IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 July 2024, accepted 12 August 2024, date of publication 15 August 2024, date of current version 11 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3444314

RESEARCH ARTICLE

Fast Marching Based Rendezvous Path Planning for a Team of Heterogeneous Vehicles

JAEKWANG KIM[1], HYUNG-JUN PARK[2], ADITYA PENUMARTI[3], AND JAEJEONG SHIN[3], (Member, IEEE)
[1] Department of Mechanical and Design Engineering, Hongik University, Sejong 30016, Republic of Korea
[2] School of Mechanical and Aerospace Engineering, Sunchon National University, Suncheon 57922, Republic of Korea
[3] Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611, USA

Corresponding author: Jaejeong Shin (jane.shin@ufl.edu)

This work was supported in part by the Hongik University New Faculty Research Support Fund and in part by the National Research Foundation of Korea Grant funded by the Korean Government under Grant RS-2024-00333943.

ABSTRACT This paper presents a formulation for deterministically calculating optimized paths for a multi-agent system consisting of heterogeneous vehicles. The key idea is the calculation of the shortest time for each agent to reach every grid point from its known initial position. Such arrival time map is efficiently computed using the Fast Marching Method (FMM), a computational algorithm originally designed for solving boundary value problems of the Eikonal equation. By leveraging the FMM, we demonstrate that the minimal time rendezvous point and paths for all member vehicles can be uniquely determined with minimal computational overhead. The scalability and adaptability of the present method during online execution are investigated, followed by a comparison with a baseline method that highlights the effectiveness of the proposed approach. Then, the potential of the present method is showcased through a virtual rendezvous scenario involving the coordination of a ship, an underwater vehicle, an aerial vehicle, and a ground vehicle, all converging at the optimal location within the Tampa Bay area in minimal time. The results show that the developed framework can efficiently construct continuous paths of heterogeneous vehicles by accommodating operational constraints via an FMM algorithm.

INDEX TERMS Autonomous vehicles, fast marching method, heterogeneous vehicle system, multi-agent system, path planning.

I. INTRODUCTION

Recent advancements in various types of autonomous vehicles have sparked interest in multi-agent systems, which hold the potential to efficiently address complex tasks. Strategic multi-agent path finding (MAPF) becomes crucial, particularly when the team comprises heterogeneous vehicles with varying operational domains and capabilities, such as different speeds, sizes, and maneuverability. These agents may encompass a wide range of vehicles, including ships, underwater vehicles, aerial vehicles, and ground vehicles. Each type of vehicle can have unique navigational and environmental constraints depending on each one's operation domain [1], [2]. Incorporating heterogeneous vehicles across multiple domains enhances the system's ability to handle complex and large-scale operations, significantly impacting real-world applications such as autonomous vehicle fleets for delivery, disaster response teams, and environmental monitoring.

Previous studies in multi-agent planning for homogeneous systems have primarily concentrated on scheduling, with an emphasis on task allocation and agent coordination [3], [4]. As scheduling algorithms focus on finding the optimal sequences and coordination among heterogeneous vehicles, the continuous path planning of each agent is often neglected and considered as a lower level problem. However, considering continuous path planning[1] ensures smooth and uninterrupted motion of vehicles [5], since all

The associate editor coordinating the review of this manuscript and approving it for publication was Su Yan.

[1] Here, a continuous path refers to a path defined on continuous real-world space and thus can serve as a smooth path for autonomous vehicles.

agents in real-world applications must adapt their paths in response to changing environmental conditions and dynamic obstacles. Similar problems have been considered as rendezvous (RDV) search problems as well. However, rendezvous search approaches in existing literature focus on finding the optimal strategy for rendezvous given limited sensing and communication capabilities among homogeneous agents [6], [7], [8], [9], [10], [11], [12], [13]. The common assumptions in rendezvous search approaches, such as unknown environment, unknown initial conditions, and asynchronous systems, are important to consider in certain scenarios; however, these assumptions may be relaxed in a larger-scale scenarios where heterogeneous systems are usually deployed.

The aforementioned rendezvous path planning can also be viewed from the perspective of computational science, as a general form of the continuous MAPF problem, which is known as an NP-hard problem [14]. The MAPF problems, however, mostly focus on collision avoidance among the agents. Therefore, it is not possible to directly apply existing MAPF methods to the considered rendezvous path planning problem. Moreover, one of the main challenges in MAPF is related to the high dimensionality of the problem. With many agents in an environment, the number of potential paths and interactions can become overwhelmingly large. The complexity of the multi-agent path finding problem also stiffly increases, as the problem as the number of agents in the system increases, and thus solving MAPF problems becomes computationally expensive. In addition, in many cases, efficiency is not the sole concern; safety (collision-free paths) must also be taken into account. Due to the complex nature and conflicting objectives encountered in MAPF problems, one often needs to reduce or approximate the original problem to a simpler form, compromising accuracy and global optimality. The comparison between the existing and presented problems is summarized in Table 1. The table shows that why the existing approaches cannot be directly applied to solve the rendezvous path planning problem considered in this paper.

In this work, we consider continuous path planning for a multi-agent system for minimal time rendezvous tasks. In these tasks, some agents initially operating at different locations are tasked with meeting to exchange information or resources. Such tasks are frequently encountered in spacecraft docking scenarios [15], [16]. A team of autonomous underwater vehicles also often needs to initiate information exchange tasks at close distances due to limited data transfer capabilities in deep water [17]. In these scenarios, identifying the optimal rendezvous point and the path for each agent to achieve the earliest possible rendezvous time as a team (or other optimizing goals) is important. Unfortunately, however, planning paths that accommodate differences of vehicles, while optimizing overall performance remains a significant challenge.

TABLE 1. Comparison of existing approaches to the presented FMM-based method on the rendezvous path planning problem for heterogeneous vehicle systems. The first column (RDV) indicates whether the approach can consider rendezvous point search. The second column (Path Finding) shows if the approach can compute a continuous trajectory in the environment. The third column (Heterogeneous) specifies whether the approach can apply to a multi-agent system consisting of heterogeneous agents operating in multiple domains. The fourth column (Obs.) denotes if the method can consider obstacles in the environment.

Approaches	RDV	Path Finding	Heterogeneous	Obs.
Scheduling	○	×	○	×
RDV Search	○	×	×	○
MAPF	×	○	×	○
Presented	○	○	○	○

The primary contribution of this paper lies in formulating the rendezvous problem for a multi-agent system in a way that is suitable for assessment using the fast marching method (FMM). The FMM is a well-established numerical technique originally developed for solving the Eikonal equation. Beyond its original purpose, however, the FMM has also demonstrated its capability in efficiently computing the shortest paths on continuous grids [18], [19], [20], [21], [22]. Extending these works, we show how the use of the FMM for rendezvous MAPF also enables the enhancement of collaboration, reduction of complexity, and optimization of the overall mission performance of the team. Specifically, we first define an optimization problem that involves continuous path planning for a team of heterogeneous vehicles, each with its own operational domain. Then, we exploit the direct output from the FMM as a key component of a new path planning approach. Our approach deterministically calculates the time-optimal rendezvous point for heterogeneous vehicles and determines the path to the optimal rendezvous point from different initial agent positions. Throughout this process, the method also takes into account their unique operational constraints.

The remainder of the paper is organized as follows. Section II introduces the methodologies of the FMM and FMM-based path planning. In Section III, we formulate an optimization problem for multi-agent path planning of a rendezvous task and introduce a new methodology to efficiently solve the problem. Section IV presents a virtual path planning experiment to demonstrate the potential of our proposed approach, while Section V discusses important features and highlights the merits of the suggested methodology. Finally, we conclude the paper in Section VI, listing potential future research directions.

II. BACKGROUND ON FAST MARCHING METHOD AND ITS APPLICATION TO PATH OPTIMIZATION

In this section, we provide a brief overview of the FMM, which will be used to address the challenges of multi-agent path planning for rendezvous missions. Originally developed for solving a nonlinear first-order partial differential equation, the FMM has shown high efficiency in dealing with interface mechanics compared to other algorithms [23], [24],

[25], [26]. The FMM has also found applications in diverse research domains, encompassing materials science [27], computer graphics [28], wave propagation [29], and image processing [30]. Particularly, its application to path optimization has a long history in various domains of applications, ranging from marine vehicles to social navigation [31], [32], [33], [34], [35]. Since the FMM is a numerical technique for solving the propagation of interfaces (or waves), its applications are determined by how the speed function for interface propagation is defined. For instance, in the field of image segmentation, the sign of the speed function flips when the interface reaches an object, causing the interface to propagate outward along the object's boundary. On the other hand, in path planning, a speed function based on the distance to obstacles is often used to dampen the speed of a vehicle as it approaches an obstacle, as detailed in Section II-B. While the FMM can be applied across various research fields, the assumption that the medium through which interfaces propagate is homogeneous and isotropic remains a challenging issue for specific applications. In the following, we begin by summarizing the main ideas of the FMM in its original context.

A. THE FAST MARCHING METHOD

First introduced in [36], the FMM is an efficient computational algorithm for tracking the front, or *interface*, that evolves with the outward unit normal direction with speed V. The explicit outcome of FMM is the arrival time $T(\boldsymbol{x})$ that the initial surface needs to reach every point $\boldsymbol{x}$ in the given domain Ω. For example, Fig. 1 demonstrates the result of FMM used to track an initial surface Γ (the innermost blue line) growing with a uniform outward normal velocity $V(\boldsymbol{x}) = 1$. Below, we outline the FMM algorithm as described in [37].

Let $s(t)$ describe a surface evolving at speed $V(\boldsymbol{x})$ from a given initial surface $s(0) = \Gamma$. Instead of solving a time-dependent problem for $s(t)$ to track the moving surface, the FMM solves a function $T(\boldsymbol{x})$ defined as

$$T(s(t)) = t, \tag{1}$$

with $T = 0$ on Γ. Differentiating (1) and noting that ∇T is normal to the surface, one arrives at the following boundary value problem,

$$|\nabla T|V = 1. \tag{2}$$

Also, the boundary condition for T equivalent to the original time-dependent problem is

$$T = 0 \quad \text{on } \Gamma. \tag{3}$$

Equation (2) is commonly referred to as the Eikonal equation.

