methods mentioned above. They do not consider the impact of target following motion on target perception. Two typical scenarios are the target escaping from the perception of FOV and the obscured target.

In order to make the USV tracking system more stable, it is essential to deal reasonably with the relationship between target perception and target tracking motion, especially underactuated USV restricted by non-holonomic constraints. Some trajectory planning methods combined with active perception (Zhou et al., 2021; Wang et al., 2023; Ji et al., 2022) effectively solve the interaction problem between perception and planning in Unmanned Aerial Vehicle (UAV) tracking. Zhou et al. (2021) proposed a risk-aware trajectory refinement method to optimize the trajectory twice according to the distribution of unknown areas in the environment and the orientation of the perception FOV, effectively improving the collision avoidance ability against obstacles in unknown areas.

However, multiple optimizations make it challenging to guarantee the optimality of the final generated trajectory. Ji et al. (2022) design an occlusion-aware path-searching method and define the visible region to establish the analytical occlusion cost so that the target can be

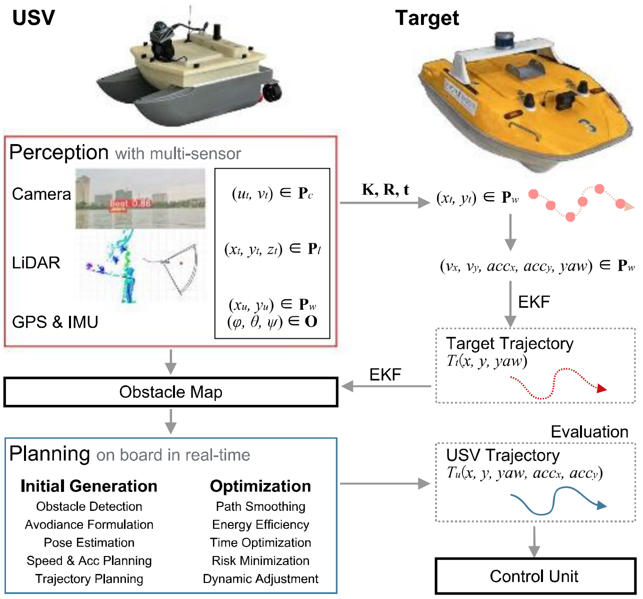


Fig. 2. Framework of the USV tracking system. The system is divided into two main components: perception and planning. The perception module outputs an obstacle map and predicts the target trajectory, obtaining a stable and smooth path via the EKF. The planning module uses this information to plan the USV's trajectory, which is then sent to the control system.

stably maintained in the FOV. Nevertheless, this method treats the heading of the UAV as being directly oriented to the target, which is unrealistic for the underactuated USV. In Wang et al. (2023), rotation trajectory optimization based on obstacle visibility metric and environment complexity metric is proposed to optimize heading motion on the translational trajectory, enhancing the perception of the target and the unknown area in the environment simultaneously. Although this method considers heading trajectory optimization, it designs the translational and heading trajectory optimization separately; that is, the motion dimension (x, y) and ψ are not associated in trajectory optimization, which is also not feasible for the underactuated USV.

3. Perception module and target prediction

The perception module in the USV tracking system achieves precise real-time 3D localization by integrating data from multiple sensors, including cameras, LiDAR, GPS, and IMU. The innovation lies in the dual application of the EKF: first to stabilize 3D coordinates, and second to predict the target's motion trajectory, as illustrated in Fig. 2.

3.1. Multi-sensor integration

The perception system combines RGB images for target recognition and LiDAR data for depth estimation. GPS and IMU data are integrated to estimate the USV's state, allowing for calculating the target's 3D position in the global frame. To ensure the accuracy and stability of these 3D coordinates, an initial application of the EKF is employed. This EKF processes the integrated sensor data to filter out noise, providing a stable and accurate real-time estimate of the target's position.

In the initial development and testing phases, a simulated environment is used to validate the system's algorithms and performance. For instance, the transformation between different coordinate frames, including the use of Rodrigues' rotation formula, is implemented and verified in the simulation:

$$R = I + \sin(\psi)K + (1 - \cos(\psi))K^2, \tag{1}$$

where ψ is the rotation angle, and K is the skew-symmetric matrix derived from the unit rotation vector.

Once validated in simulation, these transformations are applied to real-world scenarios. The target's global position P_t^G is calculated by applying the rotation matrix R_ψ to the target's local position P_t^L and adding the USV's global position P_u^G:

$$P_t^G = R_\psi P_t^L + P_u^G, \tag{2}$$

where ψ represents the yaw angle of the USV in the global coordinate system, P_t^G represents the target's global position, P_t^L denotes the target's local position relative to the USV, and P_u^G indicates the USV's global position. The rotation matrix R_ψ, which accounts for the orientation of the USV in the global coordinate system, is given by:

$$R_\psi = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

The first application of the EKF is used to enhance these global coordinates' accuracy further. This EKF processes the integrated sensor data to filter out noise and provide a stable estimate of the target's position. The state vector $\boldsymbol{x}_k$ at time step k represents the estimated 3D position $\boldsymbol{p}_k$ of the target:

$$\boldsymbol{x}_k = \left[\boldsymbol{p}_k\right]. \tag{4}$$

The EKF updates this estimate based on the incoming sensor measurements:

$$\hat{\boldsymbol{x}}_k = \boldsymbol{A}_k \hat{\boldsymbol{x}}_{k-1} + \boldsymbol{w}_k, \tag{5a}$$

$$\boldsymbol{z}_k = \boldsymbol{H}_k \hat{\boldsymbol{x}}_k + \boldsymbol{\epsilon}_k. \tag{5b}$$

Here, $\boldsymbol{A}_k$ is the state transition matrix, $\boldsymbol{H}_k$ is the measurement matrix, and $\boldsymbol{w}_k$, $\boldsymbol{\epsilon}_k$ represent the process and measurement noise, respectively. This first EKF application ensures that the 3D coordinates are stabilized and accurate.

3.2. Target trajectory prediction

After stabilizing the 3D coordinates, the system employs a second application of the EKF to predict the target's future motion trajectory. This prediction is essential for dynamic tracking, allowing the system to proactively anticipate the target's movements and adjust the USV's path accordingly. The state vector for this EKF includes both the target's position and velocity:

$$\boldsymbol{x}_k = \begin{bmatrix} \boldsymbol{p}_k \\ \boldsymbol{v}_k \end{bmatrix}. \tag{6}$$

Using the current state estimate, the EKF predicts the future position $\hat{\boldsymbol{p}}_{k+1}$ and velocity $\hat{\boldsymbol{v}}_{k+1}$ of the target:

$$\hat{\boldsymbol{x}}_{k+1} = \boldsymbol{A}_k \hat{\boldsymbol{x}}_k + \boldsymbol{B}_k \boldsymbol{u}_k, \tag{7}$$

where $\boldsymbol{u}_k$ represents any control inputs, and $\boldsymbol{B}_k$ is the control input matrix. This second EKF refines the trajectory prediction, minimizing the uncertainty associated with the target's future position, and plays a crucial role in the planning and control modules.

3.3. 3D perception efficiency and robustness

The dual application of EKF within the perception module significantly enhances the system's robustness and accuracy. The first EKF stabilizes the 3D coordinates, ensuring the system has a reliable foundation for subsequent calculations. The second EKF builds on this by predicting the target's future trajectory, enabling the USV to respond effectively to dynamic changes in the environment.

