2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

# Flying Together: Human-Guided Immersive Shared Control for Aerial Robot Teams in Unknown Environments

Lou De Bel-Air[1,2], Luca Morando[1,3], Ruitao Chen[1], Keru Wang[1], Benjamin Jarvis[2], Charbel Toumieh[2], Yang Zhou[1], Ken Perlin[1], Dario Floreano[2], and Giuseppe Loianno[3]

***Abstract*— While autonomous multi-robots can achieve safe and coordinated navigation, they often struggle to adapt to unforeseen conditions and to capture operator-driven objectives in unstructured environments. We present a Virtual Reality (VR)-based shared control framework for teams of drones operating in constrained and unknown environments, enabling real-time, user-guided exploration. At the core of our approach is a novel, user-guided motion-primitive-based planner that computes continuous, collision-free trajectories while continuously integrating operator input. This planner is coupled with an admittance controller, allowing the operator to flexibly influence team behavior and guide drones toward regions of interest that autonomous planners may overlook. The system supports mixed-reality operations with both physical and simulated drones, and implements a bilateral VR-based interface, allowing the operator to guide the robot team via migration points while receiving immediate visual feedback of the team state. Experimental results show that shared control improves obstacle avoidance, maintains inter-agent spacing, and reduces operator effort, demonstrating the feasibility and advantages of immersive, human-in-the-loop multi-robot navigation.**

## I. Introduction

Unmanned Aerial Vehicles (UAVs) are increasingly deployed for tasks such as search and rescue [1], environmental monitoring [2], and infrastructure inspection [3]. In many of these applications, autonomous navigation offers scalability and can maximize coverage while reducing operator workload, yet full autonomy often limits user interaction and adaptability in complex, dynamic environments. Consider a disaster response scenario in which a swarm of drones explores a partially collapsed building. While autonomy can manage large-scale coverage, a human operator may still need to guide drones toward specific areas of interest, adjust their trajectories in real time, or prioritize regions based on visual cues unavailable to the planner. This tension between autonomy and user input motivates the use of shared control, a paradigm well established for manipulators but still limited for aerial robots given their virtually unbounded 3D workspace. In addition, the field still lacks an approach that scales to multi-robot coordination and incorporates real-time human guidance through immersive, virtualized interaction, which functions as an integral modality for operator input within the shared control loop.

The key problem we address in this work is how to leverage aerial robot teams for user-guided navigation and exploration in a way that is both immersive and goal-oriented, while still allowing the operator to influence trajectories for exploration purposes. Recent advances have shown that aerial robot teams can outperform single-drone systems in exploration and coverage tasks due to their larger parallelized sensing and robustness to failures [4]. At the same time, drone teleoperation research has demonstrated that 3D interfaces, such as Virtual Reality (VR), provide more intuitive control compared to traditional 2D interfaces, and even outperform First-Person-View (FPV) control in terms of task completion time, path optimality, and operator workload [5]. Our prior work on user-in-the-loop drone navigation [6] introduced an immersive framework for guided drone control. A subsequent user study [7] demonstrated that operators placed high trust in mixed-reality (MR) representations and preferred guided trajectories over approaches relying on excessive autonomy. However, several limitations remained: (i) trajectories were only recomputed at intermediate goals, preventing escape from confined regions; (ii) the generated trajectories were not always dynamically feasible, limiting their applicability in multi-robot settings; and (iii) user control was minimal, with drones forced to strictly follow planner outputs without allowing operator adjustments.

To address these challenges, we propose a 3D VR-based shared control framework for teams of aerial robots operating in unknown environments. At its core, we design a novel real-time, collision-free, minimum-time, operator-aware sampling-based motion planner. This planner, which balances autonomy with user-guided navigation, is integrated with an admittance control mechanism to ensure safe and dynamically feasible motion while maintaining flexibility for real-time operator inputs. Motion primitives are used to guarantee feasibility at the team level, enabling consistent group behavior and collision-free coordination. Individual agent motion is coordinated by decomposing the overall group behavior and distributing it across the team, while accounting for cohesion, trajectory tracking, velocity consensus, and obstacle avoidance in a distributed fashion. Finally, we present a web-based immersive interface, an integral component of the shared control loop, which enables intuitive, real-time operator input and full control of aerial robot teams. While the web-based design introduces additional

[1]The authors are with New York University, NY 10012, USA. {ld3280, luca.morando, rc4000, kw2727, yangzhou, kp1}@nyu.edu.

[2]The authors are with the Ecole Polytechnique Federale de Lausanne, Station 9 CH-1015 Lausanne, Switzerland. email: {benjamin.jarvis, charbel.toumieh,dario.floreano}@epfl.ch.

[3]The author is with the University of California Berkeley, Department of Electrical Engineering and Computer Sciences, Berkeley, CA 94720, USA. email: loiannog@eecs.berkeley.edu.

This work was supported by the NSF CPS Grant CNS-2603416, the NSF CAREER Award 2546659, and the DARPA YFA Grant D22AP00156-00.