Now, we describe the algorithm to solve (2) on a two-dimensional discrete grid, i.e., $\boldsymbol{x} = (x, y)$. However, it is worth noting that the algorithm can be conveniently generalized to arbitrary dimensions. Let $D^{-x}_{ij}(\cdot)$ denote the standard backward-difference operation on the grid point ij

$$D^{-x}_{ij}T = \frac{T_{ij} - T_{(i-1)j}}{\delta x}. \tag{4}$$

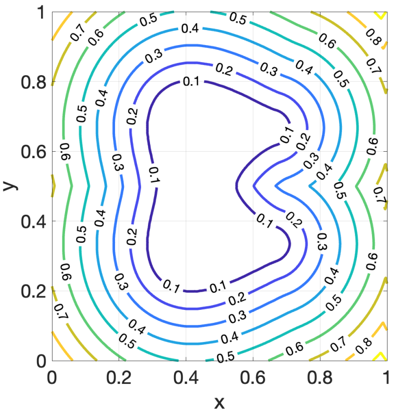


FIGURE 1. The level sets of the solution to the Eikonal equation (2) computed using the fast marching method, describe a surface evolving with outward normal velocity $V(x) = 1$. The level set values are indicative of the time it takes for the initial surface (represented by the innermost blue line) to reach each grid point within the computational domain.

Likewise, we use D^{+x}, D^{-y}, and D^{+y} to represent forward in x, backward and forward in y backward finite difference operators, respectively. In order to ensure a unique *viscosity solution* for the Eikonal equation (2), we necessitate the consistent utilization of an upwind finite difference scheme when computing the gradient. This step is compactly written as

$$\frac{1}{V(\boldsymbol{x})} = \Big[(\max(D^{-x}_{ij}T, -D^{+x}T_{ij}, 0)^2 + \max(D^{-y}_{ij}T, -D^{+y}T_{ij}, 0)^2\Big]^{1/2} \tag{5}$$

When the neighboring values of T_{ij} are known, the discrete Eikonal equation (5) becomes a quadratic equation for T_{ij} at each grid point, allowing for straightforward analytical solutions.

The FMM initiates by performing the following initialization step.

1) Assign $T(\boldsymbol{x}) = 0$ for grid points in the area enclosed by the initial surface, and tag them as *accepted*.
2) Assign $T(\boldsymbol{x}) = +\infty$ for the remaining grid points, and tag them as *far*.
3) Among the *accepted* points, identify the points that are in the neighborhood of points tagged as *far*, and tag them as *considered*.

The key step in the fast marching method is to update T with a trial value using Eq. (5) for grid points tagged as *considered*, while only accepting the update with the smallest value at each iteration. This procedure requires keeping track of the smallest T-value among points tagged as *considered*. The potential T values are managed in a specialized data structure inspired by discrete network algorithms [38]. This data structure is known as a min-heap data structure, which represents a complete binary tree with a property that the value at any given node is less than or equal to the values of its children. Utilizing the min-heap, the FMM then proceeds as follows.

1) Form a min-heap structure for the *considered* points.
2) Access the minimum value of the heap, located at the root of the binary tree.
3) Determine a trial solution $\tilde{T}$ on the neighbors of the root using (5). If the trial solution $\tilde{T}$ is smaller than the present values, then update $T(\boldsymbol{x}) = \tilde{T}$.
4) If a point, previously tagged as *far*, is updated using a trial value, relabel it as *considered*, and add it to the heap structure.
5) Tag the root of the heap as *accepted*, and delete it from the heap.
6) Repeat steps 2 to 5, until every grid point is tagged as *accepted*.

In a min-heap, the time complexity for insertion and deletion operations is $\mathcal{O}(\log n_e^H)$, where n_e^H is the number of elements in the heap, and $\mathcal{O}(\cdot)$ denotes Big O notation, a limiting behavior of a function. The notation expresses an upper bound on the execution time required by an algorithm. The time complexity of the mean-heap arises from its binary tree structure, which ensures that the height of the tree is $\log n_e^H$. Consequently, only a logarithmic number of comparisons are needed for adding or removing elements, enhancing the efficiency of these operations. In the FMM, every grid point has a mean-heap structure, resulting in a time complexity of $\mathcal{O}(n \log n)$, where n is the total number of grid points.

B. ADAPTATION OF THE FMM FOR PATH OPTIMIZATION

While the FMM is originally developed for interface problems, numerous studies have also successfully applied the FMM in vehicle path planning scenarios, enabling agents to navigate complex environments, avoid obstacles, and reach their destinations efficiently. These studies have primarily focused on single-agent path planning under various external conditions. These include time-varying environmental factors, such as waves and currents in oceans [18], time-varying environments with predictive models [19], angle guidance for uncrewed surface vehicles [39], the anisotropic Fast Marching (FM)-based approaches for dynamic obstacles [20] and bridge obstacles [21], as well as path planning for autonomous ships [22]. In contrast, its application in multi-agent systems remains relatively unexplored. A few examples include swarm coordination [40] and formation control involving vehicles with different dynamic properties [41].

In the context of path optimization, the computational domain Ω of the FMM takes on a new perspective as the configuration space for mobile agents, often depicted through a binary occupancy map as illustrated in Fig. 2. The binary image, which is in a size of $n = n_1 \times n_2$ pixels, takes the value of 0 if the position is occupied by obstacles, and 1 otherwise. Also, the initial surface Γ is reduced to a single wave-source point $\boldsymbol{x}_0$, representing the initial location of an agent. The velocity field $V(\boldsymbol{x})$ signifies the permissible speed of vehicles at a given position while considering the proximity of obstacles (such as walls and barriers) to the agents. As part of the FMM's initialization step, every grid point located on obstacles is initially labeled as *accepted*.

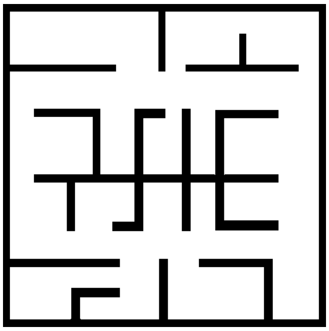


FIGURE 2. An example of binary occupancy map. The binary image, which is on 512 × 512 pixel size, takes the value of 0 if the position is occupied by obstacles, and 1 otherwise.

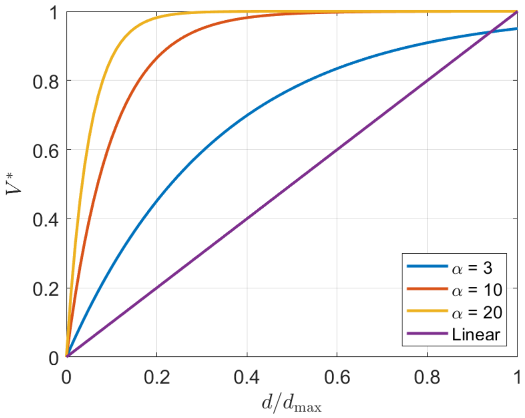
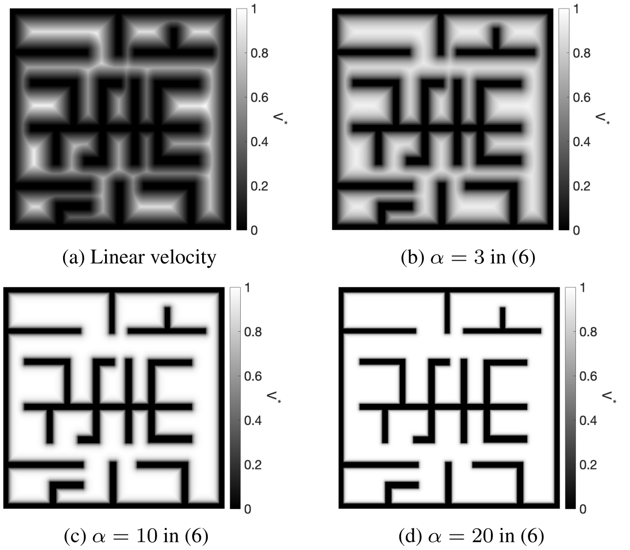


FIGURE 3. Plots of velocity functions (6) as a function of normalized distance $d/d_{\max}$ for different values of α. The vertical axis $V^*(= V/\mathcal{V}_{\max})$ is a normalized velocity by the maximum speed. In general, a smaller value of α results in a larger imposed safety distance.

Next, the FMM algorithm is executed to compute the shortest time $T(\boldsymbol{x})$ for the propagating wave to arrive at each grid point. The trajectory of the agent is finally determined by extracting the maximum gradient direction of $T(\boldsymbol{x})$ from the target point to the initial point. Since $T(\boldsymbol{x})$ is derived from the target point, the resulting T-field uniquely exhibits a single minimum at the target point, ensuring a unique solution [31].

A remaining task is to employ an appropriate model for the velocity field $V(\boldsymbol{x})$ that respects the environment. While one might simply consider the simplest option, which is to use a constant value $\mathcal{V}_{max}$ representing the maximum speed of the agents, it is observed that the resulting trajectory lacks realism as it fails to ensure both smoothness and a safe distance between agents and obstacles [31]. To address these issues, the FMM has been advanced into the Fast Marching Square (FMS) method. In order to guarantee a safe distance between obstacles and agents, this approach introduces a penalty to the agent's velocity as it navigates in proximity to obstacles. The FMS method entails the implementation of two distinct FMMs.



(a) Linear velocity

(b) $\alpha = 3$ in (6)

(c) $\alpha = 10$ in (6)

(d) $\alpha = 20$ in (6)

FIGURE 4. Comparison of velocity maps generated from the different velocity forms shown in Fig. 3. Sharper increase of V^* to value 1 results in a larger safety distance.

The objective of the first FMM is to construct a velocity grid map that takes into account the presence of obstacles. This objective is achieved by evolving initial surfaces, which represent the boundaries of obstacles in the environment, with a constant velocity $V(\boldsymbol{x}) = 1$. The outcome of this process is the computation of the distance $d(\boldsymbol{x}) = T(\boldsymbol{x}) \in \mathbb{R}^+$ at each grid point, indicating the shortest distance to the nearest obstacle. Consequently, a velocity grid map V is computed as a function of d, which is artificially designed to penalize the vehicle's speed as it approaches obstacles. A common choice for this penalizing function is a linear relationship $V \propto d$, which is inspired by a two-dimensional repulsive electrostatic potential [42].