Using simulation environments during the development phase further strengthens the system's robustness, providing a platform for comprehensive testing and validation before real-world deployment. This layered approach allows the system to operate efficiently in real-time, optimizing computational resources while maintaining high accuracy. The system's design is particularly well-suited for deployment

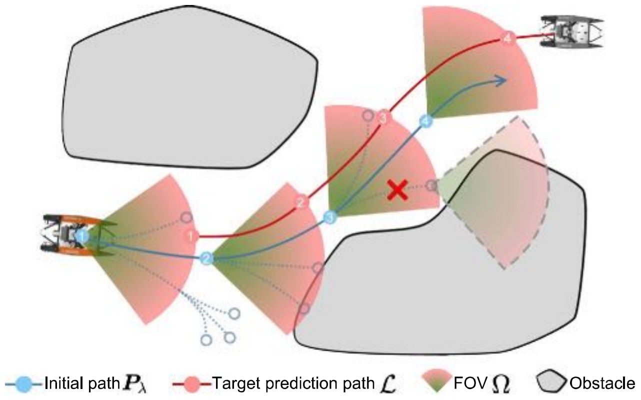

1
2
3
4
Initial path $\boldsymbol{P}_\lambda$
Target prediction path $\mathcal{L}$
FOV Ω
Obstacle

Fig. 3. Illustration of the initial tracking path method. In the path search process, each target point $\boldsymbol{p}_{\iota,i}$ is used as a search goal, and the FOV of the search point $\boldsymbol{p}_{\lambda,i+1}$ extending from $\boldsymbol{p}_{\lambda,i}$ should contain $\boldsymbol{p}_{\iota,i}$ after searching and adjustments, and the expanded node where the target point is obscured by obstacles will be rejected.

in maritime environments, where accurate and responsive tracking is critical.

The combined use of EKF for stabilization and trajectory prediction within the perception module provides a robust and efficient solution for real-time target tracking in dynamic environments.

4. Visibility-aware motion planning

4.1. Initial tracking path searching

Underactuated USVs cannot be propelled horizontally directly, while the FOV's orientation is linked to the USV's heading ψ, implying that the visibility of trackers to the target must be meticulously considered at the path planning stage. Therefore, the multi-goal hybrid A* algorithm is utilized as a front-end path search method to obtain an initial tracking path without occlusion and target loss.

In this section, we rewrite the target prediction trajectory, derived from Section 3.1, as a set of goals for the initial tracking path searching:

$$\mathcal{L} = \left\{ \boldsymbol{p}_{\iota,i} \in \Omega \mid t_i \in [0, T_{\mathcal{L}}], 0 < i \le M \right\}. \tag{8}$$

where $T_{\mathcal{L}}$ denotes the duration of the target prediction trajectory $\mathcal{L}$, $\boldsymbol{p}_{\iota,i}$ represents the position of the target at time t_i, Ω is the set of FOVs during target tracking. Therefore, the path search method is to find a suitable position $\boldsymbol{p}_{\lambda,i}$ for each FOV in the Ω so that it can contain the corresponding target point $\boldsymbol{p}_{\iota,i}$, then we can obtain the initial tracking path $\boldsymbol{P}_\lambda = \{\boldsymbol{p}_{\lambda,1}, \dots, \boldsymbol{p}_{\lambda,M}\}$, as shown in Fig. 3.

4.2. Trajectory representation with flatness

USV system is typically underactuated and subject to non-holonomic constraints, making motion planning quite complex. By utilizing the differential flatness of the USV system, the state and inputs of the system are transformed into the flat output space through the flat transformation determined by the USV's motion equations. This allows the motion planning of the USV to focus solely on solving in the decoupled flat output space, thus avoiding the direct handling of the complex differential constraints of the USV. Additionally, the dimension of the flat output is lower than that of the state-inputs of the USV, facilitating an efficient solution to the planning problem.

In our previous work (Huang et al., 2023b), differential flatness was employed to reduce the trajectory planning dimensions of USV into two independent dimensions x and y, which is a typical form of flat output for a USV system. However, the ability to adjust the FOV flexibly is crucial to ensure the stability and flexibility of the target tracking. It is imperative to incorporate the heading angle as a planning dimension in trajectory planning.

We assume that the USV is a fully actuated vessel and chooses $\boldsymbol{p} = (x, y, \psi)^T$ as the flat output of USV, which simplifies the flatness transformation of all USV system states and inputs. And the typical 3 degrees of freedom (DoF) motion equation of the USV (Fossen, 2011) used in this paper is derived from the following assumptions:

Assumption 1. The USV platform has a homogeneous mass distribution and xz-plane symmetry such that $\boldsymbol{I}_{xz} = \boldsymbol{I}_{yz} = 0$.

Assumption 2. In the target tracking application, the USV does not engage in high-speeds (≤4 m/s), and only considers the influence of linear elements in the damping matrix $\boldsymbol{D}$.

Therefore, the USV system state $\boldsymbol{x} = (x, y, \psi, u, v, r)^T$ and inputs $\boldsymbol{u} = (\tau_u, \tau_v, \tau_r)^T$ can be parameterized by the flat output $\boldsymbol{p}$ and its finite-order derivatives:

$$\boldsymbol{x} = \phi_{\boldsymbol{x}}(\boldsymbol{p}, \dot{\boldsymbol{p}}) = \begin{bmatrix} x \\ y \\ \psi \\ \sin(\psi)\dot{y} + \cos(\psi)\dot{x} \\ \cos(\psi)\dot{y} - \sin(\psi)\dot{x} \\ \dot{\psi} \end{bmatrix}, \tag{9a}$$

$$\boldsymbol{u} = \phi_{\boldsymbol{u}}(\boldsymbol{p}, \dot{\boldsymbol{p}}, \ddot{\boldsymbol{p}}) = \begin{bmatrix} \phi_{\tau_u} \\ \phi_{\tau_v} \\ \phi_{\tau_r} \end{bmatrix}, \tag{9b}$$

where $\phi_{\boldsymbol{x}}$ and $\phi_{\boldsymbol{u}}$ are flatness transformations, uniquely determined by the USV system equation. Therefore, we can quickly obtain the desired trajectory of the USV's state $\boldsymbol{x}$ and inputs $\boldsymbol{u}$ by using the flatness transformation and the planning result in the flat output space. ϕ_{τ_u}, ϕ_{τ_v}, ϕ_{τ_r} are the flat transformation expression for each inputs $\boldsymbol{u}$, described as follows:

$$\begin{aligned} \phi_{\tau_u} &= m_{11}[\cos(\psi)\ddot{x} + \sin(\psi)\ddot{y}] \\ &\quad + [(m_{11} - m_{22})\cos(\psi)\dot{\psi} - X_u \sin(\psi)]\dot{y} \\ &\quad - [(m_{11} - m_{22})\sin(\psi)\dot{\psi} + X_u \cos(\psi)]\dot{x} \\ &\quad - \frac{1}{2}(m_{23} + m_{32})\dot{\psi}^2, \end{aligned} \tag{10a}$$

$$\begin{aligned} \phi_{\tau_v} &= m_{22}[\cos(\psi)\ddot{y} - \sin(\psi)\ddot{x}] + m_{23}\ddot{\psi} - Y_r\dot{\psi} \\ &\quad + [(m_{11} - m_{22})\sin(\psi)\dot{\psi} - Y_v \cos(\psi)]\dot{y} \\ &\quad + [(m_{11} - m_{22})\cos(\psi)\dot{\psi} + Y_v \sin(\psi)]\dot{x}, \end{aligned} \tag{10b}$$

$$\begin{aligned} \phi_{\tau_r} &= m_{32}[\cos(\psi)\ddot{y} - \sin(\psi)\ddot{x}] + m_{33}\ddot{\psi} - N_r\dot{\psi} \\ &\quad + [(m_{11} - m_{22})\sin(\psi)\cos(\psi)](\dot{x}^2 - \dot{y}^2) \\ &\quad + [\frac{1}{2}(m_{23} - m_{32})\sin(\psi)\dot{\psi} - N_v \cos(\psi)]\dot{y} \\ &\quad + [\frac{1}{2}(m_{23} - m_{32})\cos(\psi)\dot{\psi} + N_v \sin(\psi)]\dot{x} \\ &\quad - (m_{11} - m_{22})\cos(2\psi)\dot{x}\dot{y}, \end{aligned} \tag{10c}$$

where $m_\star$ is the element in the inertia matrix $\boldsymbol{M}$. X_u, Y_v, Y_r, N_v and N_r represent the linear damping coefficients in the damping matrix $\boldsymbol{D}$. Considering the platform configuration and application scenarios of the USV, a constraint $\tau_v = \phi_{\tau_v} = 0$ is imposed in the trajectory optimization problem to recover the inherent underactuated vessel dynamics.