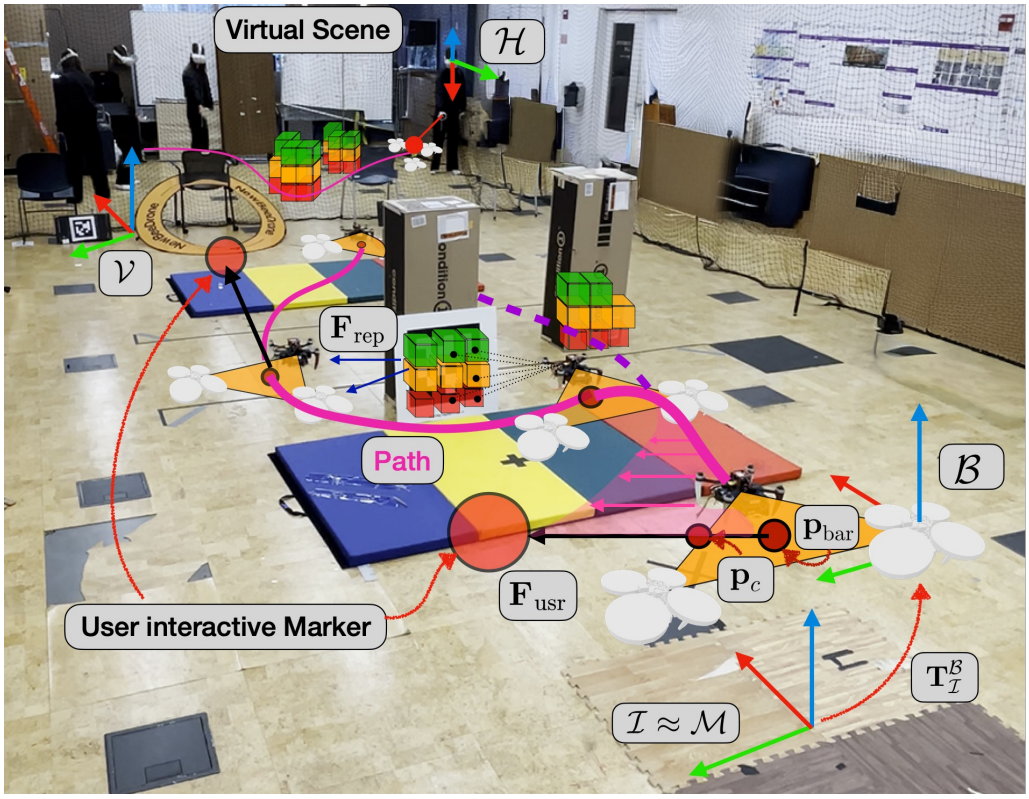


Fig. 1: Mixed-reality experiment with one real and two simulated drones (white). The operator in the background guides the drones in the VR environment using a red marker ($\mathbf{F}_{\text{usr}}$), and the planner reacts to this input in real-time (pink). $\mathbf{p}_{\text{bar}}$ and $\mathbf{p}_c$ are robots barycenter and migration point; $\mathbf{F}_{\text{rep}}$ is the obstacle repulsion. Frames $\mathcal{I}, \mathcal{M}, \mathcal{B}, \mathcal{V}, \mathcal{H}$ are defined in Section III-A.

development complexity, it offers significant advantages: it is open-source, cross-platform, multi-user, and widely accessible, eliminating reliance on proprietary software and democratizing immersive human–robot interaction. This approach provides a transparent, reproducible, and widely deployable solution for multi-robot shared control.

An overview of the proposed system with one real and two virtual aerial robots, highlighting the immersive interface and planner adaptation, is shown in Fig. 1.

## II. Related Works

Human–robot collaboration has progressed beyond industrial manipulators [8] to include small-scale aerial robots, which are increasingly vital for search and rescue [1], infrastructure inspection, and patrolling [9]. Interaction with aerial vehicles remains limited, often depending on visual feedback [10] or joystick control. Furthermore, current methods scale poorly when managing teams of aerial robots, as operators are typically restricted to assignment of high-level waypoints [11].

**Shared Control**. Shared control has often been studied in simpler systems such as manipulators [12]. Extending these solutions to aerial robots is challenging due to their higher motion complexity, virtually unbounded workspace compared to traditional manipulators and the impossibility to have tactical feedback, which often requires a virtualization of the interaction. An approach for single agent is proposed in [13], where operator commands are combined with autonomous behaviors to improve safety, precision, and task efficiency. Extending these approaches to team of aerial robots introduces unique challenges: multiple agents must coordinate in three-dimensional, unconstrained environments while simultaneously following human intent, maintaining inter-agent cohesion, and avoiding collisions.

**Teleoperation**. Teleoperation remains one of the most direct approaches to robot control, yet turning robots into effective human partners rather than simple executors is still a challenge [14]. Current autopilots can generate agile trajectories [15] or optimize exploration paths in indoor environments [16], but offer limited means for intuitive human-robot communication [17]. A promising step forward is made in [6], where an admittance controller based on interaction forces allowed the user to perceive feedback from both the robot and its environment. However, the planning support in that work was minimal and not scalable to multi-robot scenarios. Similarly, Jump Point Search has been proposed as a middle layer between user motion inputs and robot execution [5], [18], but these methods still lack the ability to generate real-time, dynamically feasible trajectories [19]. Bridging this gap with planners that can seamlessly integrate user preferences and robot dynamics in cluttered environments remains an open research direction.

**Multi-Robot Exploration**. Several methods have been proposed for cooperative exploration using aerial and ground robots [20]. Although approaches rooted in traditional control theory can offer mathematical guarantees, they remain

limited when handling complex indoor environments [21], especially when a human operator is part of the control loop. This challenge is increasingly relevant as demand for aerial swarm systems grows, making Human-Swarm Interaction (HSI) a critical research area.

In most current multi-robot applications, operators are forced to divide their attention among individual robots increasing workload and reducing situational awareness issues that can have severe consequences [22], [23]. Some promising solutions have emerged, including a swarm interaction framework grounded in cognitive models of human decision-making, designed to reduce operator load in complex tasks [24], and a multi-robot task allocation mechanism that enables effective human oversight, as demonstrated in the DARPA Subterranean Challenge [25].

Despite these advances, a general and scalable framework for centralized, human-in-the-loop swarm control in constrained environments remains absent.

**Immersive Interfaces for Multi-Robot Control**. The advent of extended reality interfaces has significantly advanced intuitive and effective interaction in robotics, improving both information exchange and visualization [26]. These enable more natural communication of user intentions to robotic teammates [27], [28]. In [29], users interact with a virtual exocentric view of a drone and specify final poses via intuitive pick-and-place gestures, while in [30], operators assign sequences of waypoints. Similarly, [17] proposes a Mixed Reality (MR) framework that replaces the ground station with a holographic satellite map displaying live robot status updates, thereby improving situational awareness during human-robot collaboration. More advanced approaches, such as [5], [6], extend this paradigm by introducing perception-based interaction, where robot information is directly translated into graphical cues facilitating cooperation. Meanwhile, recent progress in graphics, connectivity, and rendering quality has enabled new virtual environments like WebXR [31]. These are customizable, web-based, cross-platform, and support multi-user interaction, showing promising results in robotics research and human-robot interaction tasks [32], [33].

Building on these advances, our approach integrates custom VR graphics with a low-latency communication layer linking the VR interface and the robots, combined with an operator-assisted multi-robot motion planner.

