sensors

MDPI

Article

Optimized Dynamic Collision Avoidance Algorithm for USV Path Planning

Hongyang Zhu [1] and Yi Ding [2,*]

[1] College of Mathematics and Computer, Guangdong Ocean University, Zhanjiang 524091, China
[2] Maritime College, Guangdong Ocean University, Zhanjiang 524091, China
* Correspondence: dingyi@gdou.edu.cn

Abstract: Ship collision avoidance is a complex process that is influenced by numerous factors. In this study, we propose a novel method called the Optimal Collision Avoidance Point (OCAP) for unmanned surface vehicles (USVs) to determine when to take appropriate actions to avoid collisions. The approach combines a model that accounts for the two degrees of freedom in USV dynamics with a velocity obstacle method for obstacle detection and avoidance. The method calculates the change in the USV's navigation state based on the critical condition of collision avoidance. First, the coordinates of the optimal collision avoidance point in the current ship encounter state are calculated based on the relative velocities and kinematic parameters of the USV and obstacles. Then, the increments of the vessel's linear velocity and heading angle that can reach the optimal collision avoidance point are set as a constraint for dynamic window sampling. Finally, the algorithm evaluates the probabilities of collision hazards for trajectories that satisfy the critical condition and uses the resulting collision avoidance probability value as a criterion for course assessment. The resulting collision avoidance algorithm is optimized for USV maneuverability and is capable of handling multiple moving obstacles in real-time. Experimental results show that the OCAP algorithm has higher and more robust path-finding efficiency than the other two algorithms when the dynamic obstacle density is higher.

Keywords: collision avoidance; velocity obstacle method; trajectory optimization; optimal collision avoidance point

Citation: Zhu, H.; Ding, Y. Optimized Dynamic Collision Avoidance Algorithm for USV Path Planning. *Sensors* **2023**, *23*, 4567. https://doi.org/10.3390/s23094567

Academic Editor: Antonio M. Pascoal

Received: 31 March 2023
Revised: 29 April 2023
Accepted: 4 May 2023
Published: 8 May 2023

1. Introduction

Ship collision is an imperative task for navigation safety at sea [1]. Unmanned surface vehicles (USVs) have gained significant attention in recent years due to their potential for various applications such as oceanographic research, environmental monitoring, and maritime security [2]. However, the increasing use of USVs also raises concerns about the safety of navigation, especially when operating in crowded environments [3]. Collision avoidance is a critical issue that needs to be addressed to ensure safe and efficient navigation of USVs. A USV's collision avoidance algorithm can be considered a local path-planning algorithm. This paper focuses on local path-planning methods. Many local path-planning algorithms have been reported for collision avoidance against static and dynamic obstacles [4,5]. By now, many obstacle avoidance algorithms have been proposed by international scholars, all of which rely more or less on global path planning and mapping, such as a bug algorithm [6], a vector field histogram method [7], and an artificial potential field method [8]. Many improved heuristic algorithms have also been studied for local path planning. For instance, a hybrid adaptive path-planning scheme based on global path planning and local dynamic collision avoidance for unmanned surface vehicles under complex marine environments was proposed in [9]. This method systematically considers the impact of waves and currents on the navigation of USVs. In recent years, some scholars have emphasized the dynamic collision avoidance of ships by incorporating methods such as reinforcement learning and

COLREGS [10]. Based on literature statistics, 56% of the collisions at sea are caused by the violation of COLREGS by ships [11]. However, data are difficult to collect in real time, and it is difficult to show a model with mathematical formulas; incorporating regulations in collision prevention algorithms is still a challenge [12]. Some scholars have tried to design a ship navigation safety domain to solve the ship collision problem [13–15]. However, most of them only consider static obstacles or semi-dynamic obstacles that do not change course [16,17], a highly ideal motion model is used in collision avoidance [18], or the balance between efficiency and effectiveness is ignored [19].

Some other researchers have evaluated collision avoidance trajectories generated from the perspective of risk assessment [20] using multiple parameters such as navigation risk [21], navigation smoothness, and other metrics. For instance, a review [22] described collision risk assessment, but it neglected techniques for conflict resolution. The most frequently used distance parameter for conflict detection and obstacle avoidance is the distance to the closest point of approach d_{cpa}; the time to the closest point of approach t_{cpa} is often used with it. It has been proposed that various techniques can be developed to overcome the limitations of d_{cpa} and t_{cpa} alone for collision avoidance [23]. In [24,25], the authors discussed ASV developments in depth, while conflict detection and obstacle avoidance received a lesser amount of attention. Only a few studies related to reacting to collision avoidance for unmanned ships were included in [26].

Comparing other obstacle avoidance algorithms, Fox et al. reported a dynamic window approach (DWA) [27], which has become a popular academic research method in recent years. It is mainly used for navigation and obstacle avoidance in a dynamic environment. Avoiding unpredictable obstacles can better solve the DWA [28]. DWA is widely used in dynamic obstacle avoidance path optimization of UAVs, robots, and USVs [29–31]. Dobrevski reported local path planning based on DWA and deep reinforcement learning to improve path optimization [32]. Liu developed a global dynamic path-planning fusion algorithm combining the jump-A* algorithm and DWA [33]. In addition, several useful local path-planning methods based on DWA have been reported [34]. However, DWA generates path candidates by assuming constant velocities for a certain period of time. Due to the small distances between obstacles and USVs, unexpected collisions often occur during encounters, which makes it challenging to fulfill the safety requirements of USVs. However, DWA generates path candidates by assuming constant velocities for a certain period, making it easy for it to fall into local optima [35]. In the path evaluation stage, it relies heavily on the settings of the parameter value ranges. For example, when the distance between an obstacle and a USV is small, accidental collisions often occur during the encounter, which is a challenge to meeting the safety requirements of USVs. In addition, the increased complexity of the application scenarios and environments of unmanned devices make standard DWA unable to solve complex path-planning problems. Traditional DWA focuses on path generation at each step of the planning process but ignores that obstacles are also intelligent agents that generate abrupt behavior [36].

In this study, we present a novel approach to collision avoidance for USVs. While existing methods mainly rely on static obstacle maps or simple heuristics, our approach combines a two-degree-of-freedom model for USV dynamics with a velocity obstacle method for obstacle detection and avoidance. This approach allows for real-time adaptation to dynamic and complex environments, making it particularly suitable for USVs operating in areas with high traffic density or unpredictable obstacles. The resulting collision avoidance algorithm is optimized for USV maneuverability and is capable of handling multiple moving obstacles simultaneously.

The contents of this paper are as follows. Section 2 describes the USV dynamic model, the classification of encounter situations, and the basic process framework of USV collision avoidance decision-making based on the category related to this study. Section 3 contains a detailed description of the optimal timing point model for collision avoidance based on the improved DWA. We propose a collision avoidance algorithm based on the velocity obstacle method. In Section 4, the design of a dynamic obstacle avoidance algorithm for

USVs is considered and a detailed algorithm flow and design are presented. In Section 5, the results of computational experiments performed for the evaluation of the proposed algorithm are presented. According to simulation experiments, we compare the effects of three different algorithms on the collision avoidance path selection of a USV and analyze the degree of excellence resulting from the influence of various collision avoidance factors in path selection. Finally, in Section 6, the conclusions are discussed.

2. USV Dynamic Obstacle Avoidance Modeling

2.1. USV Dynamic Model

Figure 1 shows a schematic description of course angles, heading angles, and sideslip angles of the USV dynamic model. In this coordinate frame, x_i and y_i represent north and east directions, while v and ν are the divisions of V on the x_0- and y_0-axes, respectively [37,38].

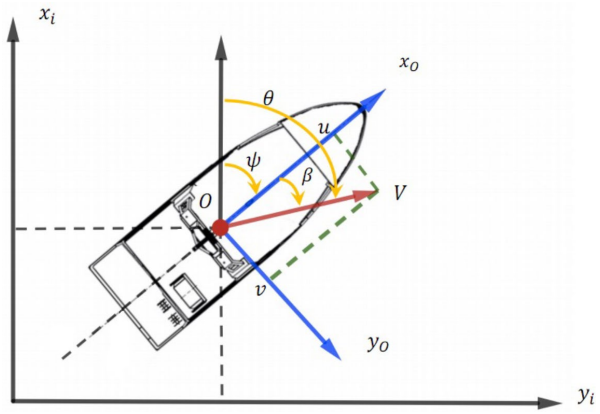

Figure 1. Schematic description of course, heading, and sideslip angles.