In target tracking, the follower must adjust the trajectory from both time and space dimensions according to the predicted target trajectory. The MINCO representation (Wang et al., 2022) conducts spatial–temporal deformation of the flat-output trajectory while maintaining characteristics similar to those of a B-spline. It is represented as an m-dimensional trajectory set composed of M pieces and $K = 2s - 1$ degree polynomial segments:

$$\begin{aligned} \mathfrak{T}_{\mathrm{MINCO}} = \quad & \{\boldsymbol{p}(t) : [0, T_\Sigma] \mapsto \mathbb{R}^m \mid \boldsymbol{c} = \boldsymbol{c}(\boldsymbol{q}, \boldsymbol{T}), \\ & \forall \boldsymbol{q} \in \mathbb{R}^{m(M-1)}, \boldsymbol{T} \in \mathbb{R}^M_{>0}\}, \end{aligned} \tag{11a}$$

$$\boldsymbol{p}_i(t) = \begin{bmatrix} p_{xy,i}(t) \\ \psi_i(t) \end{bmatrix} = \boldsymbol{c}_i^T \boldsymbol{\beta}(t), \forall t \in [0, T_i], \tag{11b}$$

where $\boldsymbol{p}(t)$ is the flat-output trajectory, $\boldsymbol{c} = (\boldsymbol{c}_1^T, \ldots, \boldsymbol{c}_M^T)^T$, $\boldsymbol{c}_i \in \mathbb{R}^{2s\times m}$ is the coefficient matrix of the piece, and $\boldsymbol{\beta}(t) = (1, t, \ldots, t^K)^T$ is the natural basis. $\mathfrak{T}_{\text{MINCO}}$ is uniquely determined by $\boldsymbol{q}$ and $\boldsymbol{T}$, where $\boldsymbol{q} = (q_1, \ldots, q_{M-1})$ denotes the intermediate points vector and $\boldsymbol{T} = (T_1, \ldots, T_M)^T$ is the relative time vector. $T_\Sigma = \sum_{i=1}^{M} T_i$ is the total duration of the trajectory.

To ensure the smoothness of the USV state $\boldsymbol{x}$ and input $\boldsymbol{x}$, we choose $K = 5$ degree polynomial to describe the trajectory.

4.3. Optimization problem formulation

After obtaining the initial tracking path, the constraints can be formed to optimize the trajectory. The nonlinear constraints in the visibility-aware tracking trajectory optimization can be converted to penalty terms using the C^2 penalty function $f(x) = \max\{0, x\}^3$. Therefore, the unconstrained optimization problem for general USV tracking is given by:

$$\begin{aligned} \min_{\boldsymbol{q},\boldsymbol{T}} \mathcal{J} &= w_e \cdot \mathcal{J}_e + w_v \cdot \mathcal{J}_v + w_u \cdot \mathcal{J}_u + w_d \cdot \mathcal{J}_d \\ &\quad + w_o \cdot \mathcal{J}_o + w_s \cdot \mathcal{J}_s + w_t \cdot \mathcal{J}_t \\ &= \mathbf{W} \cdot \left[\mathcal{J}_{TV}, \mathcal{J}_{MF}, \mathcal{J}_{FT}\right], \end{aligned} \tag{12}$$

where $\mathcal{J}_\star$ is the constraint term in the optimization problem, and $w_\star$ represents the weight of each cost function. To clarify the role of each optimization item, we classify all constraints into three categories $\mathcal{J}_{TV}$, $\mathcal{J}_{MF}$ and $\mathcal{J}_{FT}$ and a weight matrix can be defined as $\mathbf{W} = [w_{TV}, w_{MF}, w_{FT}]^T \in \mathbb{R}^{3\times 3}$.

$\mathcal{J}_{TV} = \left[\mathcal{J}_e, \mathcal{J}_v, 0\right]^T$ represents the set of visibility tracking constraints for the target, designed to enable the USV to maintain stable perception of the visual FOV in complex water areas, where $\mathcal{J}_e$ and $\mathcal{J}_v$ represent elastic distance constraint and visibility constraint respectively. $\mathcal{J}_{MF}$ covers the constraints on the motion characteristics of the underactuated USV expressed as $\mathcal{J}_{MF} = \left[\mathcal{J}_u, \mathcal{J}_d, 0\right]^T$, $\mathcal{J}_u$ and $\mathcal{J}_d$ denote underactuated motion constraint and dynamic feasibility constraint respectively. $\mathcal{J}_{FT}$ represents the set of the necessary trajectory optimization constraints, which include the obstacle avoidance cost $\mathcal{J}_o$, smoothness cost $\mathcal{J}_s$, and time adjustment cost $\mathcal{J}_t$.

The magnitude of the cost values computed for the constraints in Eq. (12) does not represent the different priorities of each constraint. This needs to be adjusted through the weight matrix $\mathbf{W}$ so that the optimization problem is guided towards the direction of the set priority of constraints during the optimization process. In Eq. (12), we set the constraints $\mathcal{J}_{FT}$ to have the highest priority, followed by the constraints $\mathcal{J}_{TV}$, and the constraints $\mathcal{J}_{MF}$ with the lowest priority. This is because the constraints $\mathcal{J}_{FT}$ are the necessary constraints for generating a safe trajectory and need to be satisfied first, while the constraints $\mathcal{J}_{MF}$ describe the motion characteristics of the USV and can be assigned the most negligible weight without affecting the generation of an effective tracking trajectory.

4.3.1. Tracking visibility constraint

Maintaining stable target visibility requires keeping the target within the FOV and ensuring obstacles do not obscure it. Additionally, space limitations in specific environments, such as narrow waterways, further complicate FOV adjustments. Therefore, USVs must adjust the tracking distance based on environmental complexity to avoid target occlusion and ensure tracking visibility.

Inspired by the visible region concept in Ji et al. (2022), the feasible area for USV tracking is defined as the visible region $\mathcal{V}$. As illustrated in Fig. 4(a), this region is sector-shaped, extending from each target prediction point and oriented towards the USV. A differentiable function $\mathcal{E}(\boldsymbol{\delta}_i)$ is then designed to represent the complexity of the environment as follows:

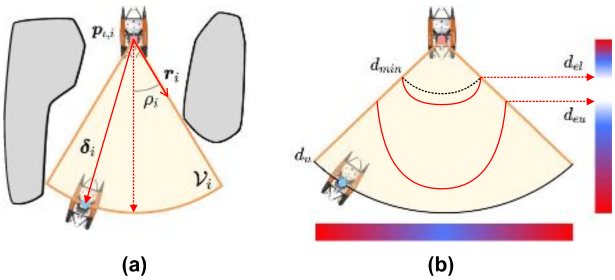


Fig. 4. (a) Definition of the occlusion-free region $\mathcal{V}_i$, (b) illustration of adaptive following distance constraint. The tracking distance interval $[d_{el}, d_{eu}]$ is limited to a small range close to d_{min} at the boundary of both sides of $\mathcal{V}_i$. Red in the color bar indicates high cost, blue indicates low cost, and light color indicates no cost.