## III. Methodology

In this section, we present the key components of the proposed immersive shared control framework, which allows operators and autonomous planners to collaboratively guide multi-robot teams. A motion-primitive-based planner (Section III-B) generates dynamically feasible paths, which are integrated with user input through an admittance controller (Section III-C). At the lower level, inter-agent force-based interactions provide robust multi-robot coordination (Section III-D). Finally, Section III-E presents the virtual reality interface and low-latency middleware connecting the user to the robots. An overview of the complete framework is shown in Fig. 2.

### A. Frames, Notation, Localization, and Mapping

As shown in Fig. 1, the inertial frame $\mathcal{I}$ is shared between the user and the robots, while the map frame $\mathcal{M}$ is aligned with $\mathcal{I}$ at the take-off location. Each robot $i$ has its own body frame $\mathcal{B}_i$. The user's head-mounted display is represented by frame $\mathcal{H}$, and the interactive marker is attached to frame $\mathcal{K}$. Finally, the virtual world is represented by frame $\mathcal{V}$, which aligns with the inertial frame. The main frames are illustrated in Fig. 1. We denote by $\mathbf{T}_i^j$ the homogeneous transformation matrix from frame $i$ to frame $j$. Each robot $A_i^{\mathcal{I}}$, with $i \in \{0, \ldots, N\}$, is localized in the inertial frame through the transformation $\mathbf{T}_{\mathcal{I}}^{\mathcal{B}_i}$, obtained from on-board state estimation and localization algorithms. The same alignment holds for the user, where $U^{\mathcal{V}}$ is expressed in the virtual frame $\mathcal{V}$, which is aligned with $\mathcal{I}$.

On the perception side, the agents can generate a point cloud $P_{\mathcal{B}}^{\mathcal{I}}$, initially expressed in its body frame $\mathcal{B}_i$ and then transformed into the map frame $\mathcal{M}$ using $\mathbf{T}_{\mathcal{I}}^{\mathcal{B}_i}$. This point cloud is converted into a real-time voxel map representation $V_{\mathcal{B}_i}^{\mathcal{I}}$, providing a geometric model of the environment.

### B. User-in-the-Loop Motion Primitive Planner

To assist the user-guided navigation, we employ a motion-primitive-based planner, which generates continuous, smooth trajectories by concatenating short-duration primitives derived from optimal control solutions [19]. Unlike previous approaches for assisted drone teleoperation [6], which rely on geometrically disconnected RRT* segments, our framework ensures path continuity, minimizing abrupt maneuvers that could destabilize swarm formations. For real-time navigation in partially known environments, planning is performed within a 3D voxel map $V_{\mathcal{B}}^{\mathcal{I}}$ collected by one of the agents. This map is limited to the robot's horizon $h$ and planning is performed toward a goal $G$ projected within this horizon. Precomputed motion primitives form a connected graph, where edges correspond to feasible state transitions and sequences are selected using graph search (A*) to minimize a cost function that balances smoothness, safety, and task efficiency

$$J^* = \min_{D,T} \; J_q(D) + \rho T + \rho_c J_c(D) + \rho_{\text{usr}} J_{\text{usr}}(D), \quad (1)$$

where $D$ is a candidate trajectory, $T$ its duration, $J_q$ penalizes control effort, $J_c$ penalizes obstacle proximity, and $J_{\text{usr}}$ incorporates live user input. The weights $\rho$, $\rho_c$ and $\rho_{\text{usr}}$ adjust the relative influence of each term.

The key novelty of our approach is the user alignment term $J_{\text{usr}}$, which allows the input of the live operator to influence the planned trajectory in real time

$$J_{\text{usr}}(D) = \int_0^T \|\mathbf{F}_{\text{usr}}\| \big(1 - \hat{\mathbf{F}}_{\text{usr}} \cdot \hat{\mathbf{v}}_D(t)\big) e^{-t/\tau} dt. \quad (2)$$

Here, the user force $\mathbf{F}_{\text{usr}}$ encodes the user's intent and is computed as $\mathbf{F}_{\text{usr}} = \mathbf{K}_p(\mathbf{p}_{\text{bar}} - \mathbf{p}_u) - \mathbf{K}_d(\dot{\mathbf{p}}_{\text{bar}} - \dot{\mathbf{p}}_u)$, with $\mathbf{K}_p$ and $\mathbf{K}_d$ the proportional and derivative gains, $\mathbf{p}_{\text{bar}}$ the barycenter (mean position of agents), and $\mathbf{p}_u$ the position where the user intends to drag the robots. In $J_{\text{usr}}$, the magnitude $\|\mathbf{F}_{\text{usr}}\|$ ensures that stronger inputs exert proportionally

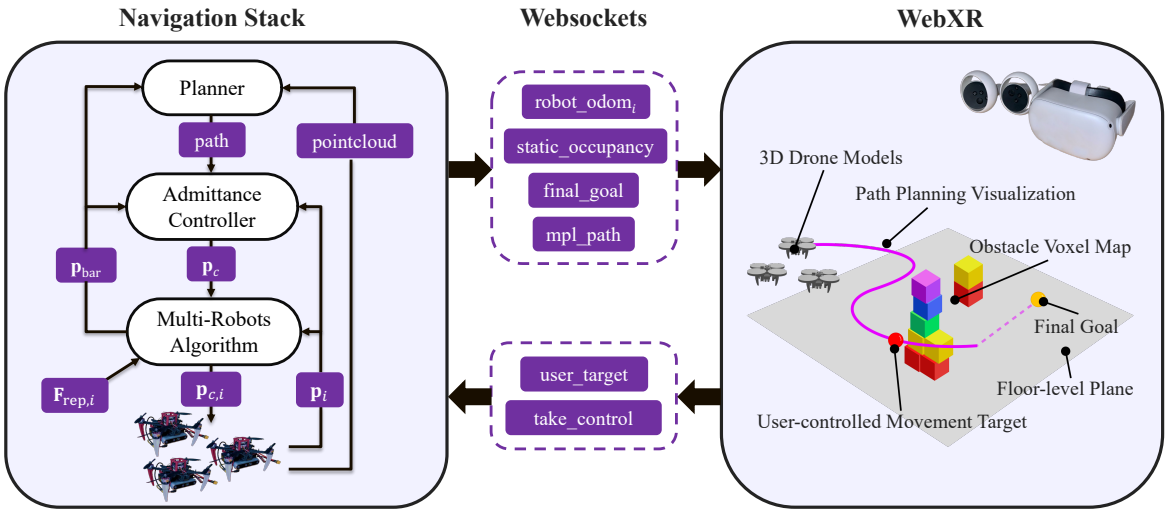


Fig. 2: Overview of the system: navigation stack (left; topics include barycenter of the robots $\mathbf{p}_{\text{bar}}$, obstacle repulsive forces $\mathbf{F}_{\text{rep},i}$, agent positions $\mathbf{p}_i$, migration point $\mathbf{p}_c$, and commanded positions $\mathbf{p}_{c,i}$), bidirectional WebSocket communication layer (center; topics detailed in Table I), and VR visualization of drones, local planner path, and environment (right).