From Figure 1, we can see that the sideslip angle β can be calculated as $\beta = \sin^{-1}\left(\frac{v}{V}\right)$, the course angle θ can be defined using the heading angle (yaw) ψ, and the sideslip angle is $\theta = \psi + \beta$ [39,40]. The equations of the USV motion state can be represented as:

$$\begin{aligned} \dot{x} &= V\cos(\psi) \\ \dot{y} &= V\sin(\psi) \\ \dot{\psi} &= \omega \end{aligned} \tag{1}$$

where x and y represent the position coordinates of the USV, V represents the velocity of the USV, ψ represents the heading angle of the USV, and ω represents the angular velocity of the USV.

The equations of the obstacle motion state can be represented as:

$$\begin{aligned} \dot{x}_o &= v_o\cos(\psi_o) \\ \dot{y}_o &= v_o\sin(\psi_o) \end{aligned} \tag{2}$$

where x_o and y_o represent the position coordinates of the obstacle, v_o represents the velocity of the obstacle, and ψ_o represents the heading angle of the obstacle.

In this paper, we present a four-layered control structure in Figure 2, which consists of context awareness, behavioral decision-making, the obstacle avoidance algorithm, and executive control. In this study, we describe a USV marine collision avoidance strategy by the self-discipline method, t_{cpa} stands for the time to the closest point of approach, and d_{cpa} stands for the distance to the closest point of approach [41].

To support the collision avoidance of USVs in complex environments, this paper proposes a dynamic model that takes into account factors such as the mass, velocity, acceleration, water resistance, propulsive force, and gravitational acceleration of the USV in order to predict the changes in the position and velocity of the USV over time.

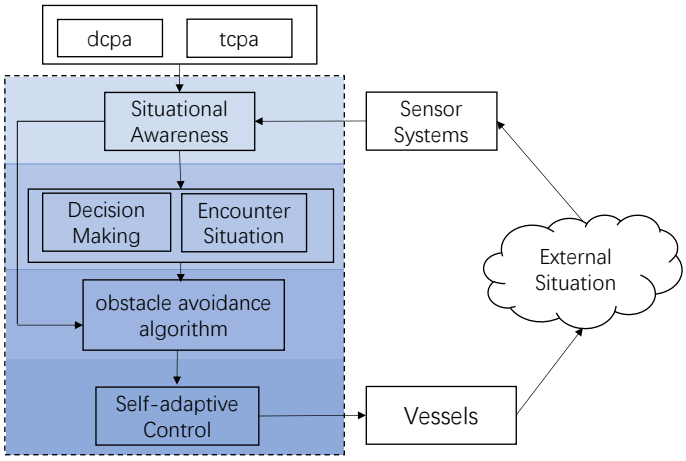

Figure 2. Architecture of intelligent vessels.

2.2. Recognition of Collision Avoidance Situations

The closest point of approach (CPA) is the point at which the distance between the ship and another target object will reach its minimum value. The geometry of the distance at CPA is illustrated in Figure 3; r is the Euclidean distance between the center points of the two ships. The equation describing the distance between the USV and obstacles can be represented as:

$$\begin{aligned} r &= \sqrt{(x - x_o)^2 + (y - y_o)^2} \\ v_r &= v - v_o - \dot{d} \end{aligned} \tag{3}$$

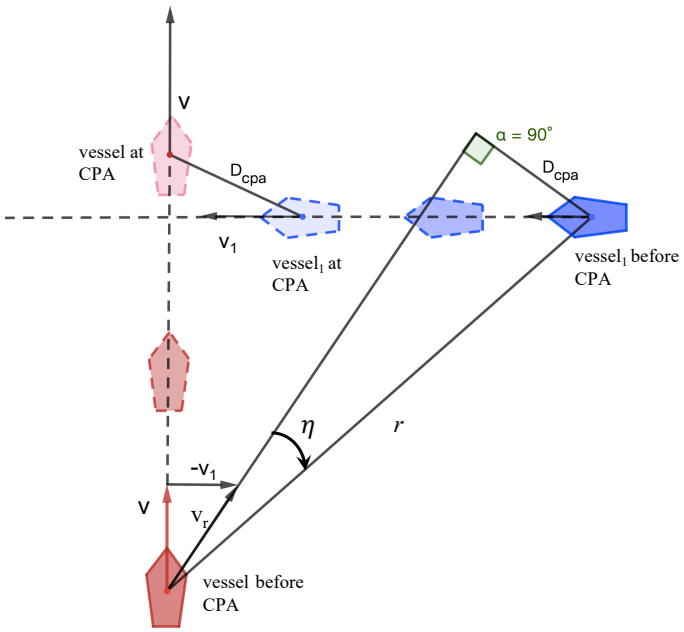

Figure 3. Distance at the closest point of approach.

CPA consists of two parameters: the distance at the closest point of approach (d_{cpa}) and the time to the closest point of approach (t_{cpa}); $\alpha$is the angle between the relative bearing of the obstacle ship and the heading of the USV [41,42].

$$\begin{aligned} d_{cpa} &= r \times \sin(\alpha) \\ t_{cpa} &= r \times \cos(\alpha) / v_r \end{aligned} \tag{4}$$

2.2.1. Safety Threshold d_{safe} Establishment

In this study, we set a safety threshold d_{safe} to evaluate the risk of a collision between vessels. A crash may occur whenever d_{cpa} falls below this safety threshold. A pathfinding algorithm based on this criterion readjusts the parameters in the path trajectory evaluation function when the distance at the closest point of approach (CPA) is below d_{safe}, which is given by

$$d_{safe} = -\frac{\frac{d_{cpa}}{r} - r}{\dot{r} \times t_{cpa}} \tag{5}$$

where γ is the range between the two vessels and $\dot{r} = -v_r \cos\eta$ is the range rate. Here, η is the incident angle and v_r denotes the relative velocity of the two vessels. If $d_{cpa} = 0$, it expresses that after the two ships have sailed for the t_{cpa} period, if they do not change their track, the two vessels will inevitably collide; $d_{cpa} > 0$ indicates that the target ship passes by the USV's bow; $d_{cpa} < 0$ indicates that the target ship passes the USV's stern [43]. However, the collision risk of the ship is not limited to $d_{cpa} = 0$: the collision risk also considers t_{cpa}. A larger t_{cpa} indicates that it will take a long time to reach the nearest encounter distance and the degree of urgency is not high. When t_{cpa} is small, it shows that the ship encounter will occur immediately and the situation is more dangerous [44].

2.2.2. Collision Avoidance Urgency Identification

This paper establishes the optimal collision avoidance velocity and the latest steering selection time for USVs based on the velocity obstacle method. At the same time, it provides a sufficient time–space margin for avoiding obstacles using the information of relative position and relative speed and the extensibility of the relative speed vector in space–time information.

As illustrated in Figure 4, circle O is a dynamic obstacle, and gray shading ξ_{all} denotes the maximum explored space of all feasible domains of linear and angular velocities obtained by using the velocity window of the DWA as the USV passes through point A. Point B is the optimal collision avoidance point.

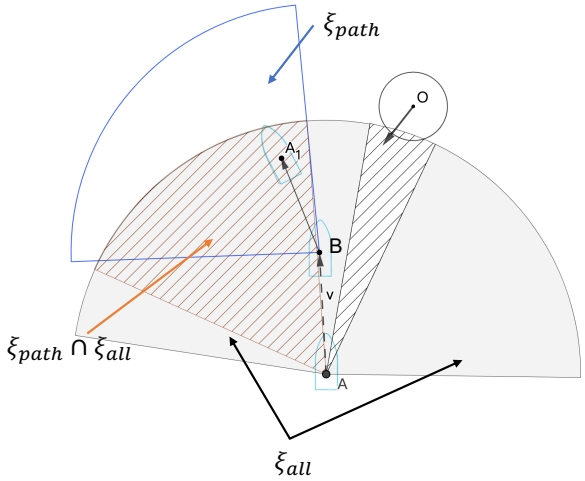

Figure 4. Velocity window sampling diagram.