$$\mathcal{E}(\boldsymbol{\delta}_i) = \begin{cases} 0, & \boldsymbol{p}_{xy,i} \notin \mathcal{V}_i, \\ \mathcal{G}_{\mathcal{E}}(\boldsymbol{\delta}_i), & \boldsymbol{p}_{xy,i} \in \mathcal{V}_i \wedge \rho_i < \frac{\pi}{2}, \\ 1, & \boldsymbol{p}_{xy,i} \in \mathcal{V}_i \wedge \rho_i \geq \frac{\pi}{2}, \end{cases} \tag{13a}$$

$$\mathcal{G}_{\mathcal{E}}(\boldsymbol{\delta}_i) = 2\left(\frac{\|\boldsymbol{r}_i \times \boldsymbol{\delta}_i\|}{\|\boldsymbol{\delta}_i\|}\right)^2 - \frac{\|\boldsymbol{r}_i \times \boldsymbol{\delta}_i\|^4}{\sin^2(\rho_i) \cdot \|\boldsymbol{\delta}_i\|^4}, \tag{13b}$$

where $\boldsymbol{\delta}_i$ is the vector from the target point $\boldsymbol{p}_{t,i}$ to the tracking position $\boldsymbol{p}_{xy,i}$, $\boldsymbol{r}_i$ is a unit vector representing the boundary of the visible region $\mathcal{V}_i$, and ρ_i denotes the semi-angle of the visible region $\mathcal{V}_i$.

Based on complexity $\mathcal{E}$, a tracking distance field within the visible region $\mathcal{V}$ is constructed. As illustrated in Fig. 4(b), d_u denotes the desired tracking distance, while d_{min} represents the minimum distance, which can be estimated from extreme tracking scenarios. The tracking distance interval $[d_{el}, d_{eu}]$ can vary according to the position of the USV. Thus, the visibility distance cost can be written as:

$$\mathcal{J}_e = \sum_{i=1}^{M} f\left(d_{el,i}^2 - \|\boldsymbol{\delta}_i\|^2\right) + f\left(\|\boldsymbol{\delta}_i\|^2 - d_{eu,i}^2\right), \tag{14a}$$

$$d_{eu,i} = \mathcal{E}(\boldsymbol{\delta}_i)\left(d_u - d_{\min}\right) + d_{\min} + d_\epsilon, \tag{14b}$$

$$d_{el,i} = \gamma \cdot \mathcal{E}(\boldsymbol{\delta}_i)\left(d_u - d_{\min}\right) + d_{\min}, \tag{14c}$$

where d_ϵ is tracking distance tolerance and γ is a constant less than 1.

Additionally, the USV must align the axis of the sensor's FOV directly towards the target to maximize tracking perception. Therefore, the cost function for visibility heading is designed as:

$$\mathcal{J}_v = \sum_{i=1}^{M} f\left(\cos(\rho_\epsilon) + \frac{e(\psi_i) \cdot \boldsymbol{\delta}_i}{\|\boldsymbol{\delta}_i\|}\right), \tag{15}$$

where ρ_ϵ is a angle clearance and $e(\psi_i) = \left[\cos(\psi_i), \sin(\psi_i)\right]^T$.

4.3.2. Motion feasibility constraint

To guarantee the trajectory includes the underactuated characteristics of USV, the constraint $\tau_v = 0$ needs to be satisfied. By relaxing the equality constraint into inequality constraint $\|\tau_v\| \leq \tau_\epsilon$, the underactuated model cost can be constructed as:

$$\mathcal{J}_u = \sum_{i=1}^{M} \int_0^{T_i} f\left(\phi_{\tau_v}\left(\boldsymbol{p}_i, \dot{\boldsymbol{p}}_i, \ddot{\boldsymbol{p}}_i, t\right)^2 - \tau_\epsilon^2\right) \mathrm{d}t, \tag{16}$$

where τ_ϵ is a tolerable upper limit of lateral force.

The dynamic feasibility cost is determined by the velocity and acceleration of the trajectory $\boldsymbol{p}(t)$. In this work, trajectory curvature constraint $\mathcal{J}_\sigma$ needs to be added to satisfy the characteristics of USV underactuated:

$$\mathcal{J}_d = \mathcal{J}_{d,v} + \mathcal{J}_{d,a} + \mathcal{J}_\sigma, \tag{17a}$$

$$\mathcal{J}_\sigma = \sum_{i=1}^{M} \int_0^{T_i} f\left(\left[\frac{\dot{\psi}_i(t)}{\left\|\dot{p}_{xy,i}(t)\right\|}\right]^2 - \sigma^2\right) \mathrm{d}t, \tag{17b}$$

where σ is a tunable maximum curvature of trajectory. We refer to Zhou et al. (2020) to define velocity feasibility cost $\mathcal{J}_{d,v}$ and acceleration feasibility cost $\mathcal{J}_{d,a}$.

4.3.3. Fundamental trajectory constraint

The tracking USV primarily needs to focus on the environment surrounding the predicted target trajectory, which typically involves river courses and sparsely obstructed waters. Thus, the feasible area $\mathcal{F}$ for the USV is approximated by a combination of $M_{\mathcal{P}}$ closed and convex polyhedral elements:

$$\mathcal{F} \simeq \bigcup_{i=1}^{M_{\mathcal{P}}} \mathcal{P}_i, \mathcal{P}_i = \left\{p_h \in \mathbb{R}^2 \mid \mathbf{A}_i \cdot p_h \leq b_i\right\}. \tag{18}$$

The polyhedron generation method is referenced from Ji et al. (2022). Define the set of distances from $\boldsymbol{p}_i(t)$ to each boundary of the related polyhedral $\mathcal{P}_{p_i} \subset \mathcal{F}$ as $\mathcal{D}(\boldsymbol{p}_{xy,i}(t)) = [\mathcal{D}_1, \ldots, \mathcal{D}_H]^T$. The obstacle avoidance penalty is then obtained by calculating the integral of the violation of safety conditions:

$$\mathcal{J}_o = \sum_{i=1}^{M} \int_0^{T_i} \sum_{j=1}^{H} f\left(d_{\text{safe}} - \mathcal{D}_j\right) \mathrm{d}t, \tag{19}$$

where obstacle avoidance safety threshold d_{safe} is set according to the USV outline.

The smoothness cost $\mathcal{J}_s$ minimizes the third derivative of the trajectory $\boldsymbol{p}^{(3)}$ to ensure its smoothness. The time adjustment cost $\mathcal{J}_t$ optimizes the time duration, ensuring that the time slack $T_\Sigma > T_\mathcal{L}$ satisfies dynamic feasibility constraints when the target moves faster. For a detailed description of these two constraints, please refer to Wang et al. (2022).

4.3.4. Numerical optimization

The cost function $\mathcal{J}_\star(\boldsymbol{c}, \boldsymbol{T})$ is defined as a continuous time form with penalty function $f(x)$, which is intractable in numerical optimization. A simple method is to approximate the weighted sum of the sampled constraint function. Solving the optimization problem Eq. (12) needs $\partial\mathcal{J}_\star/\partial\boldsymbol{c}$ and $\partial\mathcal{J}_\star/\partial\boldsymbol{T}$ which can be easily obtained through the chain rule. According to the proof in Wang et al. (2022), any second-order continuous cost function $\mathcal{J}_\star(\boldsymbol{c}, \boldsymbol{T})$ can be converted to be represented by $\boldsymbol{q}$ and $\boldsymbol{T}$ with linear time and space complexity. Thus $\partial\mathcal{J}_\star/\partial\boldsymbol{c}$ and $\partial\mathcal{J}_\star/\partial\boldsymbol{T}$ can be converted to a gradient representation of $\boldsymbol{q}$ and $\boldsymbol{T}$ in the form of linear complexity to guide the numerical optimization process. Finally, L-BFGS solves the unconstrained numerical optimization problem (Zhou et al., 2020).