greater influence on the trajectory. The unit vector $\hat{\mathbf{F}}_{\text{usr}}$ and $\hat{\mathbf{v}}_D(t)$, which is the unit velocity vector along the primitives, favor trajectories aligned with the user's intention while penalizing misaligned ones. The exponential decay $e^{-t/\tau}$ restricts user influence to the early portion of the trajectory, allowing later primitives to prioritize efficiency toward the goal. The effect is shown in Fig. 1, where the optimal trajectory is smoothly perturbed to reflect user intention.

During execution, the planner incrementally sequences motion primitives into a locally feasible trajectory. This process is re-evaluated every two seconds, enabling adaptation to dynamic obstacles while maintaining continuous user involvement.

### C. Variable Admittance Controller

Once a path has been generated by the planner, the challenge is that the drones do not follow it autonomously and instead adapt to the user's commands. To address this, we define an extended Variable Admittance Controller (VAC) [6], which combines the motion-primitive path with the user's intent to generate a commanded position for the agents to follow, while maintaining proximity to the planned trajectory. The virtual force input driving the VAC is defined as $\mathbf{F}_v = \mathbf{F}_{\text{usr}} + \mathbf{F}_{\text{rep}}$, where $\mathbf{F}_{\text{usr}}$ is the user force introduced in Section III-B, and $\mathbf{F}_{\text{rep}}$ is an obstacle repulsion force that ensures the commanded position remains clear of obstacles. The repulsion force is computed as

$$\mathbf{F}_{\text{rep}} = \sum_{i=0}^{N_v} \mathbf{F}_{r_i} = \sum_{i=0}^{N_v} \frac{F_s}{k} e^{-\lambda d_i} \left(1 - e^{h-d_i}\right) \tag{3}$$

with $v_i \in V^W$ the voxels within the robot horizon $h$, $d_i$ is the distance to voxel $v_i$, $F_s$ is the maximum force, and $k = 1 - e^h$ normalizes the force $\|\mathbf{F}_{r_i}\| = F_s$ when $d_i = 0$.

Next, the barycenter of the agents, $\mathbf{p}_{\text{bar}}$, is projected onto the path generated by the planner to define a sequence of reference positions $\mathbf{p}_r$. The VAC then computes a commanded position $\mathbf{p}_c$, which serves as the migration point for the team and evolves according to mass–spring–damper dynamics

$$\mathbf{M}\left(\ddot{\mathbf{p}}_c - \ddot{\mathbf{p}}_r\right) + \mathbf{D}\left(\dot{\mathbf{p}}_c - \dot{\mathbf{p}}_r\right) + \mathbf{K}\left(\mathbf{p}_c - \mathbf{p}_r\right) = \mathbf{F}_v \tag{4}$$

where $\mathbf{M}$, $\mathbf{D}$, and $\mathbf{K}$ are diagonal matrices encoding mass, damping, and stiffness in the world frame.

### D. Multi-robots Coordination Execution

The commanded position $\mathbf{p}_c$ produced by the admittance controller must be distributed among the individual agents to enable safe and coordinated tracking. We adopt a force based algorithm inspired by Olfati–Saber [34]. For each agent $i$, the reference acceleration is computed as the combination of inter-agent cohesion, velocity consensus, centralized tracking, and obstacle repulsion

$$\begin{aligned}\mathbf{a}_i^{\text{ref}} = &\underbrace{\sum_{j\in\mathcal{N}_i} \phi(\mathbf{p}_{ji})}_{\text{cohesion}} + \beta \underbrace{\sum_{j\in\mathcal{N}_i} (\mathbf{v}_j - \mathbf{v}_i)}_{\text{consensus}} \\ &+ \underbrace{K_p(\mathbf{p}_c - \mathbf{p}_i) + K_v(\mathbf{v}_c - \mathbf{v}_i)}_{\text{tracking}} + \underbrace{\mathbf{F}_{\text{rep},i}}_{\text{obstacle}}\end{aligned} \tag{5}$$

where $\mathbf{p}_{ji} = \mathbf{p}_j - \mathbf{p}_i$, with $\mathbf{p}_i$ and $\mathbf{v}_i$ the position and velocity of agent $i$, and $\mathbf{F}_{\text{rep},i}$ the obstacle repulsion force described in Section III-C. This repulsion term must also be incorporated into the multi-robots algorithm itself, as the admittance controller ensures that the migration point avoids obstacles, but each agent still needs to individually avoid collisions. The inter-agent cohesion function $\boldsymbol{\phi}(\mathbf{p}_{ji})$ is defined as

$$\boldsymbol{\phi}(\mathbf{p}_{ji}) = \alpha\, \rho_h\left(\frac{\sigma(\|\mathbf{p}_{ji}\|)}{\sigma(R)}\right)\left(\sigma(\|\mathbf{p}_{ji}\|) - \sigma(d_{ij}^{\text{ref}})\right)\frac{\mathbf{p}_{ji}}{\|\mathbf{p}_{ji}\|} \tag{6}$$

where $\sigma(\cdot)$ denotes the $\sigma$-norm, $\rho_h(\cdot)$ is a smooth cutoff function, $R$ is the interaction radius, and $d_{ij}^{\text{ref}}$ is the desired inter-agent distance.

Acceleration is converted into a position command $\mathbf{p}_{c,i}$ for each $i^{\text{th}}$ agent through double integration. Both acceleration

and velocity are saturated to respect the dynamic limits of the drones and prevent unsafe maneuvers.