Sector area ξ_{path} is the velocity window after t_{cpa} time as the feasible area of the USV, which is denoted as the blue sector area. Sector area $\xi_{path} \ominus \xi_{all}$ is the optimal area of

the USV driving area; the focus of this paper is to obtain the optimal driving area after screening the existing DWA velocity sampling window.

$$V_A(t+1) = V_A(t) + \frac{\xi_{path}(t+1) \ominus \xi_{path}(t)}{\xi_{all}} v_{max} \tag{6}$$

$$\omega_r(t+1) = \omega_{max} - \frac{v_r(t)}{v_{max}} \omega_{r(t)} \tag{7}$$

where V_A is the current speed of the USV and V_r is the relative speed of the USV relative to the incoming ship. In the optimal feasible area, the USV determines its optimal travel directly through the joint adjustment of different evaluation factors in the following formula:

$$f(d, v, \psi, \nabla) = w_1 \times g_1(d) + w_2 \times g_2(v) + w_3 \times g_3(\psi) + w_4 \times g_4(\nabla) \tag{8}$$

In Equation (8), w_1, w_2, w_3, and w_4 are four collision risk coefficients that represent the impact of different factors on the urgency of obstacle avoidance by the USV. These weight coefficients can be adjusted according to specific situations to better adapt to different obstacle avoidance scenarios; $g_1(d)$, $g_2(v)$, $g_3(\psi)$, and $g_4(\nabla)$ are function expressions of different factors representing the impact of the distance between the USV and the obstacle, the velocity of the USV, and the direction of the USV on the urgency of obstacle avoidance.

3. USV Dynamic Obstacle Avoidance Path Planning

3.1. Generation of Dynamic Obstacle Avoidance Points

To determine the optimal avoidance point that is dynamically generated based on the current position, velocity, and direction of the USV as well as the obstacle information obtained from sensors, it is necessary to compute the obstacle avoidance urgency using the function $f(d, v, \psi, \nabla)$ as defined in Equation (8). The velocity and direction of the USV are adjusted based on the obstacle avoidance urgency and dynamically generated best avoidance point. As the USV moves and encounters new obstacles, the obstacle avoidance urgency and dynamically generated best avoidance points are continuously updated to ensure safe and effective obstacle avoidance.

Algorithm 1 shows the pseudo-code to generate the optimal collision avoidance point for path candidates.

The generation of the optimal collision avoidance decision point is mainly based on the velocity obstacle method, and the optimal collision avoidance angle is added to the generated nodes (CPA) as an expanded search range. The traditional VOM algorithm uses the cost function of path length in dynamic obstacle avoidance, which can make the generated path shortest; however, the shortest-path optimization index may make the initial path close to the obstacle and increase the collision risk when the unmanned ship is driving. To solve this problem, this paper adopts the weighted cost function pair of integrated path length and optimal steering collision avoidance angle to generate the best collision avoidance decision point to ensure the safety of unmanned ship driving.

3.2. Selection and Evaluation of Obstacle Avoidance Points

In Equation (8), the w_1, w_2, w_3, and w_4 weight coefficients can be adjusted according to specific situations to better adapt to different obstacle avoidance scenarios; $g_1(d)$, $g_2(v)$, $g_3(\psi)$, and $g_4(\nabla)$ are four function expressions that can be designed and adjusted according to the actual situation to more accurately reflect the impact of different factors on the urgency of obstacle avoidance.

Algorithm 1 Generate OCAP

Require: s_t: current USV position vector; V_t: current USV velocity vector; s_t^{ob}: array of obstacle positions; V_t^{ob}: array of obstacle velocities; ψ: USV heading angle; d_{safe}: avoidance distance limit; t_{cpa}: avoidance time limit

Ensure: trajectory: array of USV trajectories

function FIND OCAP(current_Position, obstacle_Positions, obstacle_Velocities, vehicle_Velocity, vehicle_Heading, lookahead_Distance, avoidance_Distance, t_{cpa})
  trajectory = [s_t];
  t = 0
  while t < t_{cpa} **do**
    $OptimalAvoidancePoint \Leftarrow$ GENERATE OCAP(s_t, V_a, s_{ob}, V_{ob}, ψ, d_{safe}, t_{cpa}) ▷ calculate optimal collision avoidance point
    trajectory.append(Optimal Avoidance Point)
    compute $x_t = OptimalAvoidancePoint$
    (obstacle_Positions,obstacle_Velocities) = update_Obstacle_States (obstacle_Positions,obstacle_Velocities, t)
    $t = t + 1$;
  end while;
  return *trajectory*
end function

- Distance evaluation: Calculates the distance ratio between the minimum encounter distance and the safety threshold under the current USV state. The smaller the ratio, the safer it is.

$$g_1(d) = 1/(1 + \exp(-k_1 \times (d - d_{safe}))) \tag{9}$$

- Velocity evaluation: Evaluates the ratio of the time taken to reach the d_{safe} location to the t_{cpa} at the current speed of the USV. The smaller the ratio, the safer it is.

$$g_2(v) = 1 - \exp(-k_2 \times v) \tag{10}$$

- Direction angle evaluation: Evaluates the steering angle between the USV coordinates and the CPA coordinates. The larger the steering angle, the greater the risk.

$$g_3(\psi) = 1 - \exp(-k_3 \times |\psi|) \tag{11}$$

- Direction angle evaluation: The symbol ∇ represents the minimum gradient, and $g_4(\nabla)$ is the function that describes the effect of the minimum gradient on the obstacle avoidance urgency level. In this paper, the minimum gradient represents the rate of change in the distance between the USV and the obstacle, which can be used to evaluate the dynamic relationship between the USV and obstacle and the approach speed.

$$g_4(\nabla) = 1/(1 + \exp(-k_4 \times (\nabla - \nabla_0))) \tag{12}$$

The idea of the OCAP algorithm is based on the traditional DWA and is combined with the optimal collision avoidance point content in the previous section. First, the increments of vessel linear velocity and heading angle (ν, ψ) that can reach the optimal collision avoidance point are set as the constraints for dynamic window sampling. Secondly, based on the constraints, the algorithm calculates the critical conditions for the USV avoidance action. Finally, the algorithm evaluates the collision hazard risk probability on the trajectory formed by the optimized velocity region. It uses the collision avoidance probability value as an evaluation criterion to assess the merit of the route. The evaluation result selects the corresponding optimal velocity command (ν, ψ). The sampled data meet the optimal timing of collision avoidance and satisfy the collision avoidance rules to accomplish real-time obstacle avoidance and fast driving tasks in complex dynamic scenarios.

3.3. Search for Optimal Obstacle Avoidance Points

The primary focus of this chapter involves a three-step approach. Firstly, utilizing traditional speed sampling, a collision risk assessment is carried out by calculating the safe distance (d_{safe}) within the current speed window. Secondly, within the dynamic window, unsafe areas are excluded and the velocity obstacle is utilized to determine the optimal speed sample that satisfies the requirements of collision avoidance at sea. Lastly, the steering strategy of the USV is optimized to ensure that the optimized path aligns with the collision avoidance rules at sea.

Velocity and maximum acceleration limit minimum safety domain based on d_{safe}: Line AO is the connection between point A and point O, m and l are the two tangents of the obstacle safety expansion circle, $\left(\frac{\pi}{2} - \psi\right)$ is the heading angle of the USV, and $\left(\beta - \frac{\pi}{2}\right)$ is the heading angle of the incoming obstacle vessel, as shown in Figure 5. The dashed triangle represents the velocity vector triangle after the active collision avoidance behavior taken by the USV. $V_A^{'}$ is the velocity after steering, $\Delta\psi$ is the steering angle of the USV, $\Delta\eta$ is the steering angle of V_r, and $||\vec{AP'}||_2$ is the shortest encounter distance between the ship and the incoming ship after a period of time after the ship has performed collision avoidance behavior (such as steering or speed change). It is obvious that in the controllable speed sampling window, a larger d_{cpa} means a lower collision hazard probability. As long as $abs(\eta) \geq \mu$ exists at any time—that is, the sum velocity vector is located outside the triangle formed by m and l—the USV will not collide with the dynamic obstacle (shaded area shown in Figure 5).

$$\sin\mu = \frac{R}{||AO||_2} \tag{13}$$

The velocity of the USV is (V_A, ψ), the velocity of the target obstacle is (V_o, β), the speed of the USV relative to the obstacle is V_r, the relative angle can be expressed as: $\psi = \angle(X, V_a)$, $\theta = \angle(X, AO)$, $\lambda = \angle(X, V_r)$, $\varphi = \angle(V_r, V_A)$, $\mu = \angle(AO, m) = \angle(AO, l)$. It can be seen from Figure 5 that the velocity triangle is composed of v_A, v_o, and v_r.

$$\begin{aligned} V_o \sin(\psi - \beta) &= V_r \sin(\varphi) \\ V_A - V_o \cos(\psi - \beta) &= V_r \cos(\varphi) \end{aligned} \tag{14}$$

Geometric relationships between d_{cpa} and t_{cpa} are shown in the following equations.

$$d_{cpa} = ||(A, O)||_2 \sin(\psi + \varphi - \theta) \tag{15}$$

$$t_{cpa} = \frac{||(A, O)||_2}{V_r} \times \sqrt{1 - \sin^2(\theta - \psi - \varphi)} \tag{16}$$

By obtaining speed V_o and angle β of the obstacle using the sensor in advance, the USV can adjust speed V_A and angle ψ in advance to avoid the obstacle in order to change the angle to meet $abs(\eta) \geq \mu$.