Alternatively, one may also consider

$$V(d(\boldsymbol{x})) = \mathcal{V}_{\max}\left[1 - \exp\left(-\alpha\left(\frac{d}{d_{\max}}\right)\right)\right], \qquad (6)$$

where $d_{\max}$ is the maximum distance in the configuration space and $\mathcal{V}_{\max}$ is the maximum speed of the agent at free space, respectively. Note that the form (6) includes a dimensionless free parameter α that indirectly governs the safety distance. Fig. 3 shows the plots of the velocity function profiles at several values of α. The corresponding velocity maps for those α values in Fig. 3 are shown in Fig. 4 to visualize the impact of α values on the collision safety distance. The velocity map created from the binary map (Fig. 2), using the form (6) with $\alpha = 3$, is shown in Fig. 5.

Next, the second FMM is executed from the initial position $\boldsymbol{x}_0$ of the agent (or vehicle) to compute the time grid map $T(\boldsymbol{x})$, respecting the environmental constraints through $V(\boldsymbol{x})$. Finally, the path is obtained again by applying the gradient descent algorithm to $T(\boldsymbol{x})$ and the resulting path for the example case shown in Fig. 6. The strength of the FMM-based method lies in its unparalleled computational speed when dealing with specific types of optimization problems. For instance, to provide a more intuitive grasp of the computational efficiency inherent in FMM-based methods, we delve into some practical specifics. The process of extracting a path from a grid of size 10^7 typically demands only a matter of seconds when employing a single-core machine. To put this into a simpler perspective, it is comparable to handling a two-dimensional pixel image measuring 4000 × 4000 in dimensions. It is also noteworthy that the application of the FMM across multiple iterations does not burden the optimization process with any substantial computational time constraints. The efficiency of the FMM-based method inspires the development of a new framework for various scenarios of modern operations of uncrewed vehicles in the subsequent section.

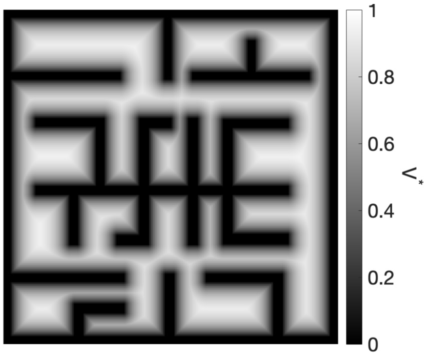


FIGURE 5. Velocity map created using the form (6) with $\alpha = 3$. The velocity values V^* are normalized with the maximum speed of agent $\mathcal{V}_{\max}$.

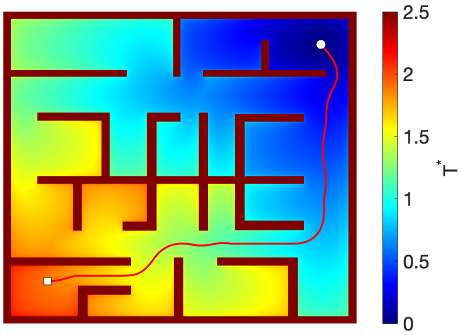


FIGURE 6. The optimized path after applying the gradient descent algorithm is plotted on the time grid. The white circle denotes the start point, while the square indicates the endpoint.

III. FMM-BASED RENDEZVOUS PATH PLANNING FOR A TEAM OF HETEROGENEOUS VEHICLES

The goal of this section is to introduce an innovative approach to leveraging the FMM-based method within the multi-agent path planning domain. In particular, we propose an FMM-based rendezvous path planning algorithm designed for a diverse team of vehicles. The team is tasked with efficiently converging at a single location, aiming for optimal efficiency in pursuit of a general goal.

A. PROBLEM STATEMENT

This paper considers the problem of finding paths for $N(\geq 2)$ heterogeneous vehicles in a team, which are tasked with rendezvousing within minimal time. The region of interest Ω is assumed to be represented by an occupancy grid map, where each pixel is either free $\mathcal{C}_{\text{free}}$ or occupied $\Omega\backslash\mathcal{C}_{\text{free}}$. According to [43], a path is viewed as a continuous function $\tau : [0, 1] \rightarrow \mathcal{C}_{\text{free}}$, in which each point along the path is given by $\tau(s)$ for some $s \in [0, 1]$. Here, $\tau(0)$ corresponds to the starting point of the agent whereas $\tau(1)$ denotes the target point. Although the orientation of each vehicle is not considered in this work, it is also feasible to incorporate their orientations using the existing methodology [39].

We assume that the starting position $\tau^i(0)$ of each vehicle in the team is given. Note that we introduced the index $i = 1, \ldots, N$ to denote each vehicle. Then, the rendezvous path planning for the team is divided into two sub-problem. The first problem is to determine the optimal rendezvous point $\boldsymbol{x}_m$ such that

$$\boldsymbol{x}_m = \underset{x\in\mathcal{C}_{\text{free}}}{\arg\min}\, \mathcal{F}(\boldsymbol{x}), \tag{7}$$

where $\mathcal{F}$ is a general cost function. The second sub-problem is to determine the optimal path τ^i from the initial point $\tau^i(0)$ of each agent to the optimal point $\tau^i(1) = \boldsymbol{x}_m$.

From now on, for the purpose of illustration, we fix the optimizing function $\mathcal{F}$ as the meeting time. In rendezvous tasks, this function corresponds to the arrival time of last agent, which is written as

$$\mathcal{F}(\boldsymbol{x}) = \max\left[T^1(\boldsymbol{x}), T^2(\boldsymbol{x}), \ldots, T^N(\boldsymbol{x})\right]. \tag{8}$$

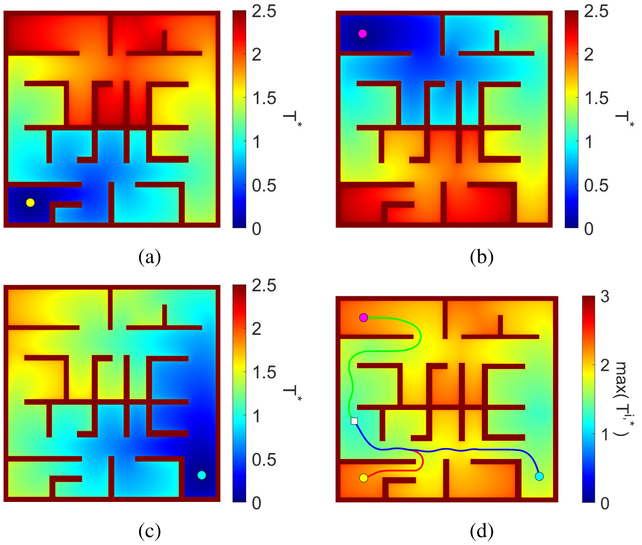
where $T^i(\boldsymbol{x})$ denotes the arrival time for all $\boldsymbol{x} \in \mathcal{C}_{\text{free}}$, which will be also referred to as a time grid from now on.

B. THE ALGORITHM

Now, we describe our approach to the aforementioned rendezvous path planning problem. Considering the different initial positions of each agent, a single implementation of the FMS method yields $T^i(\boldsymbol{x})$ for all points in Ω. During this step, one can consider the arrival time $T^i(x)$ of each agent can be determined considering the different velocities and the safe distances imposed by environments. Once the arrival time maps for all agents are prepared, the time-optimal meeting point $\boldsymbol{x}_m$, which minimizes the cost $\mathcal{F}(\boldsymbol{x})$ can be conveniently determined by

$$\boldsymbol{x}_m = \underset{x\in\mathcal{C}_{\text{free}}}{\arg\min}\left(\max\left[(T^1(\boldsymbol{x}), T^2(\boldsymbol{x}), \ldots, T^N(\boldsymbol{x})\right]\right) \tag{9}$$

The rest of this section provides the implementation details of the presented approach using an example of rendezvous planning for three agents, which are initially located at three different corners of a given binary occupancy map previously shown in Fig. 2. The initial positions are shown in Fig. 7a, Fig. 7b, and Fig. 7c. For simplicity, we assume that the vehicles are identical, which means that the vehicles travel with the same dynamics and at the same and constant speed. Specifically, $\alpha = 3$ and $\mathcal{V}_{\max} = 1$ are used in the example.



(a)

(b)

(c)

(d)

FIGURE 7. (a-c) The normalized arrival time T^* maps for three agents located at different initial points. (d) The optimized path drawn on $\mathcal{F}(x)$ as defined in the form (8).

The algorithm first begins by following the standard step of the FMS method to measure the distance $d \in \mathbb{R}^+$ to the nearest obstacles at every point in the grid. The first FMM runs from the initial surfaces of obstacles to fill in d-values on every non-occupied point in $\mathcal{C}_{\text{free}} \subset \Omega$, using the uniform velocity $V(\boldsymbol{x}) = 1$. Next, we generate a velocity map $V^i(\boldsymbol{x})$ for each agent $i \in \{1, 2, \ldots, N\}$ using the velocity function (6). Each agent may have a different value of safety parameter α and the maximum allowable speed $\mathcal{V}_{\max}$. The velocity map for the binary occupancy map using the form is shown in Fig. 5.

Then, we run the second FMM multiple times starting from each initial position of the agent $\boldsymbol{x}_0^i$, which corresponds to $\tau^i(0)$. This second round of FMM computation is executed to propagate a source wave point located at the target point until the arrival time T value at the initial point is determined. At each iteration, we obtain the arrival time map $T^i(\boldsymbol{x})$ for each agent. Once the iterations of the second FMM are complete, the optimal point $\boldsymbol{x}_m$ can be determined directly from the form (9). The result of the term in (9), $\max\left[(T^1(\boldsymbol{x}), \ldots, T^N(\boldsymbol{x})\right]$, is shown with a color map in Fig. 7d.