### E. Virtual Reality Platform and Communication Interface

By providing an immersive 3D interface of both the environment and the robot team, VR naturally functions as an integral modality for operator input within the shared control loop, enhancing operator situational awareness, enabling intuitive manipulation of migration points, and supporting real-time guidance that complements the motion-primitive planner and admittance controller.

To create an accessible and versatile immersive interface, we develop a virtual reality framework based on WebXR[1]. WebXR is a web-based framework that enables immersive Virtual and Augmented Reality experiences directly within web browsers, supporting both desktop and mobile head-mounted displays. It provides a flexible environment for rendering 3D content, capturing user interactions, and displaying spatial information in real time. WebXR also allows multiple users to share a virtual space and interact with dynamic objects, making it a natural platform for human-robot interaction and control.

In the context of aerial multi-robots teleoperation, WebXR provides the operator with a real-time visualization of the robots, trajectories and environmental context (Fig. 2). To seamlessly synchronize this virtual representation with the physical robots, a dedicated UDP WebSocket-based middleware is implemented. It bridges the control and perception stack running on the robots with the WebXR server running on the headset, enabling real-time communication of key visual cues, such as robot odometry, planned trajectories, and 3D map. Through this interface, the operator can monitor and influence the team in an intuitive, low-latency manner, while maintaining a consistent and interactive shared representation between the virtual and physical worlds. The implemented topics and related messages in the custom WebSocket based communication middleware are both visualized in Fig. 2 and summarized in Table I.

TABLE I: WebSocket middleware topics

| Topic Name | Message Type | Data Transmitted |
|---|---|---|
| robot_odom$_i$ | webxr_robot_pos | Robots' odometry |
| static_occupancy | webxr_static_occupancy | Point cloud representing the 3D occupancy map |
| final_goal | webxr_final_goal | High level goal in the map |
| mpl_path | webxr_mpl_path | Path computed by the motion primitive local planner |
| take_control | webxr_take_control | Boolean flag to switch control of the robots |
| user_target | webxr_des_pos | The user's desired position and orientation |

[1]https://www.w3.org/TR/webxr/

## IV. Results

We validate the proposed framework using a team of quadrotors (one physical quadrotor and two simulated agents). The experiments evaluate the impact of the user alignment term in the motion primitive planner and its role in supporting immersive, user-in-the-loop multi-robot navigation. Two scenarios are compared: (i) a baseline case where the user alignment term $J_{\text{usr}}$ is disabled, enforcing shortest-path planning; and (ii) a shared-control case where $J_{\text{usr}}$ is active, allowing the operator to influence the trajectory. In both conditions, the user interacts with the drones through the VR interface by moving continuously a migration point marker, while the team maintains cohesion using the coordination strategy described in Section III-D.

### A. Robotic Platform and VR Interface

The real drone used in the experiments is a quadrotor with a total weight of $1.508$ kg and thrust-to-weight ratio of $4:1$. The platform is equipped with a PixRacer Pro flight controller and an NVIDIA Jetson Orin as the central processing unit, running Ubuntu 22.04 and ROS 2[2]. For the perception side, the drone leverages an Intel Realsense D455 stereo camera for localization using a customized version of OpenVINS [35], which estimates the robot state at $100$ Hz. A neural depth package [36] is adopted for mapping purposes, providing a clear depth image as input to the NvBlox-based mapping package [37] from which a point cloud $P_{\mathcal{B}}^{\mathcal{I}}$ is generated. The line of sight of the robot perception is set at $3$ m with a voxel resolution of $20$ cm. The team is composed of two additional simulated robots, with their dynamics computed onboard the real quadrotor. For the VR interface, we employ the Meta Quest 2 headset running Meta Horizon OS, a public-level headset suitable for research prototyping.

The VR environment is rendered with layered feedback: drones as 3D models, ground plane, planned trajectories as 3D strokes, and obstacles as a voxel map reconstructed online from occupied-space samples. These samples are then rendered in three.js[3] using InstancedMesh, a technique that draws thousands of identical objects (here cubes) in a single GPU call, enabling smooth real-time updates. Voxel colors encode height with a red–green–purple gradient from low to high, helping users to quickly interpret the environment.

WebXR is a server-based, cross-user, and cross-platform VR framework that runs in the browser, transmitting only the visual cues required for headset rendering. To minimize the latency with the server, we adopt a Netgear NightHawk V2 router, providing a minimal latency between the robot, the VR server and the headset, with a measured ping of only $5-8$ ms. Communication relies on both ROS 2 topics using FastRTPS middleware and custom WebSocket messages (Section III-E).

The flying experiments are conducted in an indoor flying arena of approximately $60$ m$^2$. The operator performs VR

[2]www.ros.org

[3]https://threejs.org

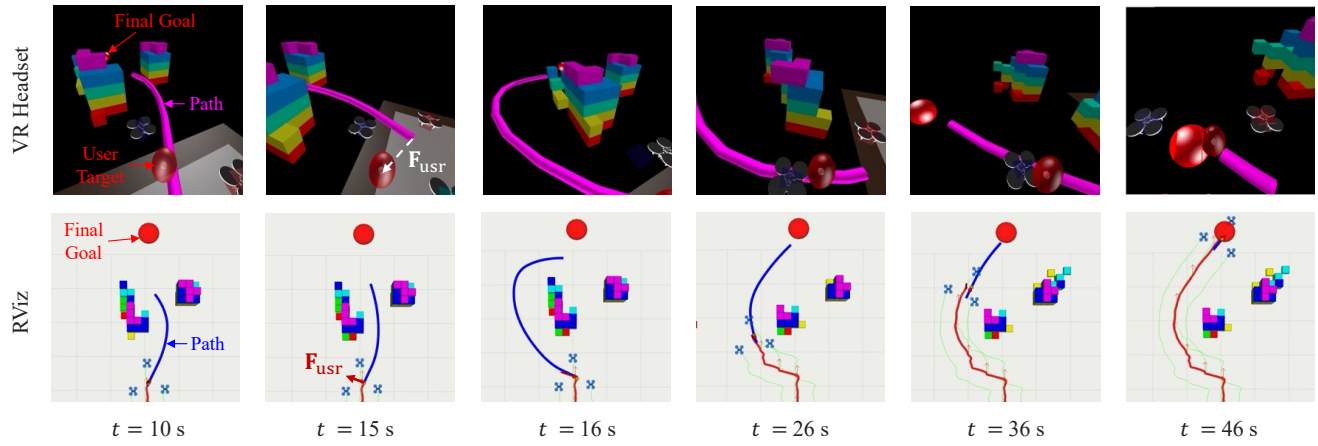


Fig. 3: Key frames from the shared-control experiment, showing the first-person VR view and the corresponding RViz visualization of the motion primitive planner's evolution as the user interacts with the VR target to guide the aerial team.

control in a separate sub-area of the same space, ensuring real-time interaction with the robot team while maintaining a safe separation from the real quadrotor (see Fig. 1).