According to the velocity obstacle method, the differential of the yaw angle is used to express the attitude change rate of the USV, i.e., the change rate of the attitude angle, as shown in Equation (13). Therefore, using Equations (14)–(16), we can calculate the derivative of the adjustment variable as follows:

$$\mathrm{d}\eta = \frac{sin\varphi}{V_r}\mathrm{d}V_a + \frac{V_a \cos\varphi}{V_r}\mathrm{d}\psi \tag{17}$$

$$\Delta\eta = \frac{sin\varphi}{V_r}\Delta V_A + \frac{V_a \cos\varphi}{V_r}\Delta\psi \tag{18}$$

The difference of yaw angle is used to express the attitude change amount of the USV, i.e., the change amount of the attitude angle, as shown in Equations (17) and (18).

These differential and difference values are crucial parameters in the control model of unmanned ships, as they directly affect the motion state and control the performance of the USV. By utilizing the differential and difference of the yaw, the control system can calculate control commands to adjust the attitude angles of the USV, thereby enabling it to maintain stable navigation or perform specific tasks, such as obstacle avoidance or search and rescue operations.

From Figure 5, we can see that the total angle after the relative speed is turned is $\Delta\eta + \eta$; where $abs(\eta + \Delta\eta) \geq \mu$, the USV can complete the obstacle avoidance.

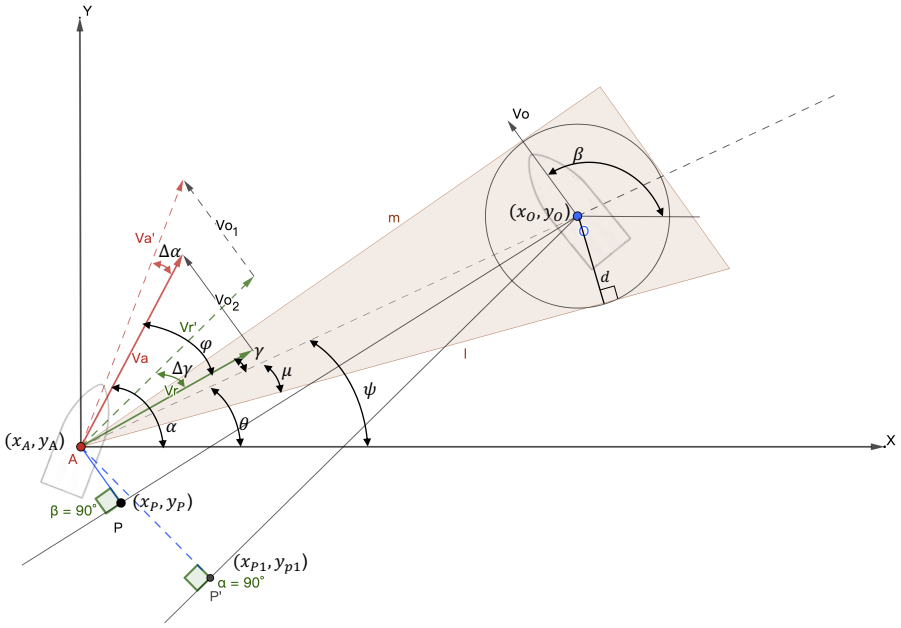

Figure 5. Improved velocity obstacle method and d_{cpa}.

4. Dynamic Obstacle Avoidance Algorithm Design for USVs

4.1. Design of the Algorithm Framework

The algorithm framework as shown in Figure 6, mainly involves three modules: dynamic sampling, dynamic obstacle avoidance, and an obstacle avoidance point-set optimization algorithm. Among them, dynamic sampling is used to determine the current optimal obstacle avoidance area, dynamic obstacle avoidance is used to calculate the optimal turning angle, and the obstacle avoidance point-set optimization algorithm is used to find the optimal obstacle avoidance point in the obstacle avoidance point set. Through this algorithm framework, the USV can achieve efficient obstacle avoidance and path planning in complex marine environments.

4.2. Design of Obstacle Avoidance Strategy

According to the calculation of d_{cpa}, the maximum total time required for USV steering is t_{cpa}; steering of at least $\Delta\eta$ is required after t_{cpa} time period to avoid dangerous areas for incoming ships and to pass at the safe closest encounter distance d_{safe}. The speed window of the USV is mainly limited by the following three factors:

1. Self maximum and minimum speed limits

$$V_s = \{(v, w) | \, v \in [v_{min}, v_{max}] \cap \omega \in [\omega_{min}, \omega_{max}]\} \tag{19}$$

The USV has a safe speed limit, and not all speeds can be used for safe USV travel. Therefore, v_{min},v_{max} represents the safe speed interval range.

2. Speed limitations affected by motor performance

$$V_d = \{(v, \omega) | v \in [v_c - \dot{v}_b \Delta t, v_c - \dot{v}_a \Delta t] \cap \omega \in [\omega_c - \dot{\omega}_b \Delta t, \omega_c - \dot{\omega}_a \Delta t]\} \tag{20}$$

3. Sampled speed affected by an obstacle.

$$V_A = \left\{ (v, w) \middle| v \leq \left[\sqrt{2dist(v, w)\dot{v}_b} \right] \cap \omega \leq \left[\sqrt{2dist(v, w)\dot{\omega}_b} \right] \right\} \tag{21}$$

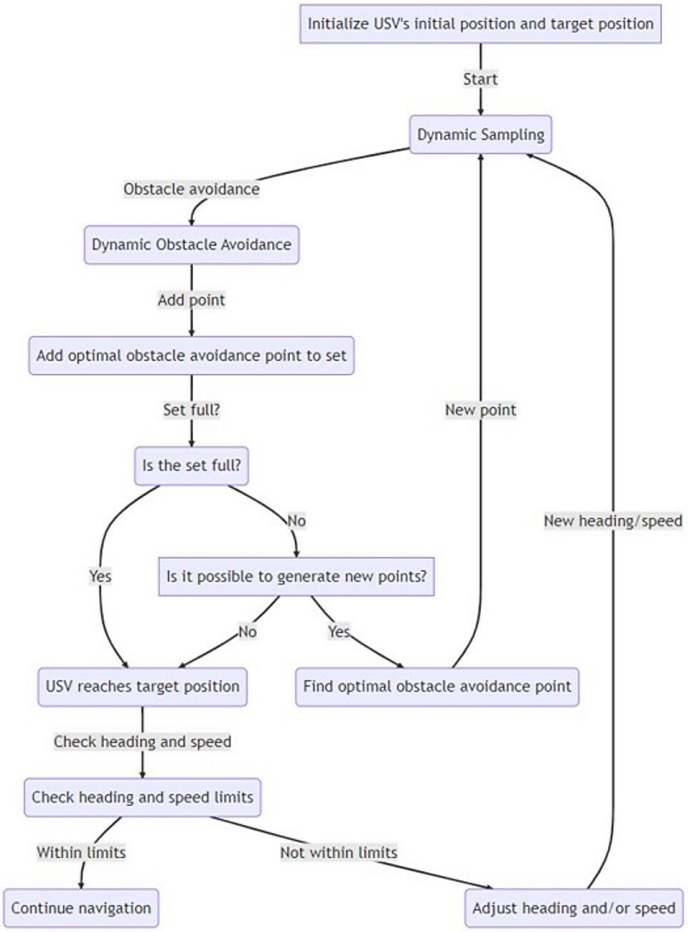

Figure 6. OCAP algorithm framework.