Lastly, the optimized path τ^i for each agent to the rendezvous point is determined by applying the gradient descent algorithms to the time grid $T^i(\boldsymbol{x})$. This trajectory optimization step is inferred from the maximum gradient direction of $T(\boldsymbol{x})$. The final outcome of the FMS method is the optimized continuous path τ, a collection of point in Ω that guides trajectory of agents as shown in Fig. 6. The procedure is summarized in Algorithm 1. The corresponding flowchart is shown in Fig. 8, which visualizes the two-step FMM procedure more clearly. There are various repositories

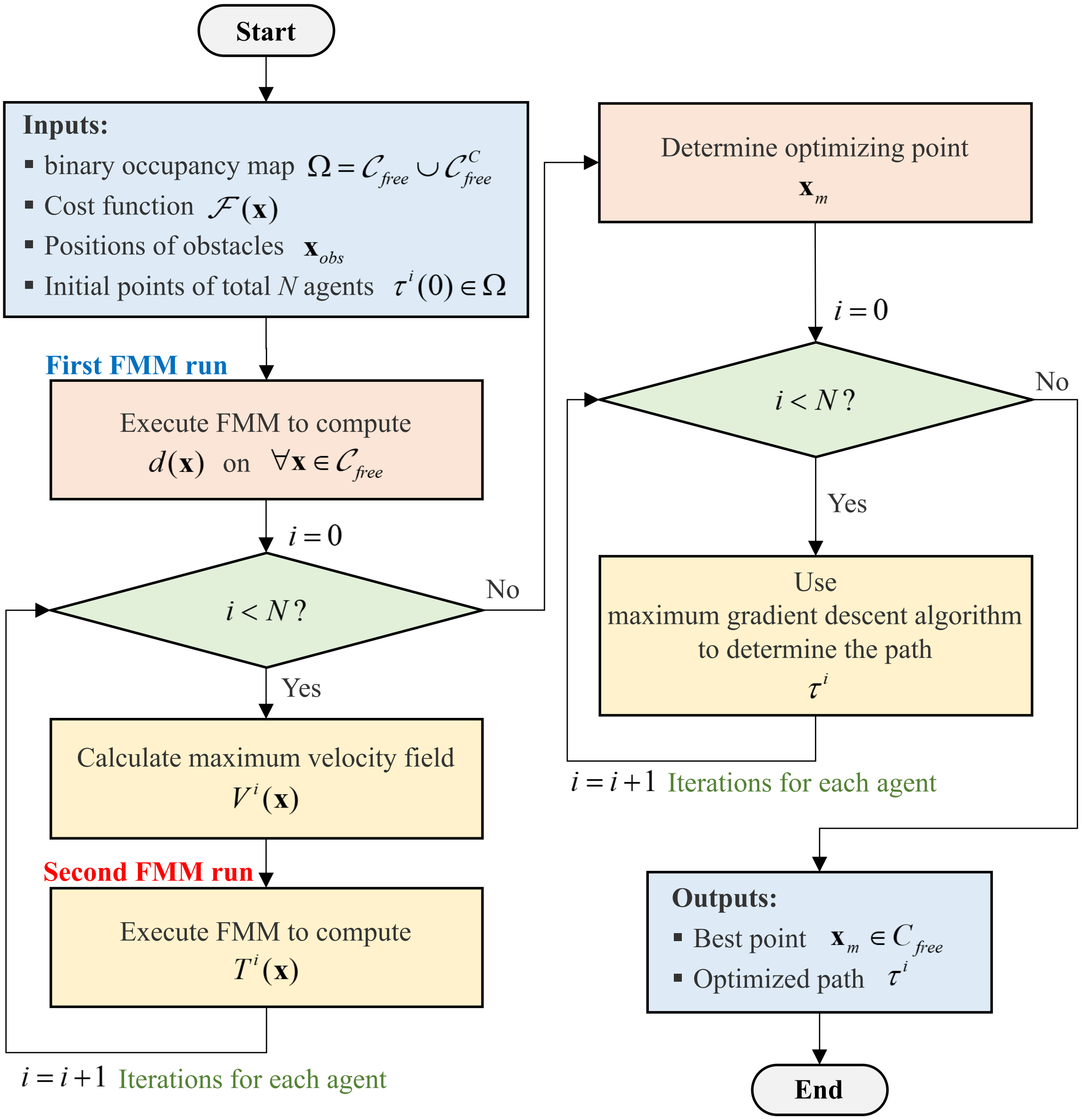

Start
Inputs:
binary occupancy map $\Omega = \mathcal{C}_{free} \cup \mathcal{C}_{free}^{C}$
Cost function $\mathcal{F}(\mathbf{x})$
Positions of obstacles $\mathbf{x}_{obs}$
Initial points of total N agents $\tau^i(0) \in \Omega$
First FMM run
Execute FMM to compute $d(\mathbf{x})$ on $\forall \mathbf{x} \in \mathcal{C}_{free}$
$i = 0$
$i < N$?
No
Yes
Calculate maximum velocity field $V^i(\mathbf{x})$
Second FMM run
Execute FMM to compute $T^i(\mathbf{x})$
$i = i + 1$ Iterations for each agent
Determine optimizing point $\mathbf{x}_m$
$i = 0$
$i < N$?
No
Yes
Use maximum gradient descent algorithm to determine the path τ^i
$i = i + 1$ Iterations for each agent
Outputs:
Best point $\mathbf{x}_m \in C_{free}$
Optimized path τ^i
End

FIGURE 8. Flow chart for Algorithm 1, the FMM-based path optimization for multi-agent rendezvous tasks. The algorithm consists of two FMMs. The objective of the first FMM is to construct a velocity grid map that takes into account the presence of obstacles, while the second FMM computes the time grid map $T(x)$, respecting the environmental constraints through $V(x)$.

that can be used to implement Algorithm 1 in various programming languages. In our study, the code is developed using a GitHub library source [44], which is written in C++ and relies on Boost libraries [45].

C. ALGORITHM EFFICIENCY AND ADAPTABILITY

In this section, we test the performance of the suggested algorithm under various conditions. This section includes scalability tests in terms of the number of grids (pixels) and the number of agents in the system. In addition, we demonstrate the additional potential of the present algorithm, which can adapt the path on the fly to reflect environmental changes. All performance tests in this section were carried out on a single 1.6 GHz core with 8 GB of RAM, and we recorded the wall-clock time required to complete one full time step for the two methods.

First, to measure the performance of the algorithm, we conducted the same task (i.e., the rendezvous mission of three agents demonstrated in Fig. 7), but with different grid sizes. Note that although we did not alter the shape of the map during this test, the increase in the number of grids reflects the size of the environment that can be accessed while maintaining the same resolution. The result is summarized in the format of a log-log scale plot in Fig. 9. At the highest grid numbers ($4096^2 = 16, 777, 216$),

Algorithm 1 FMM-Based Algorithm of Path Optimization for Multi-Agent Rendezvous Tasks

```
Input: A binary occupancy map $\Omega = \mathcal{C}_{\text{free}} \cup \mathcal{C}^c_{\text{free}}$, a cost function $\mathcal{F}(\boldsymbol{x})$, positions of obstacles $\boldsymbol{x}_{\text{obs}}$ and initial points of total $N$ agents $\tau^i(0) \in \Omega$
Output: The best point $\boldsymbol{x}_m \in \mathcal{C}_{\text{free}}$ that optimize the cost $\mathcal{F}$, and the optimized path $\tau^i$ for each agent
Execute the fast marching method from $\partial\mathcal{C}^c_{\text{free}}$ to compute the minimum distance $d(\boldsymbol{x})$ from obstacles.
for $i = 1$ to $N$ do
    Calculate the maximum velocity field $V^i(\boldsymbol{x})$ using $d(\boldsymbol{x})$.
    Execute the fast marching method from $\boldsymbol{x}^i_0$ to compute $T^i(x)$.
end for
Determine the optimizing point $\boldsymbol{x}_m$ that minimizes $\mathcal{F}(\boldsymbol{x})$.
for $i = 1$ to $N$ do
    Use the maximum gradient descent algorithm to determine the path $\tau^i$ from $\boldsymbol{x}^i_o$ to $\boldsymbol{x}_m$.
end for
```

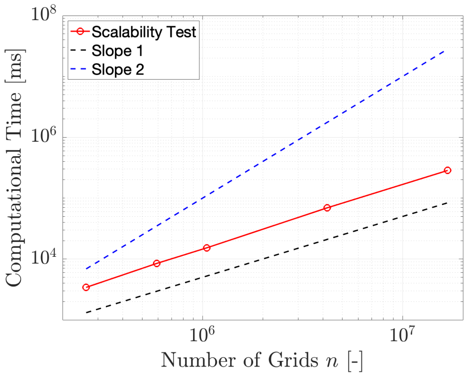
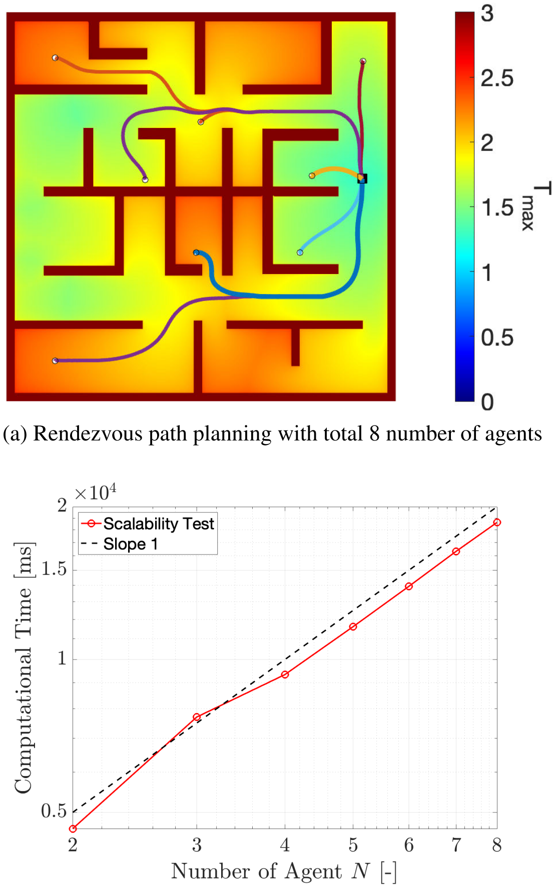


FIGURE 9. Computational time measured at different grid sizes.

the actual time it takes is approximately 285 s. While the theoretical (and conservative) computational complexity is $\mathcal{O}(n \log n)$, in practice we achieve a better scale, approaching $\mathcal{O}(n)$, which appears as a line with slope 1 in the log-log scale plot. The results suggest that the current method can be readily extended to accommodate a larger environment.

Next, we also examine how the number of participating agents in the rendezvous task affects the computational time. For example, Fig. 10a demonstrates the rendezvous point and path of each agent for a team consisting of 8 agents. While the resulting path planning for all 8 agents seems non-trivial, the increase in the number of agents does not significantly complicate our framework, and the computational time increases only linearly with the number of agents, $\mathcal{O}(N)$, as illustrated in Fig. 10b. Moreover, because the computations for each agent are independent, the algorithm can be readily parallelized for further speedup.