### B. Experimental Parameters

*Motion Primitive Planner.* The planning horizon is set at the line of sight of the robot perception to $h = 3$ m with a motion primitive duration of $\Delta t = 1.3$ s. The cost function weights are chosen to balance trajectory duration ($\rho = 1.5$), obstacle force field ($\rho_c = 0.05$), and user influence ($\rho_{\text{usr}} = 35$) with an exponential decay constant $\tau = 0.8$. The goal is defined at $G^{\mathcal{I}} = [5.5, -0.1]$ m, with a tolerance of 0.5 m. To address the continuous exploration and the missing knowledge of the map a priori, a virtual goal $G_v^I$ is projected on the robot horizon $h$, resulting as local destination of the planner motion primitives.

*Admittance Controller.* We follow the approach from [6] to set the parameters, modifying only the obstacle force field $\mathbf{F}_{\text{rep}}$ to use an exponential decay ($\lambda = 0.55$) with a maximum magnitude of $F_s = 25.0$ N for smooth obstacle avoidance.

*Team Coordination:* The desired inter-agent distance is $d_{ij}^{\text{ref}} = 1.5$ m. To keep the formation while allowing flexibility near obstacles, cohesion, alignment, and tracking gains are set to $\alpha = 1.0$ s$^{-2}$, $\beta = 0.3$ s$^{-1}$, and $K_p = 2.0$ s$^{-2}$, with damping gain $K_v = 1.5$ s$^{-1}$. Agents interact within a radius $R = 5$ m, with $\sigma$-norm smoothing $\epsilon = 0.08$ and a cutoff transition $h = 0.1$ for smooth decay of interactions.

### C. Experimental Setup and Interaction

To validate our approach in realistic human–drone interactions within unknown constrained environments, we designed the following experiments. As described above, the team consists of one real robot $A_{0,r}$, and two simulated robots $A_{1,s}$ and $A_{2,s}$. While the simulated robots are visible to the user through the virtual environment, the real robot can perceive their presence through a continuous interaction based on cohesion forces and managed by the coordination control algorithm described in Section III. All robots, real or simulated, are subject to the user pulling force $\mathbf{F}_{\text{usr}}$ and the obstacle forces $\mathbf{F}_{\text{rep}}$ perceived in the real world through the agent $A_{0,r}$.

The user guides the drone team in VR using the handheld controller, while perceiving the environment correctly localized via the transformation $\mathbf{T}_{\mathcal{V}}^{\mathcal{I}}$ between the virtual ($\mathcal{V}$) and inertial ($\mathcal{I}$) frames (Fig. 1). After activating the WebSocket connection, the user can direct the team by dragging a virtual marker from the barycenter, indicating the target position for the team. This input goes through the admittance controller to update the team's migration point $\mathbf{p}_c$, allowing the drones to track it in real time.

For visualization, we present experiments with the obstacle configuration shown in Fig. 3. However, the setup is not restricted to this case, as the algorithm adapts to different scenarios, making the experiments repeatable under various environmental conditions.

### D. Trajectory Analysis and Team Performance

Fig. 3 shows representative frames of the shared-control experiment, including the operator's first-person headset view and the corresponding RViz visualization (in ROS). In VR, the path computed by the motion primitive planner is displayed in pink, while the operator interacts with the red target to provide the force $\mathbf{F}_{\text{usr}}$ to the admittance controller. In the first frame ($t = 10$ s) the user target is not moving, leading to $J_{\text{usr}} \approx 0$, and the displayed path corresponds to the shortest trajectory, passing through the two obstacles. At $t = 15$ s, the user moves the target away from the precomputed path to suggest a safer path that avoids both obstacles. During this interval, the robots remain close to the precomputed path, as the admittance controller prevents large deviations. One second later, once the path is recomputed, the planner incorporates $\mathbf{F}_{\text{usr}}$ into the cost function and generates a new path aligned with the user's intended direction, as shown in the third frame. From this point, the operator continuously moves the target along the recomputed path until the team reaches the goal. In the final stages, the user no longer deviates the target from the path, and the planner repeatedly outputs the shortest path to the goal. For both conditions, drone trajectories, distances to obstacles, and inter-agent distances are summarized in Fig. 4. In the baseline, $\mathbf{F}_{\text{usr}}$ is active to let the user drag the drones along