5. Experimental results and analysis

In this section, dedicated tests are conducted to verify the effectiveness of the proposed methods in perception and planning. Additionally, real-world experiments demonstrate the feasibility of our USV tracking system.

5.1. Simulation and implementation details

In the simulation setup, 3D models of the USV and target objects are integrated into a virtual aquatic environment. The Robot Operating System (ROS) serves as the data exchange and control middleware, providing a unified framework for the interaction between perception and planning modules. The simulator is a controlled testbed designed to validate the proposed algorithms under various conditions, thereby assessing their generalizability and robustness before real-world implementation.

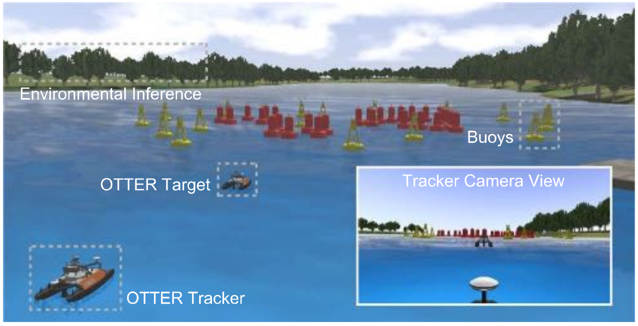


Fig. 5. Simulation environment based on VRX (Bingham et al., 2019). The USV, target, and obstacles are placed in the scene.

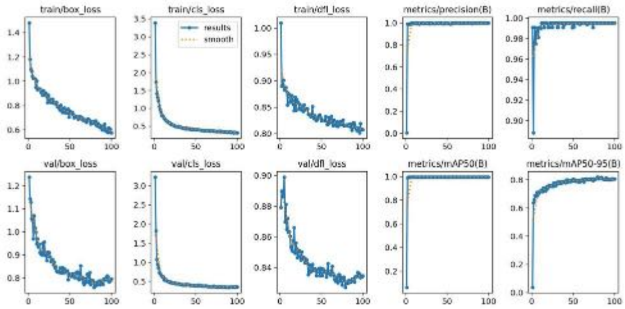


Fig. 6. Loss and accuracy progression during YOLO training.

The Virtual RobotX simulator (VRX) is used to perform simulation experiments. VRX (Bingham et al., 2019), based on Gazebo, can simulate the behavior of USVs in complex environments with wave and buoyancy conditions. Moreover, a small catamaran OTTER, equipped with LiDAR, a camera, and GPS, is provided in VRX, as shown in Fig. 5. To minimize the differences between simulation and real-world experiments, the self-designed autopilot is also used in the simulation experiments.

5.2. Target tracking and trajectory prediction in simulation

Object detection in images serves as the basis for solving perception problems. Real tracking targets are transformed into 3D models and imported into a simulation system, aligning with existing dynamic models. This process focuses on tracking rather than the coherence of the object's motion state with the dynamic models, allowing for data collection in a simulated environment and validation in real-world settings. This discussion extends to the 3D model reconstruction and data synthesis of the USV platform, enhancing the database for more accurate and diverse target-tracking simulations.

The Fig. 6 illustrates the key metrics from the object detection model's training, including training loss, validation loss, and box loss. These metrics indicate that the training has converged effectively. After applying the pruning technique, the model achieved over 98.5% accuracy at mAP@50, demonstrating its high object detection accuracy.

In Fig. 7, it is observed that as the distance increases, the number of point clouds detected by the 32-line LiDAR on the target decreases. Beyond 15 m, the number of target points falls below 10, predominantly due to single-beam reflections on the object. At distances less than 3 m, sensor visibility is obstructed by the agent's vessel, leading to only capturing partial images and point clouds of the target, which can result in misidentification. Therefore, we have set the tracking distance within

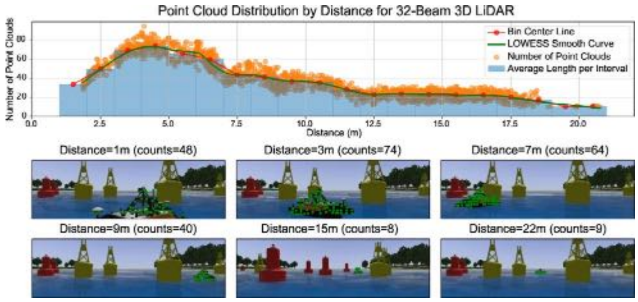


Fig. 7. Point cloud density versus distance and visualizations. The figure illustrates the inverse relationship between distance and point cloud density on a target, with a fitted curve showing the decreasing trend. The subplots of point cloud visualizations at varying distances demonstrate the reduction in density.

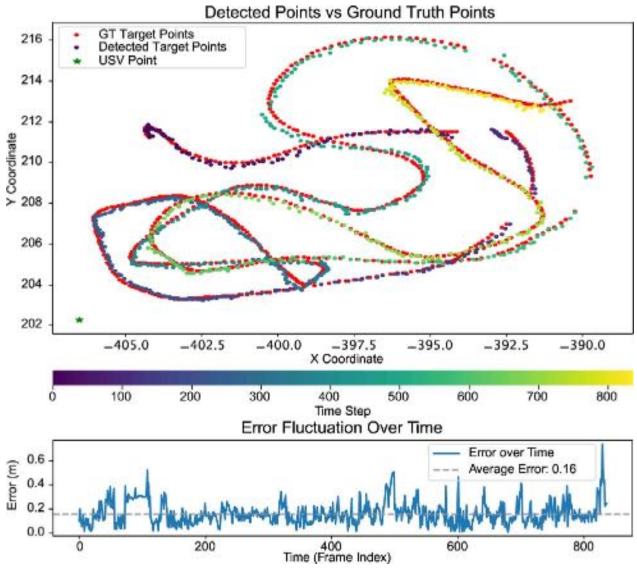


Fig. 8. Object detection result in the simulation system. Detected and ground truth positions are presented and compared in detail.

the range where more target points are observed, specifically between 5 and 10 m. This range is deemed acceptable and corresponds to the area where the 32-line LiDAR imaging is relatively complete.

We employ voxel sampling in three-dimensional space with a size of 0.1 m. Post-clustering, sparse noise points are filtered out. The 2D coordinates of targets detected in images are then mapped into 3D space, forming a ray to identify targets along it. The mean of all points in a cluster is calculated to determine the target's position in the BEV, which is essential for predicting the trajectory over time. The precision of target detection, as shown in Fig. 8, has an error margin of 16 cm, significantly less than the size of the target boats in the simulation environment.

Within the camera's FOV, global coordinate detection of the target is performed, as shown in Fig. 8. During the target's rotation, local errors arise due to changes in posture. Additionally, as the target's velocity varies, the number of LiDAR beams striking the target changes dynamically with distance, leading to shifts in the centroid after point cloud clustering. These factors contribute to short-range fluctuations. The average error is 16 cm, significantly smaller than the dimensions of the target object, and falls within an acceptable error range. Post-processing with an EKF for trajectory prediction provides valuable information for subsequent planning tasks.

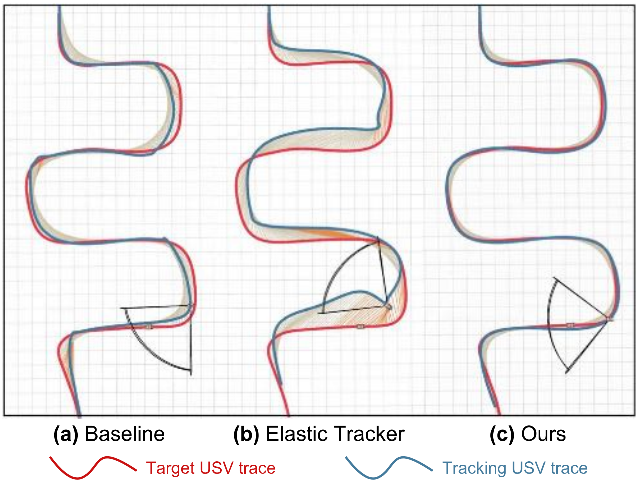


Fig. 9. Comparative experimental results in open water area. The gray lines between two traces indicate normal tracking, and the orange lines between two traces indicate target escaping from the FOV area.