Before concluding this section, we briefly show an additional adaptability of the proposed framework for online path planning, leveraging the marginal computation cost (approximately 3 seconds on a 512 × 512-sized regular grid). Specifically, we consider a case where the positions of obstacles and target points are changing over time. Here, we will limit our discussion to cases where the time scale of agents traveling in the domain is much larger than that of the target and environmental changes. In the following, we assume that the positions of agents and moving obstacles are tracked by a system at every time interval δt.

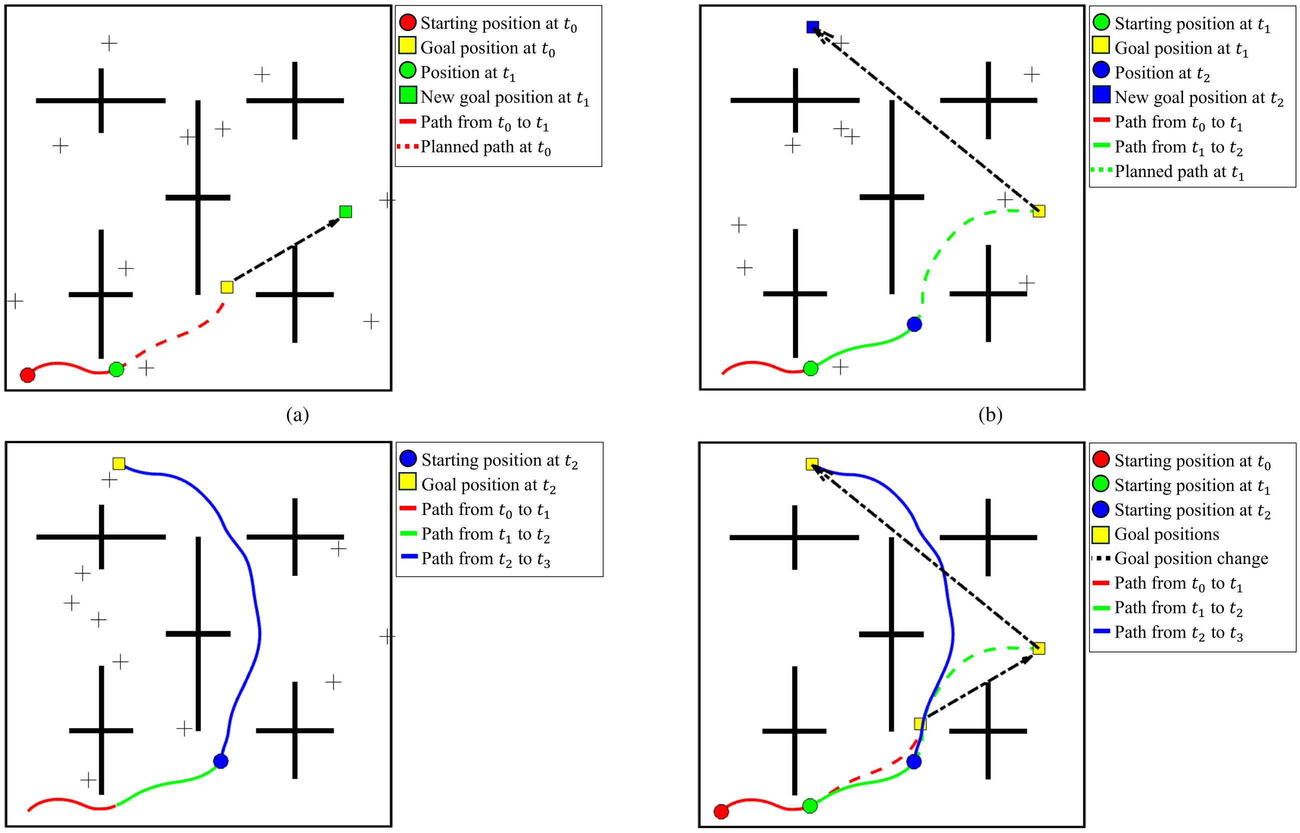


(a) Rendezvous path planning with total 8 number of agents

(b) Computational time measured with different numbers of agent

FIGURE 10. Scalability test results on the number of agents.

For demonstration, consider an initial path planning using the FMM approach illustrated in Fig. 11a. The initial positions of the agent and target are denoted by the red circle and yellow square, while the black cross markers represent the positions of moving obstacles. After a time interval of δt, we assume that when the agent arrives at the green circle, the target has moved to the green square. If such environmental changes are recorded, a new FMM-based path planning algorithm, resulting in Fig. 11b, is implemented to reflect the new positions of the target and obstacles. Fig. 11c illustrates an additional path planning for the third time interval, spanning from $2\delta t$ until the goal position is reached. Fig. 11d summarizes the sequence of path planning occurring



(a) (b) (c) (d)

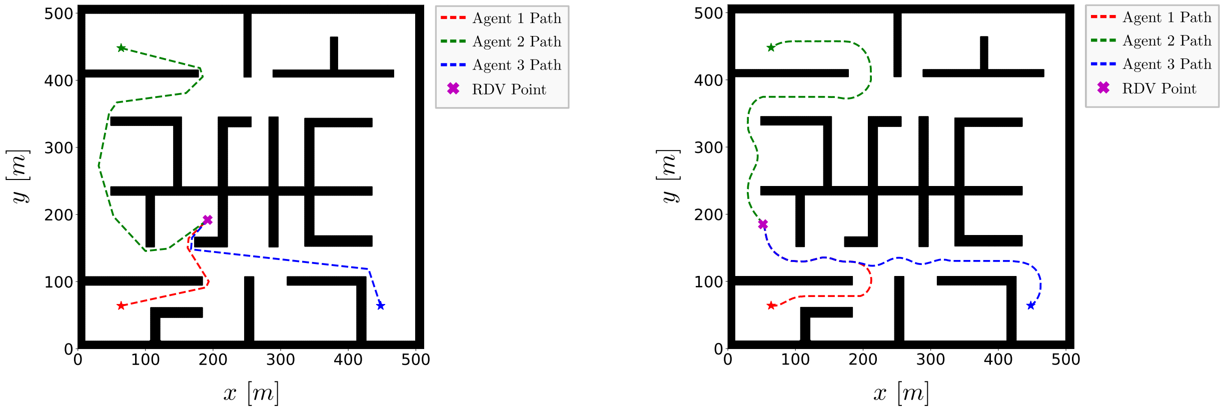
FIGURE 11. An example of semi-discretized online path planning using the FMM method. (a) The initial path planning at the first time interval $[t_0, t_1]$. The red circle and yellow square denote the initial positions of the agent and target point respectively. When the positions of the target and obstacles (black cross points) are updated after $\delta t = 200$ [s], a new path is generated. (b) and (c) demonstrate path planning for the second and third time interval $[t_1, t_2]$ and $[t_2, t_3]$. (d) Summary the sequence of path planning occurring over three time intervals. In this example, $t_0 = 0$, $t_1 = \delta t$, $t_2 = 2\delta t$, and $t_3 = \infty$ are used.

over three time intervals. The results show that online path planning, which respects environmental variations, can be addressed in a semi-discretized manner due to the extreme computational efficiency of the FMM-based method.

Finally, we note that the present approach did not explicitly consider coordination and communication, assuming that the problem's scale was large enough to focus solely on finding the rendezvous point and paths. Specifically, agents' coordination can be considered in detail by extending the implementation to treat other agents as obstacles during the planning stage, as illustrated in Fig. 11. Communication was assumed to be available throughout the operation, meaning synchronous rendezvous is considered. The method can be implemented in either a centralized or decentralized manner. On the other hand, we admit that the approach would fail to find the time-optimal rendezvous path if each agent does not recognize the location of other moving agents. Such a situation may occur with only asynchronous communication or, worse, when communication is completely unavailable. If this is the case, we expect that approximated locations (e.g., via dead reckoning) might be utilized to determine the rendezvous point and paths.

D. BASELINE COMPARISON

The current approach is compared with a baseline algorithm for evaluation. To the best of our knowledge, as described in detail in Section I, existing algorithms cannot be directly applied to the problem considered in this paper. Specifically, scheduling algorithms primarily focus on determining the sequence of heterogeneous agents, rendezvous search algorithms concentrate on identifying the rendezvous position under limited communication, and multi-agent path finding algorithms are chiefly concerned with collision avoidance among agents. Therefore, we construct a baseline algorithm by combining two methods from different objectives: the weighted centroid point [46] for determining the rendezvous point and the optimal rapidly-exploring random tree (RRT*) [47] for path planning. The inverse of each agent's velocity is used as a weight for the centroid calculation, while the RRT* is implemented using the Open Motion Planning Library (OMPL) [48] for C++ implementation. The computational complexity of the baseline method is also comparable to the present scheme: the RRT* algorithm has a computational complexity of $\mathcal{O}(\mathrm{K} \log \mathrm{K})$, where K is the number of nodes in the tree.



(a) Rendezvous path planning result from RRT*

(b) Rendezvous path planning result from the presented FMM-based approach

FIGURE 12. **Comparison of the presented FMM-based rendezvous path planning method to a baseline method consisting of a weighted centroid method to find the rendezvous point (RDV Point) and the RRT* algorithm (for pathfinding).**

Fig. 12 illustrates the comparative analysis between the baseline method and the current FMM-based path planning approach using the same example problem considered in Fig. 7. The rendezvous point and agents' paths computed by the baseline method are shown in Fig. 12 (a), where it takes 574.29 seconds for all agents to reach the rendezvous. The corresponding results from the FMM method are shown in Fig. 12 (b), where it takes 626.27 second for all the agents to reach the rendezvous point.

At first glance, the advantage of our approach may not seem clear compared to the baseline method. However, several points must be considered. First, the RRT* algorithm does not employ any path smoothing operations, so the constructed path may not be convenient depending on the vehicle's kinematics, whereas the FMM path is smooth. Second, the weighted centroid method (for determining the rendezvous point in the baseline approach) does not guarantee that the rendezvous point $\boldsymbol{x}_m^{cp}$ remains within $\mathcal{C}_{\text{free}}$. Lastly, if domain Ω is more complex, resembling a maze, the optimality of the rendezvous point suggested by the weighted centroid method becomes more questionable.