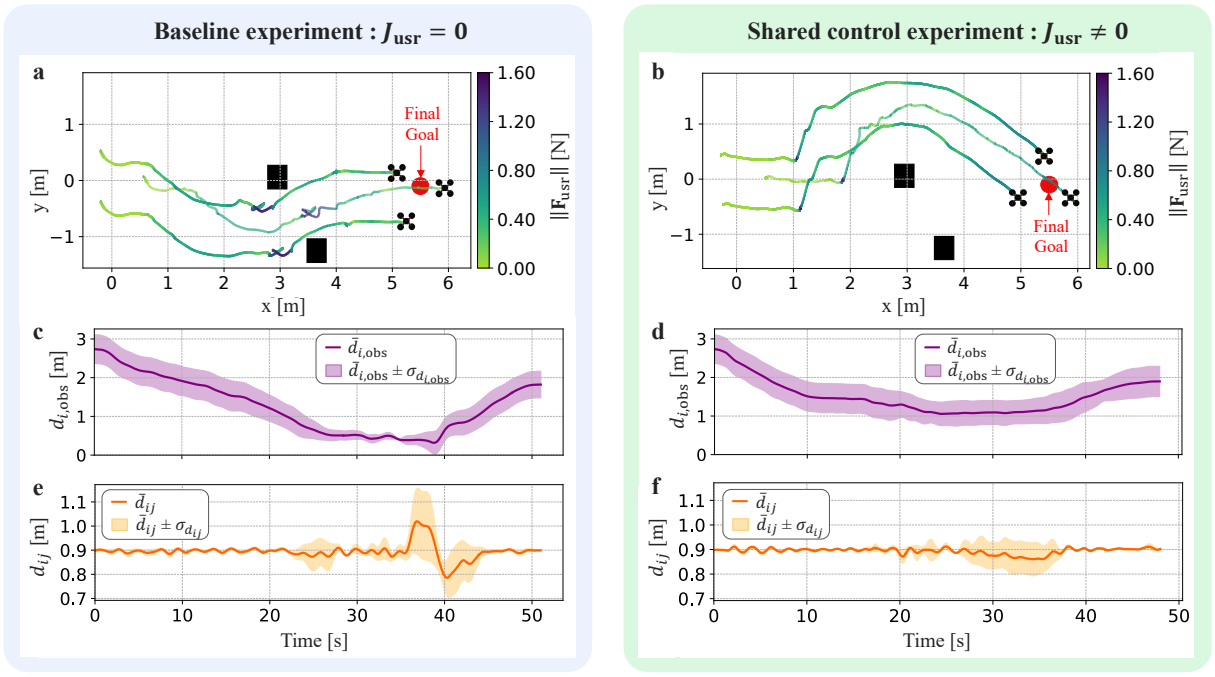


Fig. 4: Drone trajectories and multi-robots metrics for baseline and shared-control experiments. (a,b) Trajectories colored by user force $\mathbf{F}_{\text{usr}}$. (c,d) Distance to obstacles $d_{i,\text{obs}}$. (e,f) Inter-agent distances $d_{ij}$. Smoothed over 50 samples.

the computed path, but the cost $J_{\text{usr}}$ is disabled in the planning algorithm, preventing trajectory modification and forcing the robots to stay on the shortest path through the obstacles. This requires the operator to apply larger pulls on the migration point, inducing oscillations between the obstacles and increasing $\mathbf{F}_{\text{usr}}$ during 9 s. As the team passes through the gap, the distance to the closest obstacle $d_{i,\text{obs}}$ drops to 0.07 m, causing large fluctuations in inter-agent distances and higher collision risk. In the shared-control experiment, a short peak in $\mathbf{F}_{\text{usr}}$ occurs when the operator initially deviates the path, but the required force quickly stabilizes. The drones avoid both obstacles while maintaining nearly constant inter-agent distances, obstacle clearances, and user force. These results fully validate the operator-assisted planning approach, enabling trajectories that are impossible under standard planning and demonstrating a more stable, intuitive control process while maintaining assisted navigation at all times. These patterns are confirmed in the performance metrics reported in Table II. While the traveled distance remains nearly unchanged between conditions, the team reaches the goal about 6% faster under shared control. Most importantly, obstacle clearance improves substantially (0.67 m vs. 0.07 m in the baseline), and inter-agent spacing increases by 13%.

TABLE II: Multi-robots performance metrics for baseline and shared-control experiments

| Metric | Baseline | Shared-Control |
|---|---|---|
| Average distance traveled [m] | **6.8** | 6.9 |
| Time to reach the goal [s] | 51 | **48** |
| Mean velocity [m/s] | 0.133 | **0.144** |
| Min. distance to obstacles [m] | 0.07 | **0.67** |
| Min. distance inter-agent [m] | 0.68 | **0.77** |
| Average user force [N] | **0.38** | 0.39 |

### E. Discussion

The results demonstrate that incorporating the user alignment term $J_{\text{usr}}$ in the motion primitive planner effectively enhances mixed-reality multi-robot teleoperation. Shared control enables the operator to continuously influence the drones trajectory, resulting in safer and more coherent formations without increasing operator effort.

The experiments also validate the different modules of the proposed framework for VR-based multi-robot teleoperation, combining user-in-the-loop motion-primitive planner (Section III-B), admittance controller (Section III-C), Olfati–Saber coordination strategy (Section III-D), and bilateral interaction with the VR framework (Section III-E).

Our framework preserves operational assistance even when the operator deviates from precomputed paths, a feature absent in prior works such as [6], while simultaneously ensuring safety and flexibility.

## V. Conclusion

We presented a VR-based shared-control framework for teleoperating a team of drones in unknown environments. By combining a motion-primitive planner with an admittance controller, the system generates continuous, collision-free trajectories while allowing real-time operator-directed modifications. Mixed-reality experiments with both physical and simulated drones demonstrate that leveraging user intuition can concurrently improve obstacle clearance, maintain inter-agent distances, and reduce operator effort compared to fully autonomous planning.

This work provides a proof-of-concept for immersive human-in-the-loop multi-drone teleoperation, emphasizing

the benefits of operator flexibility, intuitive guidance, and bilateral interaction. Future works will investigate scaling to a larger number of robots, integrating more advanced swarm algorithms, and conducting user case studies to quantify operator performance and system usability.

## REFERENCES


[1] N. Michael, S. Shen, K. Mohta, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, K. Ohno, E. Takeuchi, and S. Tadokoro, "Collaborative mapping of an earthquake-damaged building via ground and aerial robots," *Journal of Field Robotics*, vol. 29, no. 5, pp. 832–841, 2012.

[2] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 46–56, Sept 2012.

[3] T. Ozaslan, G. Loianno, J. Keller, C. J. Taylor, V. Kumar, J. M. Wozencraft, and T. Hood, "Autonomous navigation and mapping for inspection of penstocks and tunnels with MAVs," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1740–1747, July 2017.

[4] Q. Quan, J. Xu, R. Liu, Y. Ding, J. Che, and K.-Y. Cai, "Self-organizing aerial swarm robotics for resilient load transportation : A table-mechanics-inspired approach," 2025.

[5] S. A. Salunkhe, P. Nedunghat, L. Morando, N. Bobbili, G. Li, and G. Loianno, "Intuitive human-drone collaborative navigation in unknown environments through mixed reality," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2025, pp. 862–868.

[6] L. Morando and G. Loianno, "Spatial assisted human-drone collaborative navigation and interaction through immersive mixed reality," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 8707–8713.

[7] L. Morando, X. Zhou, F. Atashzar, and G. Loianno, "Human-drone collaboration via mixed-reality for efficient navigation and interaction in constrained environments: a comprehensive user case study," *Auton Robot*, vol. 49, no. 40, 2025.