In this paper, combined with the optimal collision avoidance point, we focus on improving the speed V_A of the USV limited by obstacles. The classical dynamic window approach generally defines an adequate search space that conforms to the dynamic limit in the velocity space (v, ω). Still, the actual steering process of the USV is more based on the change of the heading angle $\Delta\alpha$ to complete the vessel's collision avoidance behavior. Therefore, this paper replaces the state space formed by the velocity (v, ω) with the state $(v, \Delta\alpha)$. According to the formula, the intersection V_r of V_s and V_d represents the adequate state space of the ship in the next period.

$$\begin{aligned} V_s &= \{(v, \Delta\psi) | \, v \in [v_{min}, v_{max}] \cap \Delta\psi \in [-2/\pi, 2/\pi]\} \\ V_d &= \{(v, \Delta\psi) | \, v \in [v_c - \dot{v}_b \Delta t, v_c - \dot{v}_a \Delta t] \cap \Delta\psi \in [-\Delta\eta_{max}, \Delta\eta_{max}]\} \\ V_A &= \left\{(v, \Delta\psi) \middle| \, v \in [v_0 - \dot{v_{min}} \Delta t, v_0 + \dot{v_{max}} \Delta t] \cap \Delta\psi \in \left[-\dot{\Delta\eta}_{max} \Delta t, \dot{\Delta\eta}_{max} \Delta t\right]\right\} \\ V &= V_s \cap V_d \cap V_A \end{aligned} \tag{22}$$

Equation (22) indicates that a set of possible motion trajectories can be generated by sampling the velocity and direction of the USV. Then, we evaluate these trajectories based on their cost function to select the optimal one as the USV's motion plan. Specifically, V_s is the velocity and direction sampling space, where v_{min} and v_{max} represent the minimum and maximum velocity the USV can reach and $\Delta\psi$ represents the angle range the USV can rotate. V_d represents the velocity and direction range the USV can reach while avoiding collisions, v_c is the current velocity, $\dot{v}_a$ and $\dot{v}_b$ are the acceleration and deceleration, respectively, Δt is the sampling-time interval, and $\Delta\eta_{max}$ represents the maximum angle the USV can rotate. V_A represents the velocity and direction range the USV can reach while maintaining a certain acceleration and turning speed, v_0 is the initial velocity of the USV, and $\Delta\dot{\eta}_{max}$ is the maximum angular velocity the USV can rotate. Finally, V is the intersection of the three sampling spaces, representing all possible combinations of velocity and direction that the USV can sample. By evaluating these combinations based on their cost functions, the algorithm selects the optimal motion plan to achieve the goal of collision avoidance.

In Algorithm 2, the main calculation task is to perform obstacle avoidance on the sampling points and find the optimal collision avoidance point. This task is completed by two functions, Generate OCAP (Algorithm 1) and dynamic_obstacle_avoidance. Generate OCAP (Algorithm 1) is responsible for generating a set of sampling points, and the dynamic_obstacle_avoidance function is responsible for performing obstacle avoidance on the sampling points. In these two functions, the algorithm uses sensor data and environmental information to calculate the trajectory of the unmanned boat and adjust its heading and speed based on the position and velocity of obstacles.

Algorithm 2 Evaluate optimal collision avoidance point

```
position_usv ←initial_position ;
position_target ← target_position;
while true do
    points_sampled ← Generate OCAP(position_usv, position_target)
    points_avoided ← dynamic_obstacle_avoidance(points_sampled)
    point_optimal ← find_optimal_avoidance_point(points_avoided)
    avoidance_set.add(point_optimal)
    if avoidance_set.is_full() then
        break
    end if
    if can generate new_points() then
        continue
    end if
    break
    if check heading_and_speed by Equations (20)–(23) then
        position_usv ← navigate to next_point()
    else
        adjust heading_and_speed by Equations (18) and (19)
    end if
end while
```

5. Simulations and Discussion

5.1. Parameter Selection

Three algorithms—dynamic window approach (DWA), dynamic window approach with virtual manipulators (DWV) [36], and OCAP—are considered in this study, where DWV and DWA are considered conventional methods and OCAP is envisioned as the proposed improved method. The simulations show the quality of the operation of such an algorithm. This paper observed a collision avoidance simulation using a minimum safety domain R for vessels. Moreover, the influence of navigational behaviors and environmental impacts (wind and currents) are ignored in the modeling process. All the algorithms in this study run on MATLAB 2020b. In this paper, two cases are considered: a constant map

and a random map. These maps' size for all cases is 20 × 20. Case1 is simulated once, and Case2 is simulated 100 times. The start and goal positions of all cases are (USVxstart, USVystart) = (2.0, 6.0) and (GBxgoal, GBygoal) = (19.0, 18.0). When the distance between the USV and the goal position is less than 0.3 m, it is judged to have reached the goal. The other parameters of the simulation experiment are set as shown in Table 1.

Table 1. Simulation setup.

Case	Obstacle Position	Obstacle Radius	Obstacle Type	Obstacle Velocity
Case1	Constant	1, 1.3	Constant	-
Case2	Random	Random	Random	[0 0.3]

As shown in Figure 7, there are 12 static obstacles, which is a static obstacle density of 0.04, in the simulation environment. The solid circle is the expanded range with the longest radius of the obstacle, and the dashed circle is the safe area, where the d_{safe} value of the obstacle relative to the USV is the radius. According to Equation (13), d_{safe} is only related to the size of the obstacle, which is stationary. When the obstacle moves, it is also related to the relative velocity of the USV and target vessel. The simulation cases are defined as follows. Combining Algorithms 1 and 2, the time complexity of our proposed algorithm is $O(n^3)$, where n represents the size of the input data. The algorithm consists of three nested loops and a recursive call. The time complexity of the first loop is $O(n)$, of the second loop is $O(n^2)$, and of the third loop is $O(n^3)$. The time complexity of the recursive call is $O(logn)$. Therefore, the total time complexity of the algorithm is $O(n^3 + logn)$. Additionally, the space complexity of the algorithm is $O(n)$ because it requires storing a copy of the input data in memory as well as intermediate results of multiple recursive calls.

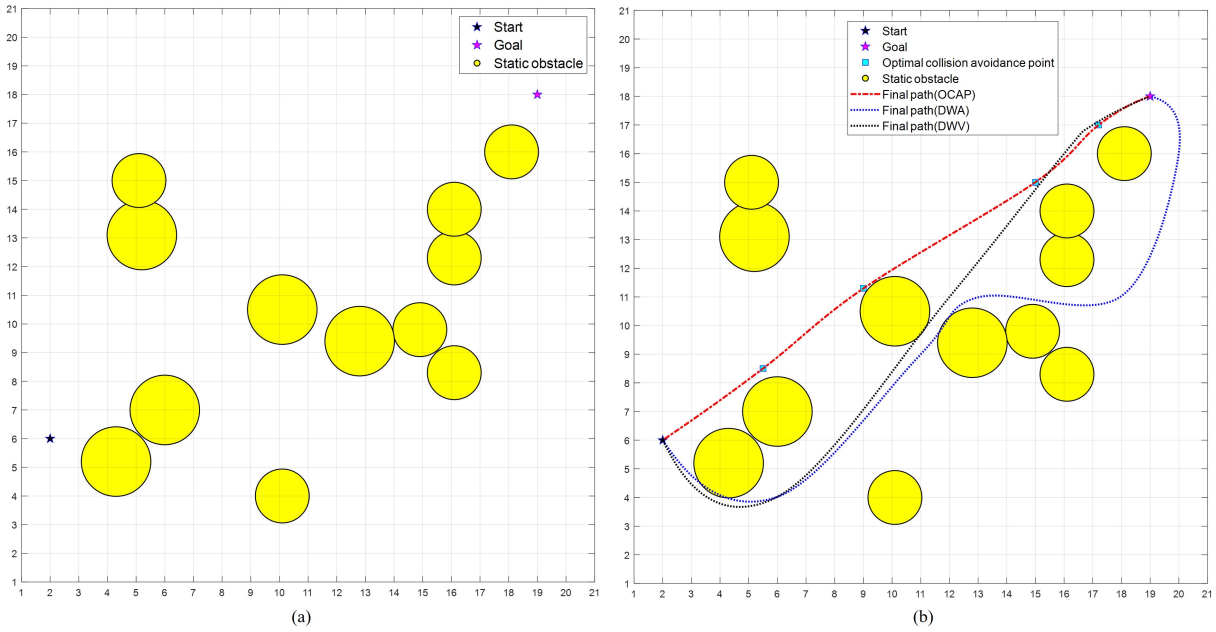

Figure 7. Simulation environment in Case1: (**a**) map at a static obstacle density of 0.04 and (**b**) comparison of paths generated by three optimization methods.

5.2. Simulation Results and Discussion

As shown in Figure 7a, there are 12 static obstacles in the simulation environment in Case1. The start and goal positions are (USVxstart, USVystart) = (2.0, 6.0) and (GBxgoal, GBygoal) = (19.0, 18.0). The density of obstacles generated on this map is 0.04.

Figure 7b displays trajectory results of OCAP, DWA, and DWV in Case1, showing that all methods reached the goal position. DWA and DWV arrived at the goal position later than OCAP.