IV. NUMERICAL EXPERIMENT

In this section, we conduct a numerical experiment to showcase an application of the suggested method in more realistic cases. We consider a virtual scenario of a rendezvous task for a team of heterogeneous vehicles. The experimental setting is as follows.

A. EXPERIMENTAL SETUP

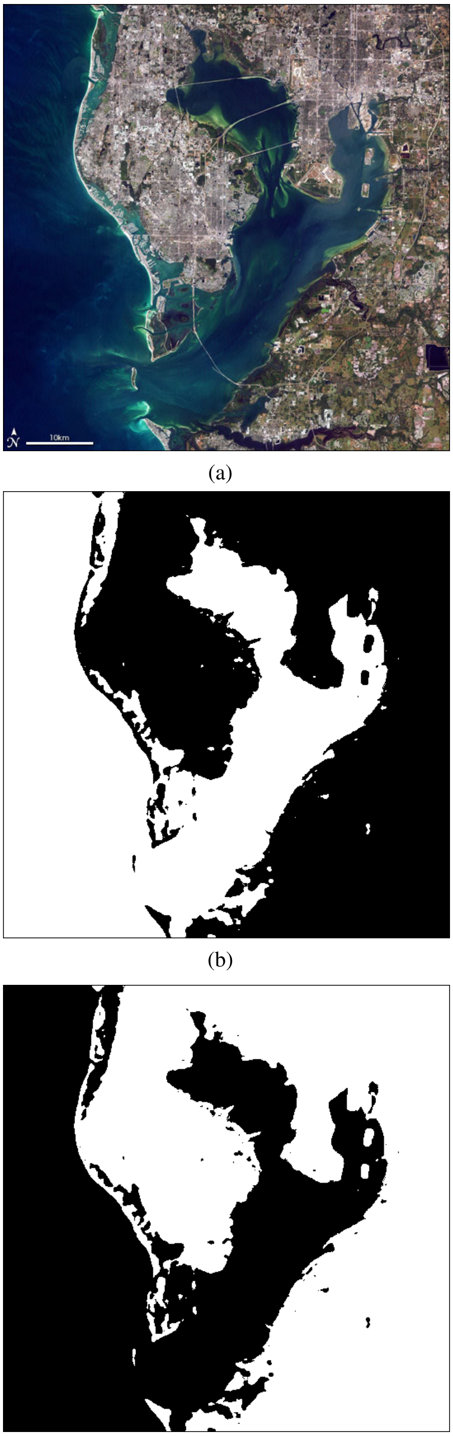
First, we create a computational domain to simulate a realistic environment. The Tampa Bay area is chosen as our test domain. We use a satellite image from NASA's Earth Observatory (as shown in Fig. 13a).[2] Then, the GRIP tool (*Graphically Represented Image Processing engine*) [49] is employed to convert the satellite image into a binary configuration space map. The primary objective of image processing at this stage is to distinguish water bodies and land areas, as illustrated by white and black pixels, respectively in Fig. 13b.

Next, we build a team of heterogeneous agents, consisting of four types of vehicles: an uncrewed underwater vehicle (UUV), an uncrewed surface vehicle (USV), an uncrewed ground vehicle (UGV), and an uncrewed aerial vehicle (UAV). The UUV operates exclusively underwater but is limited by operational depth constraints. Consequently, UUV operations are required to take place at a considerable distance from the shoreline. On the other hand, the USV is designed for slower mobility, but it has the capability to navigate areas closer to the coastline. In contrast, the UGV's operational domain is limited to land. Lastly, the UAV, being an aerial platform, is assumed to move at a constant speed without encountering any obstacles.

Operational constraints for the aforementioned heterogeneous agents are addressed using their respective velocity maps, $V^i(\boldsymbol{x})$. The primary tools are the magnitude of the penalty parameter α in (6) and mirroring of the binary image. To begin with, it is reasonable to impose a higher penalty to the operating velocity of UUV in the proximity to land, since UUV is required to operate at a far distance from the shoreline. Thus, we set the penalty parameter to $\alpha^{uuv} = 100$, while the values of $\alpha^{usv}, \alpha^{ugv}$ are set to 3. Moreover, in order to address the specific land travel limitation of the UGV, the operation domain of the UGV is obtained by the mirroring of the binary operational domain of ocean vehicles (i.e. USV, UUV), the result of which is shown in Fig. 13c. For UAVs traveling above both water and land, their domain is considered as free space without obstacles.

The remaining parameters are the maximum operational speed $\mathcal{V}_{\max}^i$ of each vehicle. In actual applications, these parameters should reflect the actual performances of agents. In this virtual test, we assume the following scenario to

[2] https://earthobservatory.nasa.gov/images/4745/tampa-bay-florida



(a)

(b)

(c)

FIGURE 13. (a) Satellite image of Tampa Bay, FL. (downloaded from NASA Earth Observatory) (b) Processed binary image from Fig. 13a for the USV and UUV; (c) Processed binary image for UGV, which is an inverse of Fig. 13b.

demonstrate the full potential of the present approach. First, the maximum operational speed of UGV is assumed to be the slowest among all agents, considering cases where UGVs need to move as a group or encounter additional environmental restrictions (such as traffic or changes in topography). Then, we normalize the velocities of the vehicles using the maximum speed of UGV, and thus we write $\mathcal{V}^{ugv}_{\max} = 1$. The maximum speeds of USV and UUV are set to the same value $\mathcal{V}^{uuv}_{\max} = \mathcal{V}^{usv}_{\max} = 2$, and the UAV is assumed to have the highest navigation speed and set to $\mathcal{V}^{uav}_{\max} = 3$. With the prescribed setting, we apply Algorithm 1 to solve the optimization problem of rendezvous path planning.

B. RESULTS

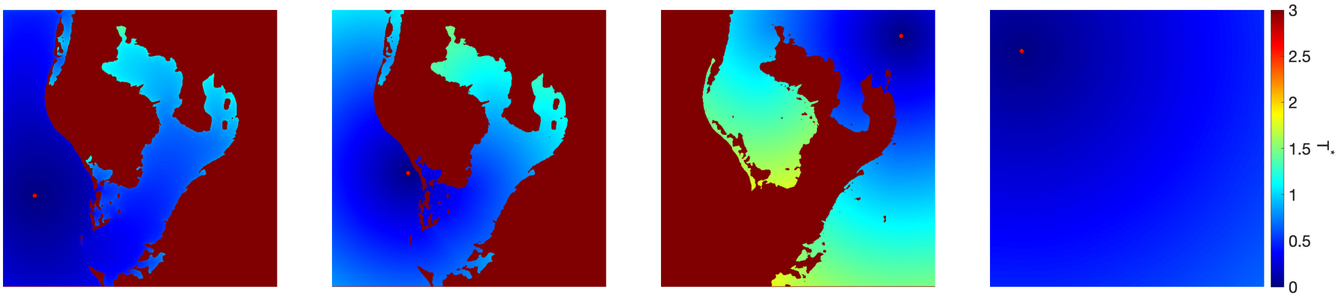
We execute the first FMM (of the FMS) for each vehicle from arbitrarily selected initial points, as seen by the red dots in Fig. 14a to Fig. 14d. The computed time grid $T^i(\boldsymbol{x})$ for each vehicle is also visualized in the same plots. One distinguished case is Fig. 14d which shows unimpeded paths for the UAV throughout the environment.

Next, we describe an additional procedure that is not stated in Algorithm 1, but is required for the present problem, wherein the UGV operates on the complementary domain of the UUV and USV. Note that candidates for the rendezvous point are located on the shoreline, which belongs to the obstacles (i.e. $\Omega \backslash \mathcal{C}_{\text{free}}$) in the first FMM for any agent. To address this issue, we extend the time grid $T^i(\boldsymbol{x})$ to the edges of obstacles. Grid cells on the edge of binary images (Fig. 13b and Fig. 13c) are first detected using MATLAB's "edge" function, and the $T^i(\boldsymbol{x})$ on edges are inferred from the minimum time value among the adjacent grid points. Fig. 15 illustrates this procedure. This process results in a subset of the shoreline emerging as the candidate for the rendezvous point as seen in Fig. 16. Then, in the same figure, the black dot, which represents the optimal rendezvous point where all vehicles can converge in the minimum time, is determined by the form (9).

We also note here that we still need to compute the full arrival time map in $\mathcal{C}_{\text{free}}$. The reason is because the ultimate outcome of the present algorithm is not only to determine the rendezvous point but also to optimize the path for each agent. The latter necessitates computing arrival times for every grid point between the initial point and the rendezvous point.

Overall, the results show that the resulting paths in Fig. 17 align well with the assumed operational constraints of the member vehicles, offering realistic planning outcomes. While the paths include overlaps between the USV and UGV, path conflicts are not an issue in the test scenario because the agents operate in distinct domains; they would not collide even if their paths overlap at the same time.

Next, the algorithm computes a path for each vehicle using the gradient descent method. Fig. 18a illustrates the UUV's trajectory, characterized by significant turns to remain within its operational range, attributable to the low alpha value. In contrast, Fig. 18b exemplifies the USV's efficiency in navigating between islands. The UGV's path in Fig. 18c remains confined to land, adhering to its intended operational domain. Lastly, the paths for all vehicles are summarized



(a) (b) (c) (d)

FIGURE 14. The arrival time grids $T^i(x)$ of the (a) UUV, (b) USV, (c) UGV, and (d) UAV. The red dot in each plot denote the initial position of the vehicle.

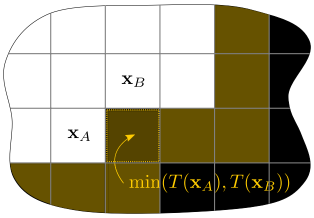


FIGURE 15. An illustration that outlines procedures for extending the time grid to accommodate situations where vehicles operate in non-intersecting domains, specifically the UUV/USV and UGV.

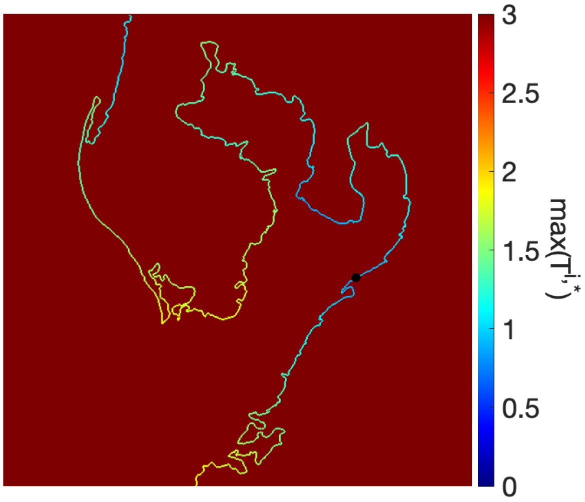


FIGURE 16. An extended time grid defined on shoreline to compute the rendezvous point x_{op}, following the form (9).