[8] H. Xing, A. Torabi, L. Ding, H. Gao, W. Li, V. K. Mushahwar, and M. Tavakoli, "Human-robot collaboration for heavy object manipulation: Kinesthetic teaching of the role of wheeled mobile manipulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 2962–2969.

[9] L. Morando, C. T. Recchiuto, and A. Sgorbissa, "Social drone sharing to increase the UAV patrolling autonomy in emergency scenarios," in *29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2020, pp. 539–546.

[10] W. A. Isop, C. Gebhardt, T. Nägeli, F. Fraundorfer, O. Hilliges, and D. Schmalstieg, "High-level teleoperation system for aerial exploration of indoor environments," *Frontiers in Robotics and AI*, vol. 6, 2019.

[11] J. Betancourt, B. Wojtkowski, P. Castillo, and I. Thouvenin, "Exocentric control scheme for robot applications: An immersive virtual reality approach," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2022.

[12] F. Stroppa, M. Selvaggio, N. Agharese, and et al., "Shared-control teleoperation paradigms on a soft-growing robot manipulator," *Journal of Intelligent & Robotic Systems*, vol. 109, no. 1, p. 30, 2023.

[13] R. Franceschini, M. Fumagalli, and J. C. Becerra, "Enhancing human-drone interaction with human-meaningful visual feedback and shared-control strategies," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2023, pp. 1162–1167.

[14] W. Bentz, L. Qian, and D. Panagou, "Expanding human visual field: online learning of assistive camera views by an aerial co-robot," *Autonomous Robots*, vol. 46, no. 8, pp. 949–970, 12 2022.

[15] A. Saviolo, G. Li, and G. Loianno, "Physics-inspired temporal learning of quadrotor dynamics for accurate model predictive trajectory tracking," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 256–10 263, 2022.

[16] J. Mao, R. C. Srinivas, S. Nogar, and G. Loianno, "Time-optimized safe navigation in unstructured environments through learning based depth completion," 2025.

[17] M. Walker, Z. Chen, M. Whitlock, D. Blair, D. A. Szafir, C. Heckman, and D. Szafir, "A mixed reality supervision and telepresence interface for outdoor field robotics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 2345–2352.

[18] D. Harabor and A. Grastien, "Improving jump point search," *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 24, no. 1, pp. 128–135, May 2014.

[19] S. Liu, N. Atanasov, K. Mohta, and V. Kumar, "Search-based motion planning for quadrotors using linear quadratic minimum time control," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 2872–2879.

[20] C. Toumieh and D. Floreano, "High-speed motion planning for aerial swarms in unknown and cluttered environments," *IEEE Transactions on Robotics*, 2024.

[21] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," in *Swarm Intelligence*, vol. 7, no. 1. Springer, 2013, pp. 1–41.

[22] S. S. Abdi and D. A. Paley, "Safe operations of an aerial swarm via a cobot human swarm interface," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1701–1707.

[23] A. Hocraffer and C. S. Nam, "A meta-analysis of human-system interfaces in unmanned aerial vehicle (UAV) swarm management," *Applied ergonomics*, vol. 58, pp. 66–80, 2017.

[24] Z. Zhou, P. Wei, Z. Wang, L. Duan, S. Hai, Z. Zhang, Y. Sun, and F. Feng, "Towards human-centered interaction with uav swarms: Framework, system design, and user study," *Design and Artificial Intelligence*, p. 100029, 2025.

[25] S. Chen, M. J. O'Brien, F. Talbot, J. Williams, B. Tidd, A. Pitt, and R. C. Arkin, "Multi-modal user interface for multi-robot control in underground environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 9995–10 002.

[26] M. Walker, T. Phung, T. Chakraborti, T. Williams, and D. Szafir, "Virtual, augmented, and mixed reality for human-robot interaction: A survey and virtual design element taxonomy," *J. Hum.-Robot Interact.*, vol. 12, no. 4, jul 2023.

[27] E. Rosen, D. Whitney, E. Phillips, G. Chien, J. Tompkin, G. Konidaris, and S. Tellex, "Communicating and controlling robot arm motion intent through mixed-reality head-mounted displays," *The International Journal of Robotics Research*, vol. 38, no. 12-13, pp. 1513–1526, 2019.

[28] M. D. Coovert, T. Lee, I. Shindev, and Y. Sun, "Spatial augmented reality as a method for a mobile robot to communicate intended movement," *Computers in Human Behavior*, vol. 34, pp. 241–248, 2014.

[29] O. Erat, W. A. Isop, D. Kalkofen, and D. Schmalstieg, "Drone-augmented human vision: Exocentric control for drones exploring hidden areas," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 4, pp. 1437–1446, 2018.

[30] A. Angelopoulos, A. Hale, H. Shaik, A. Paruchuri, K. Liu, R. Tuggle, and D. Szafir, "Drone brush: Mixed reality drone path planning," in *17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2022, pp. 678–682.

[31] "Webxr device api," World Wide Web Consortium (W3C), Candidate Recommendation Draft W3C CRD WebXR — 20250417, 2025, editors: Brandon Jones, Manish Goregaokar, Rik Cabanier.

[32] K. Wang, Z. Wang, K. Nakagaki, and K. Perlin, ""push-that-there": Tabletop multi-robot object manipulation via multimodal 'object-level instruction'," in *Proceedings of the 2024 ACM Designing Interactive Systems Conference*, ser. DIS '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 2497–2513.

[33] K. Wang, P. Liu, Y. Hu, X. Liu, Z. Wang, and K. Perlin, "A collaborative multimodal xr physical design environment," in *SIGGRAPH Asia 2024 XR*, ser. SA '24. New York, NY, USA: Association for Computing Machinery, 2024.

[34] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[35] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins: A research platform for visual-inertial estimation," in *IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 2020, pp. 4666–4672.

[36] A. Saviolo, N. Picello, J. Mao, R. Verma, and G. Loianno, "Reactive collision avoidance for safe agile navigation," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 16 125–16 132.

[37] A. Millane, H. Oleynikova, E. Wirbel, R. Steiner, V. Ramasamy, D. Tingdahl, and R. Siegwart, "nvblox: Gpu-accelerated incremental signed distance field mapping," *arXiv preprint arXiv:2311.00626*, pp. 1–9, 2024.