5.3. Trajectory generation for USV tracking in simulation

The proposed method is compared with the baseline method (Huang et al., 2023b) and the elastic tracker (Ji et al., 2022) in simulation to verify the superiority of the proposed method in USV tracking.

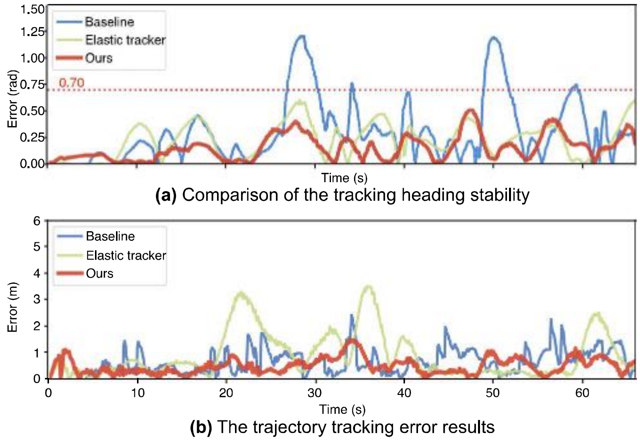
In this part of the experiment, it is assumed that motion information of the target has been obtained and a uniform motion model to predict the future trajectory of the target and set the maximum target recognition distance of USV to 15 m according to the actual platform conditions. The target's maximum speed and angular velocity are limited to 2.8 m/s and 0.9 rad/s, respectively. In comparison, the maximum speed and maximum angular velocity of the USV are limited to 3.0 m/s and 1.5 rad/s, respectively. The weights matrix $\mathbf{W}$ of the constraints has been well-tuned. Specifically, we first set the values in the weight matrix $\mathbf{W}$ to 1.0 and record the cost of each constraint through different optimization calculations. Then, we adjust the weight values according to the costs and the set priorities. Through numerous experimental tests and weight adjustments in the simulation scenario, we ultimately obtained weight parameters that satisfy the desired tracking behavior. In physical experiments, we make minor adjustments based on the weight parameters of the simulation environment according to the tracking performance. The Table 1 summarizes the critical parameters used throughout this work.

5.3.1. Tracking motion feasibility test

The target ship sails along an aggressive path in the open water area, and the desired tracking distance d_u is set as 7 m. In Fig. 9, the tracking results of baseline method and elastic tracker are twisted and show appearance of target loss when turning. Specifically, the baseline method does not consider the target's visibility. It simply tracks the position of the current target, making the tracking process discontinuous and making it hard to maintain a stable perspective and tracking distance. The elastic tracker does not consider the heading dimensions and uses the direction towards the target as the heading goal separately, resulting in trajectory tracking errors that cannot be ignored when using the same PID controller in the experiment, as shown in Fig. 10. In contrast, our method jointly optimizes position and heading, keeping the target in a moderate area of FOV shown in Fig. 10. Moreover, as shown in Fig. 10, our method incorporates USV motion constraint so that the USV using only a PID controller can easily track the planning results with a small trajectory tracking error.

Table 1
Parameters related to trajectory optimization.

Parameters	Simulation	Physical experiment
w_{TV}	$[1.0, 30.0, 0]$	$[1.0, 60.0, 0]$
w_{MF}	$[0.005, 1.0, 0]$	$[0.005, 1.0, 0]$
w_{FT}	$[10.0, 1, 0, 60.0]$	$[10, 1.0, 60.0]$
m_{11}	35.0	9.8
m_{22}	33.0	9.1
m_{23}	3.0	1.3
m_{32}	4.0	2.7
m_{33}	16.0	4.1
X_u	20.0	6.8
Y_v	20.0	7.4
Y_r	3.0	1.7
N_v	10.0	3.4
N_e	20.0	2.1
d_ϵ	3.0	1.5
γ	0.4	0.4
ρ_ϵ	0.8	0.8
τ_ε	3.5	1.0
σ	0.5	0.35
d_{safe}	0.8	0.4



(a) Comparison of the tracking heading stability

(b) The trajectory tracking error results

Fig. 10. Comparative results of tracking heading stability and trajectory tracking error.

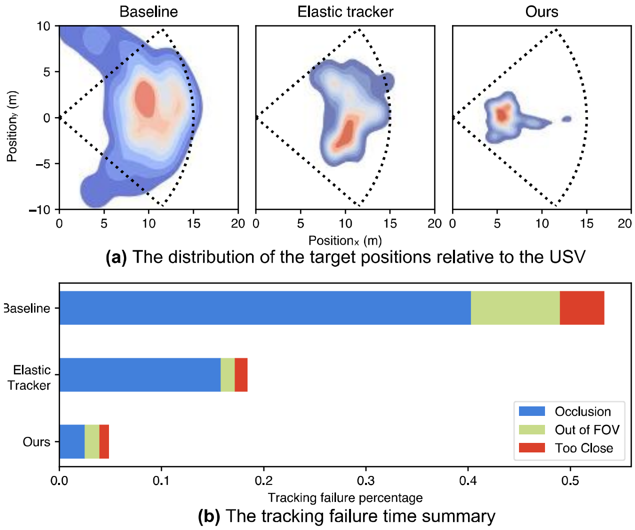
Consequently, the motion feasibility constraints and joint optimization of heading and position proposed in our method can effectively enhance the motion stability of USV in target tracking, which is conducive to maintaining a robust USV perception perspective.

5.3.2. Tracking visibility test

To prove the target visibility of the proposed method in complex water environments, all methods are compared in a narrow waterway environment using buoys as shown in Fig. 5 and set the minimum tracking distance $d_{\min}$ to 3 m based on the physical characteristics of the USV while setting the desired tracking distance d_u as 10 m.

The quantitative analysis of the failure time is depicted in Fig. 11. It can be seen that our proposed method has significantly less tracking failure time than the other two methods. In addition, the target position is recorded in the FOV of the USV as a distribution map, as shown in Fig. 11, and it can be seen that our method can maintain the target at a suitable position within the FOV, making it more robust for target tracking in complex water environments.

In Fig. 12(a), the baseline method does not consider the future motion of the target. It lakes a practical tracking visibility constraint, making it susceptible to losing the target and more occlusions in complex waters. As shown in Fig. 12(b), since the elastic tracker only avoids occlusion by adjusting the tracking position, which does not have enough space to adjust in narrow waterways, the target would be occluded at the corners of narrow waterways. Moreover, the fixed tracking distance makes it hard to actively satisfy the visibility requirements of tracking in different environments. It is also worth noting that due to the significant errors in the trajectory execution process, the USV collided with the buoy in the black box marked in Fig. 12(b), indicating that the elastic tracker cannot guarantee the safety of the USV in the target tracking process. In Fig. 12(c), we present the tracking results of our method, which associates the complexity of the environment with the target tracking distance. Therefore, in complex environments, we can flexibly adjust the tracking distance to maintain the target's visibility.



(a) The distribution of the target positions relative to the USV

(b) The tracking failure time summary

Fig. 11. (a) Distribution of target positions relative to the USV on the x-y plane. Color regions indicate the frequency of target appearances, with red representing high frequency and blue representing low frequency, and (b) failure times for three cases in complex water environments.