Table 2 displays simulation results for Case1, including the success rate, algorithm time consumption (Time), trajectory length (TL), and λ_{dis} in achieving the goal without collisions, and the average travel time. OCAP reached the goal position earlier than DWA and DWV. From Figure 7b and Table 2, the best result in Case1 was obtained by OCAP, which did not need to evaluate the optimal collision points and moved at the maximum translational velocity on the optimal path. Comparing static obstacle generation paths, the efficiency and results of the OCAP and DWV algorithms are similar. However, DWA sometimes generates a 'circling' motion when avoiding obstacles to reach a specified location. Therefore, the target time of DWA is longer than those of the other two algorithms.

Table 2. Simulation results in Case1.

Case	Method	Success (%)	Time (s)	TL (m)	λ_{dis} (%)
1	DWA	100	5.377	19.521	9.04
1 time	DWV	100	3.502	15.804	9.92
	OCAP	100	3.268	14.287	4.87

Figure 8a shows simulation environments of the random maps in Case2 at a static and dynamic obstacle density of 0.04, which presents five static obstacles and seven dynamic vessels from different directions (considered dynamic obstacles), which are 12 obstacles in total. These obstacles are placed in random positions and given random velocities that are lower than the maximum velocity of the USV. The velocities of obstacles are set randomly in the range of 0.0 (m/s) to 0.2 (m/s). Case2 is simulated 100 times. Figure 8b shows path-finding results for a certain time.

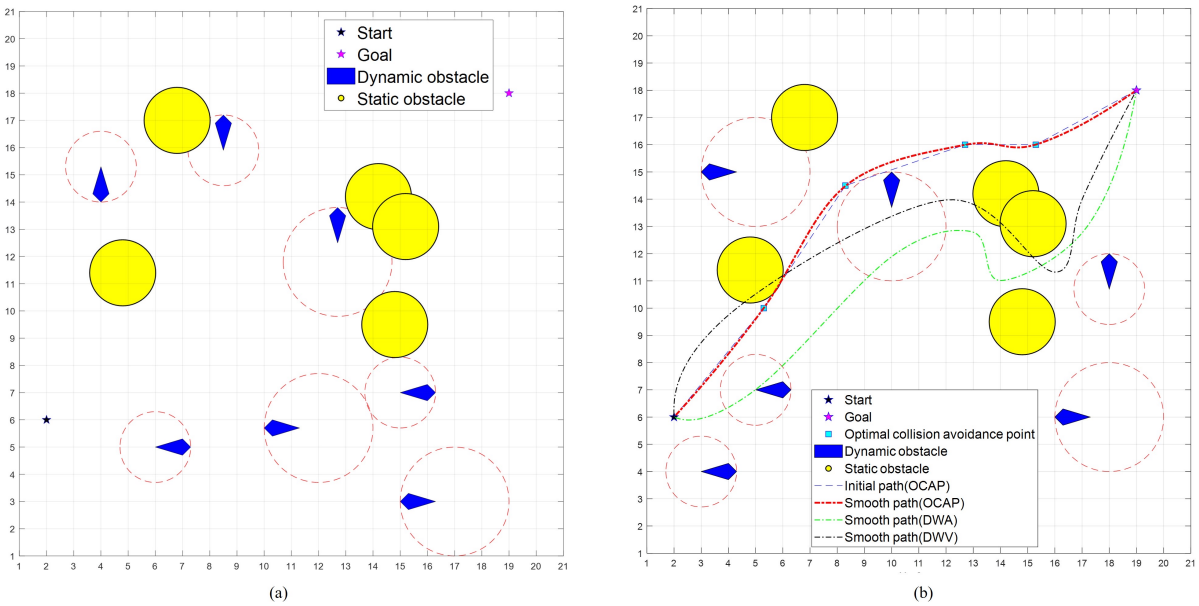

Figure 8. Simulation Environments in Case2: (**a**) random map (**b**) path generated by three algorithms.

Figure 9a–c shows only trajectories of the USV for 100 simulation times in Case2. In Figure 9a, OCAP generated path candidates considering dynamic obstacles. OCAP also considered dynamic blocks when the optimal path was selected from path candidates. Thus, OCAP reached the goal position. Overall, the path collision hazard probability obtained by the OCAP algorithm is low. The optimal obstacle avoidance trajectory results

with the lowest probabilities of collision avoidance are displayed in Figure 9b,c. DWV and DWA generated path candidates without considering dynamic obstacles. When DWA reached the goal position of avoiding moving obstacles, DWA sometimes generated backward movements.

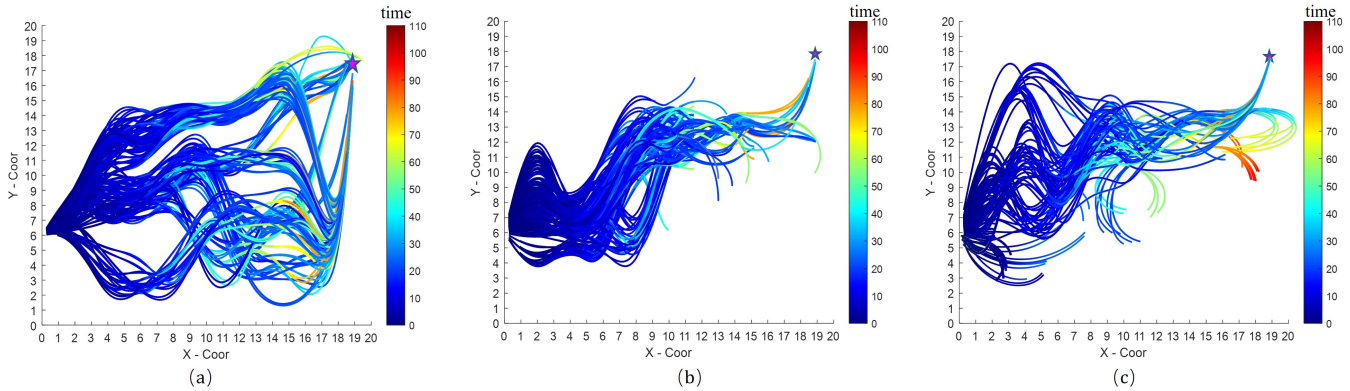

Figure 9. Simulation results of three optimization methods in Case2: (**a**) OCAP, (**b**) DWA, (**c**) DWV.

From Figure 9 and Table 3, OCAP has the highest success rate of reaching the goal at 82.73% along with the longest path-finding time and the lowest level of risk rate at 18.09% in Case2. These three results are correlated. As circling was sometimes generated to avoid the obstacle, the goal time of OCAP was longer than that of other methods. Although the time cost of the OCAP and DWV is similar, the path planned by OCAP is less likely to have collided and thus has a lower risk rate.

Table 3. Simulation results in Case2.

Case	Method	Success (%)	Time (sec)	TL (m)	λ_{dis} (%)
2	DWA	60.98	65.377	29.521	25.04
100 times	DWV	58.42	57.502	25.804	25.92
	OCAP	82.73	60.268	24.287	18.09

Figure 10 shows the evolution of four moving features during simulation. From Figure 10a,b, the obstacles are static in Case1, from which it can be seen that in the proposed new algorithm OCAP, the changes to the heading angle and course angle are minor, which means that the vessel does not have rapid turns or emergency braking during handling. The speed and t_{cpa} shown in Figure 10c,d are the two most critical parameters reflecting the USV's state of avoiding obstacles. In the new algorithm OCAP, the t_{cpa} of USV always appears within the range of change and increases linearly, indicating that the risk of collision avoidance of the USV in this algorithm is always in the acceptable range. Specifically, according to Figure 10d, in the OCAP algorithm, the minimum encounter time t_{cpa} is always greater than 0, indicating that it is effective for path planning to choose the best collision avoidance point.

The results of changing the density of obstacles in the random map and testing multiple sets of data are shown in Table 4, which includes the success rate, algorithm time consumption (T), trajectory length (TL), risk rate (λ_{dis}) in achieving the goal without collisions, and the average travel time.