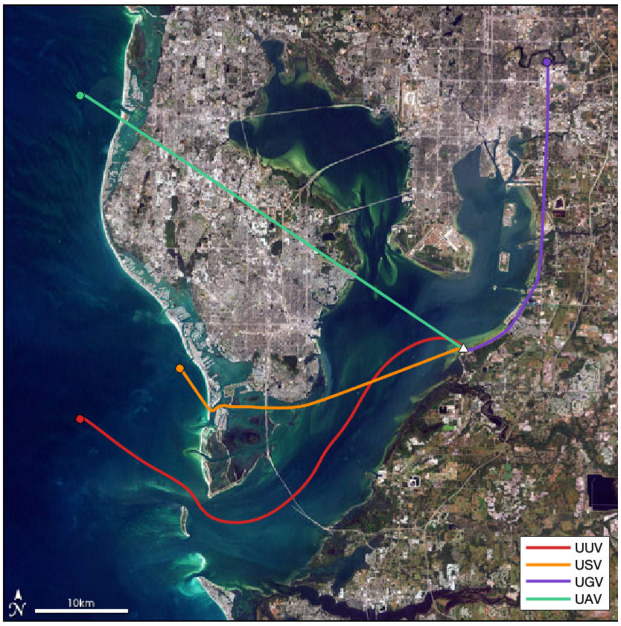


FIGURE 17. Path planning results plotted over the original satellite image. Circles denote starting points of vehicles, and the triangle denotes the computed optimal rendezvous point from the presented algorithm.

in Fig. 17, which also includes the simplest UAV's path unhindered by obstacles due to its aerial capabilities.

V. DISCUSSION

In this section, we highlight the merits of our methodology in various perspective and analyze the limitations. First, the numerical experiment presented in Section IV demonstrates how the proposed path planning method can be applied to real-world situations that demand rapid and coordinated action across various domains. For example, in disaster response scenarios, the method efficiently coordinates rescue teams composed of diverse vehicles, ensuring rapid assembly at critical locations to save lives and deliver aid. Environmental monitoring can benefit from this algorithm by synchronizing different types of sensors to converge on pollution sources or wildlife areas, facilitating comprehensive data collection. In military operations, the algorithm enhances the strategic coordination of diverse units, allowing for swift assembly and execution of missions with heightened precision and safety.

The authors are aware that constructing collision-free paths for agents is one of the most critical aspects of MAPF, even though our test scenario was free of such conflicts. However, even if agents share the same operational domain, our framework makes it easy to check whether overlapping paths will lead to collisions. This is because we have explicit information of each vehicle's arrival time at overlapping points; path conflicts occurring at different times do not result in collisions between agents. This is useful

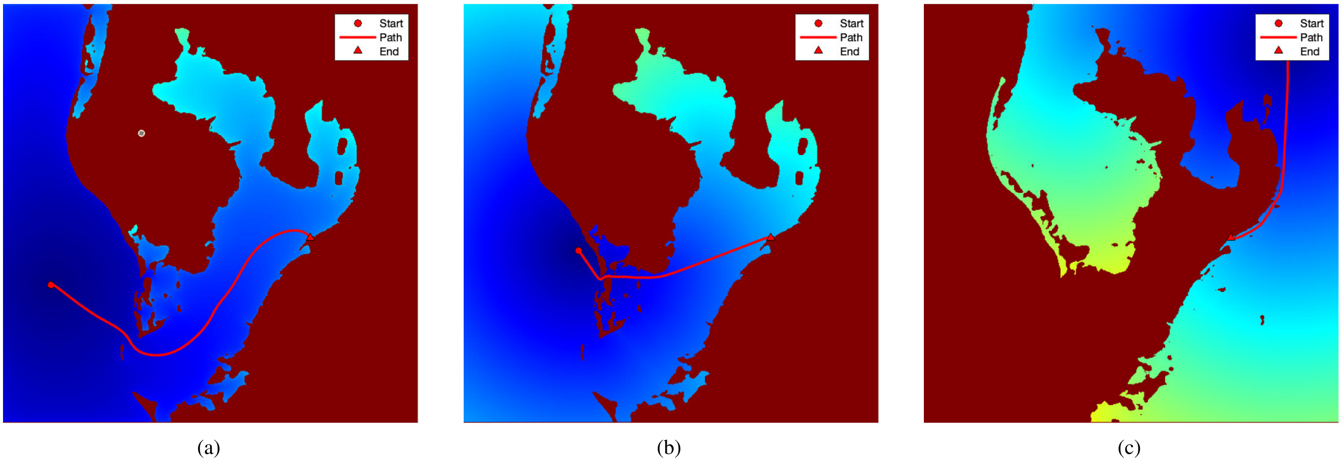


FIGURE 18. The paths planned for the (a) UUV, (b) USV, (c) UGV, shown over the each time grids Fig. 14.

information for extending the current framework to achieve truly collision-free path planning.

We re-emphasize that the main contribution of the present work is the formulation of the multi-agent rendezvous problem in the form of (8). While Algorithm 1 is straightforward under this formulation, a potential improvement can be found in the design of the velocity function (6) to consider more detailed and realistic operational constraints of various types of vehicles.

The main advantage of our approach is that the process is deterministic, implying that the resulting paths are guaranteed to be the globally optimized solution. In addition, the computational cost for determining paths of all agents is only proportional to the number of agents N. Therefore, at least for the rendezvous task as defined in (8), we claim that our approach outperforms heuristic, stochastic, and machine learning-based methods in terms of providing a unique solution and scalability.

Although optimization for energy efficiency is not within the scope of this work, we expect that it can be considered, at least in a limited sense, by modeling the velocity field to incorporate physical conditions. For example, a drone's most energy-efficient path can vary depending on wind conditions, while a UUV's most energy-efficient path can vary depending on ocean currents. Such environmental conditions can be modeled into the operating velocity grid by adding the adverse effect of the ocean current or wind. However, such approach is still restricted to the shortest path corresponds to the most efficient path, since the FMM-based method is inherently grounded on a minimum-time path finding problem. Therefore, a new formulation would be necessary to determine the most energy-efficient path. Developing such a formulation and methodology require to include vehicle dynamics and control inputs as important constraints. This related but distinct problem formulation and solution approach will mark a significant advancement in FMM-based path planning algorithms.

VI. CONCLUSION

Recent rapid advancements in uncrewed vehicle technology have significantly improved accessibility and cost-effectiveness, leading to their widespread integration across various domains, including ground, water, and air. As systems with uncrewed vehicles become ubiquitous, the demand for sophisticated navigation methodologies that can efficiently guide their interactions also becomes paramount. In this regard, the present work introduces a new approach to path planning for multi-agent systems. Our method is rooted in the well-established framework of the FMM. The methodology presented in this paper leverages the capabilities of the FMM to efficiently optimize trajectories for heterogeneous teams of agents, augmenting their operational efficiency and collective synergy.

To illustrate our approach, we consider an example path planning scenario involving four different types of uncrewed vehicles navigating around the Tampa Bay area. The results of the virtual experiment have demonstrated how the path planning task of a multi-agent system can benefit from the effectiveness of the FMM-based method, which conveniently incorporates the individual operational characteristics of the heterogeneous vehicles. The computational efficiency and flexibility of our approach open the door to various directions for future work.

- The optimization function $\mathcal{F}$ can also be extended to incorporate various scenarios of rendezvous tasks beyond minimal time. For example, we plan to include different operational costs for heterogeneous vehicles to maximize the economic efficiency of rendezvous tasks.
- The proposed framework can be extended to path planning in the presence of dynamic obstacles. This generalization will allow the algorithm to consider the collision between different agents. The future work will investigate how the fast marching method can be modified in order to efficiently incorporate moving objects or other moving agents in the computation.

- Finally, we also envision extending our framework to different purposes of path planning for heterogeneous agents beyond rendezvous missions. This extension could involve group search optimization and assignment tasks.

ACKNOWLEDGMENT

The authors would like to thank Blake Sanders for his help with creating Figures 13(b) and 13(c).

REFERENCES

[1] J. Shin, S. Chang, J. Weaver, J. C. Isaacs, B. Fu, and S. Ferrari, "Informative multiview planning for underwater sensors," *IEEE J. Ocean. Eng.*, vol. 47, no. 3, pp. 780–798, Jul. 2022.

[2] A. L. Diaz, A. E. Ortega, H. Tingle, A. Pulido, O. Cordero, M. Nelson, N. E. Cocoves, J. Shin, R. R. Carthy, B. E. Wilkinson, and P. G. Ifju, "The bathy-drone: An autonomous uncrewed drone-tethered sonar system," *Drones*, vol. 6, no. 10, p. 294, Oct. 2022.

[3] F. Thompson and D. Guihen, "Review of mission planning for autonomous marine vehicle fleets," *J. Field Robot.*, vol. 36, no. 2, pp. 333–354, Mar. 2019.

[4] M. J. Bays, T. A. Wettergren, J. Shin, S. Chang, and S. Ferrari, "Persistent schedule evaluation and adaptive re-planning for maritime search tasks," *J. Intell. Robot. Syst.*, vol. 110, no. 2, p. 65, Apr. 2024.

[5] P. Melchior, B. Orsoni, O. Lavialle, A. Poty, and A. Oustaloup, "Consideration of obstacle danger level in path planning using and fast-marching optimisation: Comparative study," *Signal Process.*, vol. 83, no. 11, pp. 2387–2396, Nov. 2003.

[6] J. Lin, A. S. Morse, and B. D. O. Anderson, "The multi-agent rendezvous problem," in *Proc. 42nd IEEE Int. Conf. Decis. Control*, vol. 2, Dec. 2003, pp. 1508–1513.

[7] J. Fang, A. S. Morse, and M. Cao, "Multi-agent rendezvousing with a finite set of candidate rendezvous points," in *Proc. Amer. Control Conf.*, Jun. 2008, pp. 765–770.