5.4. USV agent platform for physical experiments

Fig. 13 shows our self-made mini USV for tracking the target USV. The size of the mini USV platform is 0.67 m × 0.35 m × 0.26 m, and the target USV measures 1.3 m × 0.75 m × 0.45 m. The USV platform can achieve a maximum speed of 2.5 m/s and a maximum angular velocity of 1.0 rad/s. The mini USV is equipped with off-the-shelf sensors including a monocular camera (FOV = 85°×63°) for target recognition, a Livox Mid-360 LiDAR for environmental perception, a localization module comprising an IMU and RTK GPS, and an Intel NUC microcomputer (CPU: i7-1165G7, RAM: 16 GB) for running all algorithms and hardware devices. The tracking system employs a sensor fusion strategy, combining data from the camera, LiDAR, GPS, and IMU to enhance object detection and trajectory prediction in aquatic environments. The architecture facilitate a seamless transition of existing algorithms onto the real-world USV platform, validating the system's operability and effectiveness beyond simulation.

Multi-sensor integration enhances target positioning and trajectory estimation by leveraging the strengths of various data modalities. This approach employs RGB images for target recognition, matched with point clouds to obtain depth information, aligning with current research trends in multi-sensor fusion. The primary goal is to determine the target's relative position and the USV.

5.5. Experimental analysis

The tracking mission's real-world experiment is illustrated in Fig. 14, where the USV and the target operate in an open water

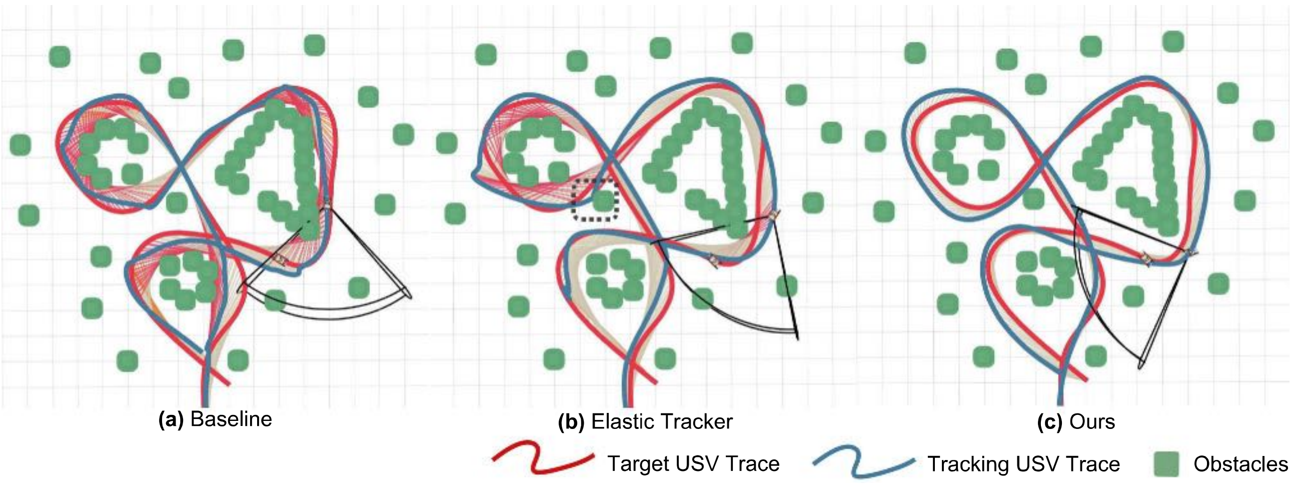


Fig. 12. Comparative experimental results in complex water areas. The red lines between the two traces indicate target occlusion, and the black dashed box marks the location where the USV collided with an obstacle.

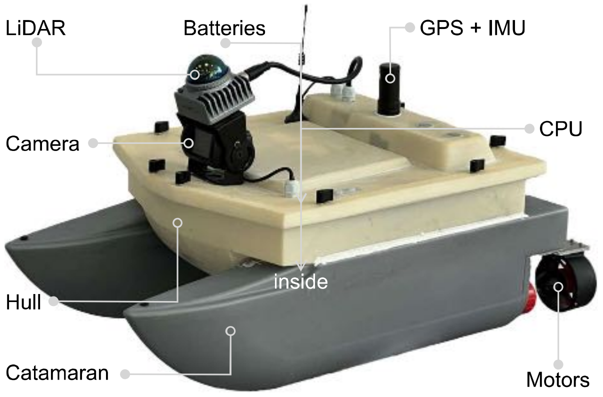


Fig. 13. Mini USV platform used in physical experiment.

environment. The target is a remotely controlled USV that does not broadcast its location, adding complexity to the tracking task.

The USV-tracker system, equipped with multiple sensors and an onboard NUC, can run all algorithms in real time. The target perception module, which includes sensor drivers (cameras, GPS, IMU, LiDAR), environment mapping, object detection, and multi-sensor fusion filtering, utilizes approximately 4 CPU cores and less than 1.5 GB of memory. Specifically, sensor drivers utilize about 8% of CPU and 160 MB of memory; environment mapping occupies 3% of CPU and 130 MB of memory, object detection uses 12% of CPU and 530 MB of memory, while multi-sensor fusion filtering consumes 2% of CPU and 150 MB of memory. The perception algorithm maintains a detection output frequency of 12 Hz, ensuring responsiveness in dynamic conditions.

The trajectory planning module, comprising initial path searching, trajectory optimization, low-level control, and FCU driving, operates efficiently within the system's computational limits. Initial path searching consumes 9% of CPU and 225 MB of memory, trajectory optimization occupies 6% of CPU and 110 MB of memory, low-level control uses 1% of CPU and 72 MB of memory, and FCU driving requires less than 1% of CPU and 65 MB of memory. The computation time for trajectory planning generally remains below 25 ms, meeting the requirements for real-time USV tracking.

The target USV followed a complex path during the experiment, while a 4-m human-crewed boat acted as an obstacle. Despite the absence of communication and dynamic object interference, the tracking USV maintained stable performance, demonstrating the system's robustness and accuracy in real-world conditions.

6. Conclusion

In this paper, we examined the factors influencing the efficiency and visibility of USVs in the tracking process. We propose a novel USV tracking system that ensures perceptual robustness and tracking concealment despite sensor limitations and environmental barriers. This system is built on a multi-sensor fusion perception and visibility-aware trajectory planner. Simulation and real-world experiments were conducted to validate the efficiency and robustness of the proposed USV tracking system. In future work, we plan to replace the experimental equipment with professional-grade instruments and extend the USV tracker to the field of multiple USVs. We aim to investigate the use of multiple USVs for adaptive formation or tracking multiple targets in monitored sea areas.

CRediT authorship contribution statement

Tao Huang: Writing – original draft, Visualization, Validation, Formal analysis. **Yiheng Xue:** Writing – original draft, Visualization, Methodology, Formal analysis, Data curation, Conceptualization. **Zhenfeng Xue:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Funding acquisition, Data curation, Conceptualization. **Zheng Zhang:** Validation, Formal analysis, Data curation. **Zhonghua Miao:** Writing – review & editing, Supervision. **Yong Liu:** Writing – review & editing, Supervision.