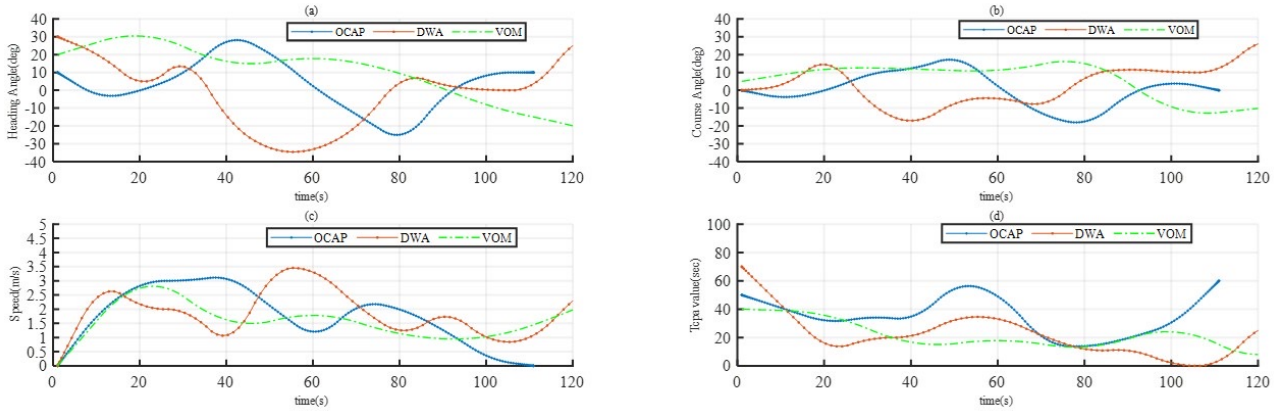

Figure 10. Three algorithm simulation parameters compared in Case2: (**a**) heading angle, (**b**) course angle, (**c**) USV speed to collision, (**d**) USV t_{cpa}.

Table 4. Comparison of OCAP and conventional algorithms at different obstacle densities.

Obstacle Density (%)	Method	Success	T (sec)	TL (m)	λ_{dis} (%)
0.02	DWV	92.340	3.157	28.496	5.281
	DWA	94.554	2.976	25.765	5.094
	OCAP	98.231	2.800	17.027	5.090
0.06	DWV	12.340	81.653	48.064	33.317
	DWA	34.554	88.910	45.109	40.338
	OCAP	80.231	79.772	37.407	26.060
0.08	DWV	10.630	156.157	98.496	53.281
	DWA	13.800	182.976	95.765	52.094
	OCAP	67.781	106.278	87.027	20.122

From Table 4, when the obstacle density is smallest (0.02), the path-finding success rates of all three algorithms are high and not much different; however, as the density of random map obstacles increases, the path-finding success rates of all three algorithms decrease, among which the decrease rate of the DWV algorithm changes the most. This is because the DWV algorithm is more sensitive to the number of obstacles: the more obstacles, the more DWV demonstrates the characteristics of a breadth-first algorithm that will fail to complete the computation in the specified time, making the sharpest decline in the success rate.

Comparing the computation time of the three algorithms, the OCAP algorithm consumes less time to generate dynamic discrete points than dynamic path generation because the optimal collision avoidance points are generated in advance before the path is generated. At the maximum obstacle density (0.08), the path-finding success rate of all three algorithms decreases greatly. However, since the OCAP algorithm generates dynamic discrete points, if the path changes (e.g., previously unmeasured dynamic obstacles) before driving the generated path, only the best collision avoidance point needs to be measured again, thus effectively improving the path-finding success rate. It can also be seen that the length of the path generated by the three algorithms is about the same when the density is large, and the main difference lies in the success rate of generation and the time consumed by the algorithm.

From Figure 11, the path-planning performance of three different algorithms on three cases is analyzed as a whole. In two other cases, the path planned by the OCAP algorithm has the shortest length and the least time cost. OCAP has the highest success rate of reaching the goal at 100%. The best result in Case2 was obtained by OCAP, which shows that this algorithm is more suitable for avoiding low-speed dynamic obstacles.

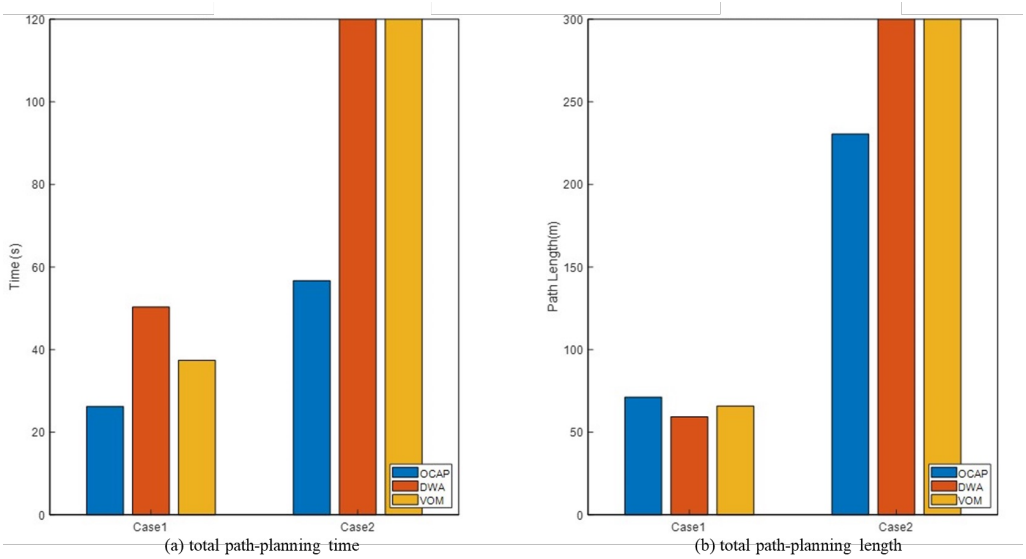

Figure 11. Three algorithms in optimal path-planning performance: (**a**) total path-planning time (**b**) total path-planning length.

6. Conclusions

In this paper, OCAP, a new collision avoidance algorithm, is proposed. OCAP generated obstacle-avoidable path candidates. Path candidates were generated using the optimal collision avoidance point based on predictions of static and dynamic obstacles. Kinematics and dynamics constraints were taken into account in OCAP. The paper used simulations and experiments, demonstrating the proposed method to be effective. Even when the obstacle density increases, the effectiveness of trajectory generation is ensured because the OCAP algorithm can effectively and dynamically evaluate the minimum obstacle avoidance distance. Through simulation experiments, the algorithm was shown to be more suitable for high-density environments, and by evaluating the optimal collision avoidance points, the generated paths can be kept away from the obstacles over a larger area. The results of this study are limited to situations based on ship encounters in the calm water conditions considered in this study. Additionally, no consideration was made for hull-to-hull interaction and hull–propeller–rudder–engine interaction between the two vessels, which is a direction for future research.

Author Contributions: Conceptualization, Y.D. and H.Z.; methodology, Y.D.; software, H.Z.; validation, Y.D. and H.Z.; formal analysis, H.Z.; resources, H.Z.; data curation, Y.D.; writing—original draft preparation, Y.D.; writing—review and editing, H.Z.; supervision, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Program for Scientific Research Startup Funds of Guangdong Ocean University, grant number: R17015; Zhanjiang Scientific and Technological Research Topics, grant number: 2022B01103; and the National College Students Innovation and Entrepreneurship Training Program of Guangdong Province, grant number: S202210566074.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

OCAP	Optimal Collision Avoidance Point
DWA	Dynamic Windows Approach
DWV	Dynamic Window Approach with Virtual Manipulators
CPA	Closest Point of Approach
USV	Unmanned Surface Vehicle