[8] J. Lin, A. Morse, and B. Anderson, "The multi-agent rendezvous problem. An extended summary," in *Cooperative Control*. Berlin, Germany: Springer, 2005, pp. 257–289.

[9] D. Ozsoyeller, Ö. Özkasap, and M. Aloqaily, "M-RENDEZVOUS: Multi-agent asynchronous rendezvous search technique," *Future Gener. Comput. Syst.*, vol. 126, pp. 185–195, Jan. 2022.

[10] E. J. Anderson and R. R. Weber, "The rendezvous problem on discrete locations," *J. Appl. Probab.*, vol. 27, no. 4, pp. 839–851, Dec. 1990.

[11] A. Pelc, "Deterministic rendezvous algorithms," in *Distributed Computing By Mobile Entities: Current Research in Moving and Computing*. Cham, Switzerland: Springer, 2019, pp. 423–454.

[12] A. Ta-Shma and U. Zwick, "Deterministic rendezvous, treasure hunts, and strongly universal exploration sequences," *ACM Trans. Algorithms*, vol. 10, no. 3, pp. 1–15, Jun. 2014.

[13] D. Dereniowski, R. Klasing, A. Kosowski, and Ł. Kuszner, "Rendezvous of heterogeneous mobile agents in edge-weighted networks," *Theor. Comput. Sci.*, vol. 608, pp. 219–230, Dec. 2015.

[14] J. Yu and S. M. LaValle, "Structure and intractability of optimal multi-robot path planning on graphs," in *Proc. AAAI Conf. Artif. Intell.*, vol. 27, 2013, pp. 1443–1449.

[15] Q. Hu, W. Chen, and L. Guo, "Fixed-time maneuver control of spacecraft autonomous rendezvous with a free-tumbling target," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 2, pp. 562–577, Apr. 2019.

[16] S. R. Sahoo, R. N. Banavar, and A. Sinha, "Rendezvous in space with minimal sensing and coarse actuation," *Automatica*, vol. 49, no. 2, pp. 519–525, Feb. 2013.

[17] V. Yordanova, H. Griffiths, and S. Hailes, "Rendezvous planning for multiple autonomous underwater vehicles using a Markov decision process," *IET Radar, Sonar Navigat.*, vol. 11, no. 12, pp. 1762–1769, Dec. 2017.

[18] R. Song, Y. Liu, and R. Bucknall, "A multi-layered fast marching method for unmanned surface vehicle path planning in a time-variant maritime environment," *Ocean Eng.*, vol. 129, pp. 301–317, Jan. 2017.

[19] Y. Liu, W. Liu, R. Song, and R. Bucknall, "Predictive navigation of unmanned surface vehicles in a dynamic maritime environment when using the fast marching method," *Int. J. Adapt. Control Signal Process.*, vol. 31, no. 4, pp. 464–488, Apr. 2017.

[20] X.-P. Yan, S.-W. Wang, F. Ma, Y.-C. Liu, and J. Wang, "A novel path planning approach for smart cargo ships based on anisotropic fast marching," *Expert Syst. Appl.*, vol. 159, Nov. 2020, Art. no. 113558.

[21] Y. Zhang, P. Chen, L. Chen, and J. Mou, "A path planning method for the autonomous ship in restricted bridge area based on anisotropic fast marching algorithm," *Ocean Eng.*, vol. 269, Feb. 2023, Art. no. 113546.

[22] P. Chen, Y. Huang, E. Papadimitriou, J. Mou, and P. van Gelder, "Global path planning for autonomous ship: A hybrid approach of fast marching square and velocity obstacles methods," *Ocean Eng.*, vol. 214, Oct. 2020, Art. no. 107793.

[23] H.-J. Park, H.-D. Seo, and P.-S. Lee, "Direct imposition of the wall boundary condition for simulating free surface flows in SPH," *Struct. Eng. Mech.*, vol. 78, pp. 497–518, Jan. 2021.

[24] H.-J. Park and H.-D. Seo, "A new SPH-FEM coupling method for fluid–structure interaction using segment-based interface treatment," *Eng. Comput.*, vol. 40, no. 2, pp. 1127–1143, Apr. 2024.

[25] A. M. Jokisaari, P. W. Voorhees, J. E. Guyer, J. Warren, and O. G. Heinonen, "Benchmark problems for numerical implementations of phase field models," *Comput. Mater. Sci.*, vol. 126, pp. 139–151, Jan. 2017.

[26] C. S. Peskin, "Flow patterns around heart valves: A numerical method," *J. Comput. Phys.*, vol. 10, no. 2, pp. 252–271, Oct. 1972.

[27] J. Kim, M. Jacobs, S. Osher, and N. C. Admal, "A crystal symmetry-invariant Kobayashi–Warren–Carter grain boundary model and its implementation using a thresholding algorithm," *Comput. Mater. Sci.*, vol. 199, Nov. 2021, Art. no. 110575.

[28] A. Telea, "An image inpainting technique based on the fast marching method," *J. Graph. Tools*, vol. 9, no. 1, pp. 23–34, Jan. 2004.

[29] R. Jiang, F. Dai, Y. Liu, and A. Li, "Fast marching method for microseismic source location in cavern-containing rockmass: Performance analysis and engineering application," *Engineering*, vol. 7, no. 7, pp. 1023–1034, Jul. 2021.

[30] N. Forcadel, C. L. Guyader, and C. Gout, "Generalized fast marching method: Applications to image segmentation," *Numer. Algorithms*, vol. 48, pp. 189–211, Feb. 2008.

[31] A. Valero-Gomez, J. V. Gomez, S. Garrido, and L. Moreno, "The path to efficiency: Fast marching method for safer, more efficient mobile robot trajectories," *IEEE Robot. Autom. Mag.*, vol. 20, no. 4, pp. 111–120, Dec. 2013.

[32] S. Garrido, L. Moreno, and D. Blanco, "Voronoi diagram and fast marching applied to path planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2006, pp. 3049–3054.

[33] S. Garrido, M. Malfaz, and D. Blanco, "Application of the fast marching method for outdoor motion planning in robotics," *Robot. Auto. Syst.*, vol. 61, no. 2, pp. 106–114, Feb. 2013.

[34] D. Álvarez, J. V. Gómez, S. Garrido, and L. Moreno, "3D robot formations path planning with fast marching square," *J. Intell. Robot. Syst.*, vol. 80, nos. 3–4, pp. 507–523, Dec. 2015.

[35] J. V. Gómez, N. Mavridis, and S. Garrido, "Fast marching solution for the social path planning problem," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 1871–1876.

[36] J. N. Tsitsiklis, "Efficient algorithms for globally optimal trajectories," *IEEE Trans. Autom. Control*, vol. 40, no. 9, pp. 1528–1538, Sep. 1995.

[37] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proc. Nat. Acad. Sci. USA*, vol. 93, no. 4, pp. 1591–1595, Feb. 1996.

[38] R. Sedgewick and K. Wayne, *Algorithms*. Reading, MA, USA: Addison-Wesley, 2008.

[39] Y. Liu and R. Bucknall, "The angle guidance path planning algorithms for unmanned surface vehicle formations by using the fast marching method," *Appl. Ocean Res.*, vol. 59, pp. 327–344, Sep. 2016.

[40] G. Tan, J. Zou, J. Zhuang, L. Wan, H. Sun, and Z. Sun, "Fast marching square method based intelligent navigation of the unmanned surface vehicle swarm in restricted waters," *Appl. Ocean Res.*, vol. 95, Feb. 2020, Art. no. 102018.

[41] G. Tan, J. Zhuang, J. Zou, and L. Wan, "Adaptive adjustable fast marching square method based path planning for the swarm of heterogeneous unmanned surface vehicles (USVs)," *Ocean Eng.*, vol. 268, Jan. 2023, Art. no. 113432.

[42] S. Garrido, L. Moreno, M. Abderrahim, and D. Blanco, "FM^2: A real-time sensor-based feedback controller for mobile robots," *Int. J. Robot. Autom.*, vol. 24, no. 1, pp. 3169–3192, 2009.

[43] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[44] J. V. Gomez. (2017). *N-Dimensional Fast Methods Library V0.7*. [Online]. Available: https://github.com/jvgomez/fast_methods

[45] J. Maddock and S. Clearly, "C++ type traits," *Dr. Dobb's J.*, vol. 25, no. 10, p. 38, 2000.

[46] Z. Zeng, A. Lammas, K. Sammut, F. He, Y. Tang, and Q. Ji, "Path planning for rendezvous of multiple AUVs operating in a variable ocean," in *Proc. 4th Annu. IEEE Int. Conf. Cyber Technol. Autom., Control Intell.*, Jun. 2014, pp. 451–456.

[47] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, Jun. 2011.

[48] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robot. Autom. Mag.*, vol. 19, no. 4, pp. 72–82, Dec. 2012. [Online]. Available: https://ompl.kavrakilab.org

[49] J. L. Leitschuh and T. J. Clark, "Grip: Graphically represented image processing engine," Dept. Comput. Sci., Worcester Polytech. Inst., Worcester, MA, USA, Tech. Rep. E-project-042816-161651, Apr. 2016.

JAEKWANG KIM received the B.S. degree in naval architecture and ocean engineering from Seoul National University, in 2017, and the M.S. and Ph.D. degrees in theoretical and applied mechanics from the University of Illinois at Urbana–Champaign, in 2023. He is currently an Assistant Professor with the Department of Mechanical and Design Engineering, Hongik University.

HYUNG-JUN PARK received the B.S. degree in naval architecture and ocean engineering from Inha University, in 2015, and the M.S. and Ph.D. degrees in mechanical engineering from Korea Advanced Institute of Science and Technology in 2021. He is currently an Assistant Professor with the School of Mechanical and Aerospace Engineering, Sunchon National University.

ADITYA PENUMARTI received the B.S. degree in mechanical engineering from North Carolina State University, in 2021, and the M.S. degree in mechanical engineering from the University of Florida, in 2023, where he is currently pursuing the Ph.D. degree with the Active Perception and Robot Intelligence Laboratory (APRILab), under the supervision of Dr. Jane Shin.

JAEJEONG (JANE) SHIN (Member, IEEE) received the B.S. degree in naval architecture and ocean engineering from Seoul National University, in 2017, and the combined M.S. and Ph.D. degrees in mechanical engineering from Cornell University, in 2021. Since 2021, she has been an Assistant Professor with the Department of Mechanical and Aerospace Engineering, University of Florida.