Declaration of competing interest

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

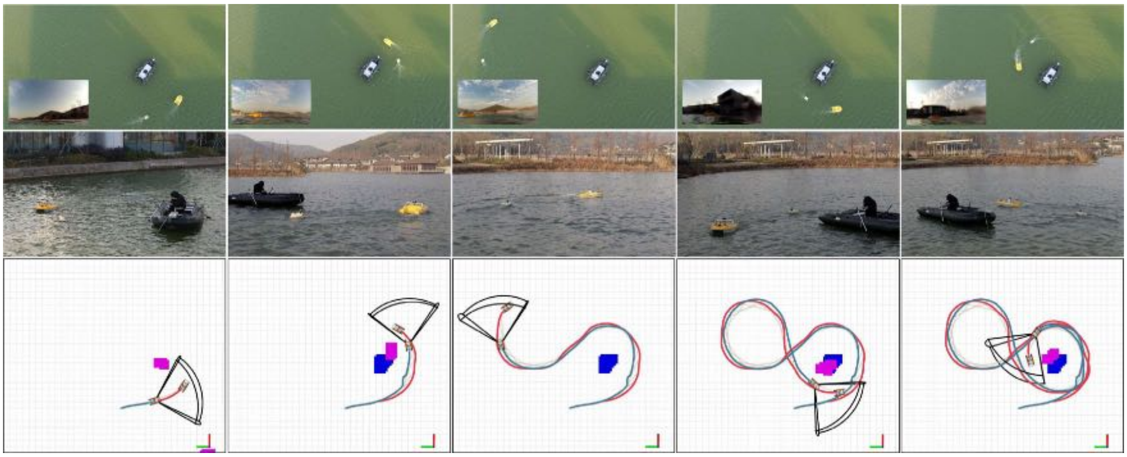


Fig. 14. Chronological extraction of five key moments from the experiment. **Top:** images from a top-down view and the front-end camera of the USV. **Middle:** close-side view display of the USV tracking scenario. **Bottom:** visualization of tracking visibility and trace.

References

Abd Rahman, N.A., Sahari, K.S.M., Hamid, N.A., 2022. An autonomous clutter inspection approach for radiological survey using mobile robot. IEEE Trans. Autom. Sci. Eng. 20 (2), 1212–1225.

Agrawal, P., Dolan, J.M., 2015. COLREGS-compliant target following for an unmanned surface vehicle in dynamic environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, pp. 1065–1070.

Bibuli, M., Caccia, M., Lapierre, L., Bruzzone, G., 2012. Guidance of unmanned surface vehicles: Experiments in vehicle following. IEEE Robot. Autom. Mag. 19 (3), 92–102.

Bingham, B., Agüero, C., McCarrin, M., Klamo, J., Malia, J., Allen, K., Lum, T., Rawson, M., Waqar, R., 2019. Toward maritime robotic simulation in gazebo. In: Oceans. IEEE, pp. 1–10.

Breivik, M., Hovstein, V.E., Fossen, T.I., 2008. Straight-line target tracking for unmanned surface vehicles.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-end object detection with transformers. In: European Conference on Computer Vision. Springer, pp. 213–229.

Chen, Z., Guo, Y., Wang, Q., Chen, Y., 2022. Research on target tracking system of unmanned surface vehicle based on hierarchical control strategy. In: Chinese Control Conference. IEEE, pp. 3651–3655.

Chen, Z., Huang, T., Xue, Z., Zhu, Z., Xu, J., Liu, Y., 2021. A novel unmanned surface vehicle with 2d-3d fused perception and obstacle avoidance module. In: IEEE International Conference on Robotics and Biomimetics. IEEE, pp. 1804–1809.

Chen, X., Ma, H., Wan, J., Li, B., Xia, T., 2017. Multi-view 3d object detection network for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1907–1915.

Dissanayaka, D., Wanasinghe, T.R., De Silva, O., Jayasiri, A., Mann, G.K., 2023. Review of navigation methods for UAV-based parcel delivery. IEEE Trans. Autom. Sci. Eng..

Fossen, T.I., 2011. Handbook of Marine Craft Hydrodynamics and Motion Control. John Wiley & Sons.

Girshick, R., 2015. Fast r-cnn. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1440–1448.

Han, Z., Zhang, R., Pan, N., Xu, C., Gao, F., 2021. Fast-tracker: A robust aerial system for tracking agile target in cluttered environments. In: IEEE International Conference on Robotics and Automation. IEEE, pp. 328–334.

Huang, T., Chen, Z., Gao, W., Xue, Z., Liu, Y., 2023a. A USV-UAV cooperative trajectory planning algorithm with hull dynamic constraints. Sensors 23 (4), 1845.

Huang, T., Xue, Z., Chen, Z., Liu, Y., 2023b. Efficient trajectory planning and control for USV with vessel dynamics and differential flatness. In: IEEE/ASME International Conference on Advanced Intelligent Mechatronics. IEEE, pp. 1273–1280.

Ji, J., Pan, N., Xu, C., Gao, F., 2022. Elastic tracker: A spatio-temporal trajectory planner for flexible aerial tracking. In: IEEE International Conference on Robotics and Automation. IEEE, pp. 47–53.

Ku, J., Mozifian, M., Lee, J., Harakeh, A., Waslander, S.L., 2018. Joint 3d proposal generation and object detection from view aggregation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, pp. 1–8.

Li, Y., Ge, Z., Yu, G., Yang, J., Wang, Z., Shi, Y., Sun, J., Li, Z., 2023. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 37, pp. 1477–1485.

Lin, B., Xie, W., Shi, Y., Du, B., Zhang, C., Zhang, W., 2023. Robust target interception strategy for a USV with experimental validation. IEEE Robot. Autom. Lett..

Liu, Z., Tang, H., Amini, A., Yang, X., Mao, H., Rus, D.L., Han, S., 2023. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In: IEEE International Conference on Robotics and Automation. IEEE, pp. 2774–2781.

Liu, Z., Zhang, Z., Cao, Y., Hu, H., Tong, X., 2021. Group-free 3d object detection via transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2949–2958.

Muchiri, G., Kimathi, S., 2022. A review of applications and potential applications of UAV. In: Proceedings of the Sustainable Research and Innovation Conference. pp. 280–283.

Philion, J., Fidler, S., 2020. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: European Conference on Computer Vision. Springer, pp. 194–210.

Qi, C.R., Litany, O., He, K., Guibas, L.J., 2019. Deep hough voting for 3d object detection in point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9277–9286.

Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J., 2018. Frustum pointnets for 3d object detection from rgb-d data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 918–927.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 652–660.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Adv. Neural Inf. Process. Syst. 30.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 779–788.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Adv. Neural Inf. Process. Syst. 28.

Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H., 2020. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10529–10538.

Shi, S., Wang, X., Li, H., 2019. Pointrcnn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 770–779.

Sinisterra, A.J., Dhanak, M.R., Von Ellenrieder, K., 2017. Stereovision-based target tracking system for USV operations. Ocean Eng. 133, 197–214.

Švec, P., Thakur, A., Raboin, E., Shah, B.C., Gupta, S.K., 2014. Target following with motion prediction for unmanned surface vehicle operating in cluttered environments. Auton. Robots 36, 383–405.

Szrek, J., Zimroz, R., Wodecki, J., Michalak, A., Góralczyk, M., Worsa-Kozak, M., 2020. Application of the infrared thermography and unmanned ground vehicle for rescue action support in underground mine—The amicos project. Remote Sens. 13 (1), 69.

Wang, H., Zhang, X., Liu, Y., Zhang, X., Zhuang, Y., 2023. SVPTO: Safe visibility-guided perception-aware trajectory optimization for aerial tracking. IEEE Trans. Ind. Electron..

Wang, Z., Zhou, X., Xu, C., Gao, F., 2022. Geometrically constrained trajectory optimization for multicopters. IEEE Trans. Robot. 38 (5), 3259–3278.

Yu, Y., Guo, C., Yu, H., 2019. Finite-time PLOS-based integral sliding-mode adaptive neural path following for unmanned surface vessels with unknown dynamics and disturbances. IEEE Trans. Autom. Sci. Eng. 16 (4), 1500–1511.

Zhou, B., Pan, J., Gao, F., Shen, S., 2021. Raptor: Robust and perception-aware trajectory replanning for quadrotor fast flight. IEEE Trans. Robot. 37 (6), 1992–2009.

Zhou, X., Wang, Z., Ye, H., Xu, C., Gao, F., 2020. Ego-planner: An esdf-free gradient-based local planner for quadrotors. IEEE Robot. Autom. Lett. 6 (2), 478–485.