References

1. Qu, C.; Gai, W.; Zhang, J.; Zhong, M. A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (UAV) path planning. *Knowl.-Based Syst.* **2020**, *194*, 15–35. [CrossRef]
2. Stateczny, A.; Gierlowski, K.; Hoeft, M. Wireless Local Area Network Technologies as Communication Solutions for Unmanned Surface Vehicles. *Sensors* **2022**, *22*, 655. [CrossRef] [PubMed]
3. Wang, C.; Chen, D.; Liao, W.; Liang, Z. Autonomous Obstacle Avoidance Strategies in the Mission of Large Space Debris Removal using Potential Function. *Adv. Space Res.* 2022, *in press*. [CrossRef]
4. Huang, Y.; Chen, L.; Chen, P.; Negenborn, R.R.; van Gelder, P. Ship collision avoidance methods: State-of-the-art. *Saf. Sci.* **2020**, *121*, 451–473. [CrossRef]
5. Chai, T.; Weng, J.; De-qi, X. Development of a quantitative risk assessment model for ship collisions in fairways. *Saf. Sci.* **2017**, *91*, 71–83. [CrossRef]
6. Rajko, S.; Lavalle, S.M. A pursuit-evasion BUG algorithm. In Proceedings of the 2001 IEEE ICRA International Conference on Robotics and Automation (Cat. No.01CH37164), Seoul, Republic of Korea, 21–26 May 2003.
7. Pommerenck, J.K.; Roundy, D. Flat-histogram method comparison on the two-dimensional Ising model. *Phys. Rev. E* **2020**, *102*, 033306. [CrossRef]
8. Hong, M.J.; Arshad, M.R. A Balance-Artificial Potential Field Method for Autonomous Surface Vessel Navigation in Unstructured Riverine Environment. *Procedia Comput. Sci.* **2015**, *76*, 198–202. [CrossRef]
9. Abebe, M.; Noh, Y.; Kang, Y.J.; Seo, C.; Kim, D.; Seo, J. Ship trajectory planning for collision avoidance using hybrid ARIMA-LSTM models. *Ocean. Eng.* **2022**, *256*, 11–27. [CrossRef]
10. Heiberg, A.; Larsen, T.N.; Meyer, E.; Rasheed, A.; San, O.; Varagnolo, D. Risk-based implementation of COLREGs for autonomous surface vehicles using deep reinforcement learning. *Neural Netw.* **2022**, *152*, 17–33. [CrossRef]
11. Han, S.; Wang, L.; Wang, Y. A COLREGs-compliant guidance strategy for an underactuated unmanned surface vehicle combining potential field with grid map. *Ocean. Eng.* **2022**, *255*, 111355. [CrossRef]
12. Campbell, S.; Naeem, W.; Irwin, G. A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. *Annu. Rev. Control.* **2012**, *36*, 267–283. [CrossRef]
13. Goodwin, E.M. A Statistical Study of Ship Domains. *J. Navig.* **1975**, *28*, 328–344. [CrossRef]
14. Zhou, J.; Ding, F.; Yang, J.; Pei, Z.; Wang, C.; Zhang, A. Navigation safety domain and collision risk index for decision support of collision avoidance of USVs. *Int. J. Nav. Archit. Ocean. Eng.* **2021**, *13*, 340–350. [CrossRef]
15. Ibadurrahman; Hamada, K.; Wada, Y.; Nanao, J.; Watanabe, D.; Majima, T. Long-Term Ship Position Prediction Using Automatic Identification System (AIS) Data and End-to-End Deep Learning. *Sensors* **2021**, *21*, 7169. [CrossRef]
16. Tam, C.; Bucknall, R.; Greig, A. Review of Collision Avoidance and Path Planning Methods for Ships in Close Range Encounters. *J. Navig.* **2009**, *62*, 455–476. [CrossRef]
17. Szlapczynski, R. Evolutionary Sets Of Safe Ship Trajectories: A New Approach To Collision Avoidance. *J. Navig.* **2011**, *64*, 169–181. [CrossRef]
18. Polvara, R.; Sharma, S.; Wan, J.; Manning, A.; Sutton, R. Obstacle Avoidance Approaches for Autonomous Navigation of Unmanned Surface Vehicles. *J. Navig.* **2018**, *71*, 241–256. [CrossRef]
19. Zhao, Y.; Li, W.; Shi, P. A real-time collision avoidance learning system for Unmanned Surface Vessels. *Neurocomputing* **2016**, *182*, 255–266. [CrossRef]
20. Yu, Y.; Chen, L.; Shu, Y.; Zhu, W. Evaluation model and management strategy for reducing pollution caused by ship collision in coastal waters. *Ocean. Coast. Manag.* **2021**, *203*, 105446. [CrossRef]
21. Szlapczynski, R. Evolutionary Sets of Safe Ship Trajectories within Traffic Separation Schemes. *J. Navig.* **2013**, *66*, 65–81. [CrossRef]
22. Brcko, T.; Androjna, A.; Srse, J.; Boć, R. Vessel Multi-Parametric Collision Avoidance Decision Model: Fuzzy Approach. *J. Mar. Sci. Eng.* **2021**, *9*, 49. [CrossRef]
23. Hasegawa, K.; Kouzuki, A. Automatic Collision Avoidance System for Ships Using Fuzzy Control. *J. Kansai Soc. Nav. Archit.* **1987**, *205*, 1–10.
24. Wu, B.; Yip, T.L.; Yan, X.; Guedes Soares, C. Fuzzy logic based approach for ship-bridge collision alert system. *Ocean. Eng.* **2019**, *187*, 106152. [CrossRef]
25. Goerlandt, F.; Kujala, P. Traffic simulation based ship collision probability modeling. *Reliab. Eng. Syst. Saf.* **2011**, *96*, 91–107. [CrossRef]

26. Horteborn, A.; Ringsberg, J.W. A method for risk analysis of ship collisions with stationary infrastructure using AIS data and a ship manoeuvring simulator. *Ocean. Eng.* **2021**, *235*, 109396. [CrossRef]
27. Fox, D.; Burgard, W.; Thrun, S. The dynamic window approach to collision avoidance. *IEEE Robot. Autom. Mag.* **1997**, *4*, 23–33. [CrossRef]
28. Chang, L.; Shan, L.; Jiang, C.; Dai, Y. Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment. *Auton. Robot.* **2020**, *45*, 51–76. [CrossRef]
29. Hall, J.; Anderson, D. Reactive route selection from pre-calculated trajectories - application to micro-UAV path planning. *Aeronaut. J.* **2011**, *115*, 635–640. [CrossRef]
30. Kabir, A.M.; Thakar, S.; Malhan, R.K.; Shembekar, A.V.; Gupta, S.K. Generation of synchronized configuration space trajectories with workspace path constraints for an ensemble of robots. *Int. J. Robot. Res.* **2021**, *40*, 98808. [CrossRef]
31. Lv, Z.; Jie, Z.; Jin, J.; Qi, L.; Gao, B. Energy Consumption Research of Mobile Data Collection Protocol for Underwater Nodes Using an USV. *Sensors* **2018**, *18*, 1211. [CrossRef]
32. Dobrevski, M.; Skočaj, D. Adaptive Dynamic Window Approach for Local Navigation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October–24 January 2020; pp. 6930–6936. [CrossRef]
33. Liu, L.; Yao, J.; He, D.; Chen, J.; Huang, J.; Xu, H.; Wang, B.; Guo, J. Global Dynamic Path Planning Fusion Algorithm Combining Jump-A* Algorithm and Dynamic Window Approach. *IEEE Access* **2021**, *9*, 19632–19638. [CrossRef]
34. Zhu, M.; Hahn, A.; Wen, Y.Q.; Sun, W.Q. Optimized support vector regression algorithm-based modeling of ship dynamics. *Appl. Ocean. Res.* **2019**, *90*, 101842. [CrossRef]
35. Kobayashi, M.; Motoi, N. Local Path Planning Method Based on Virtual Manipulators and Dynamic Window Approach for a Wheeled Mobile Robot. In Proceedings of the 2021 IEEE/SICE International Symposium on System Integration (SII), Iwaki, Fukushima, Japan, 11–14 January 2021; pp. 499–504. [CrossRef]
36. Kobayashi, M.; Motoi, N. Local Path Planning: Dynamic Window Approach With Virtual Manipulators Considering Dynamic Obstacles. *IEEE Access* **2022**, *10*, 17018–17029. [CrossRef]
37. Muske, K.R.; Ashrafiuon, H.; Haas, G.; Mccloskey, R.; Flynn, T. Identification of a control oriented nonlinear dynamic USV model. In Proceedings of the 2008 IEEE American Control Conference, Seattle, WA, USA, 11–13 June 2008.
38. Shin, J.; Kwak, D.J.; Lee, Y.i. Adaptive Path-Following Control for an Unmanned Surface Vessel Using an Identified Dynamic Model. *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 1143–1153. [CrossRef]
39. Tristan, P.; Tzeng, C.-Y.; Goodwin, G.C. Model Predictive Rudder Roll Stabilization Control for Ships. *IFAC Proc. Vol.* **2000**, *33*, 45–50.
40. Kim, D.H. Human factors influencing the ship operator's perceived risk in the last moment of collision encounter. *Reliab. Eng. Syst. Saf.* **2020**, *203*, 107078. [CrossRef]
41. Silan, J.L.; Niemann, D.L.; Ribaya, B.P.; Rahman, M.; Meyyappan, M.; Nguyen, C.V. Carbon nanotube pillar arrays for achieving high emission current densities. *Appl. Phys. Lett.* **2009**, *95*, 56–69. [CrossRef]
42. Komen, D.; Neilsen, T.B.; Mortenson, D.B.; Acree, M.C.; Hodgkiss, W.S. Seabed type and source parameters predictions using ship spectrograms in convolutional neural networks. *J. Acoust. Soc. Am.* **2021**, *149*, 1198–1210. [CrossRef]
43. Szlapczynski, R.; Szlapczynska, J. A ship domain-based model of collision risk for near-miss detection and Collision Alert Systems. *Reliab. Eng. Syst. Saf.* **2021**, *214*, 107766. [CrossRef]
44. Szlapczynski, R.; Szlapczynska, J. Review of ship safety domains: Models and applications. *Ocean. Eng.* **2017**, *145*, 277–289. [CrossRef]