2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Drones Guiding Drones: Cooperative Navigation of a Less-Equipped Micro Aerial Vehicle in Cluttered Environments

Václav Pritzl*, Matouš Vrba*, Yurii Stasinchuk*, Vít Krátký*, Jiří Horyna*, Petr Štěpán*, and Martin Saska*

***Abstract*— Reliable deployment of Unmanned Aerial Vehicles (UAVs) in cluttered unknown environments requires accurate sensors for Global Navigation Satellite System (GNSS)-denied localization and obstacle avoidance. Such a requirement limits the usage of cheap and micro-scale vehicles with constrained payload capacity if industrial-grade reliability and precision are required. This paper investigates the possibility of offloading the necessity to carry heavy sensors to another member of the UAV team while preserving the desired capability of the smaller robot intended for exploring narrow passages. A novel cooperative guidance framework offloading the sensing requirements from a minimalistic secondary UAV to a superior primary UAV is proposed. The primary UAV constructs a dense occupancy map of the environment and plans collision-free paths for both UAVs to ensure reaching the desired secondary UAV's goals even in areas not accessible by the bigger robot. The primary UAV guides the secondary UAV to follow the planned path while tracking the UAV using Light Detection and Ranging (LiDAR)-based relative localization. The proposed approach was verified in real-world experiments with a heterogeneous team of a 3D LiDAR-equipped primary UAV and a micro-scale camera-equipped secondary UAV moving autonomously through unknown cluttered GNSS-denied environments with the proposed framework running fully on board the UAVs.**

Multimedia Attachment

`https://mrs.felk.cvut.cz/drones-guiding`

I. Introduction

The ability to accurately perceive the surrounding obstacles and localize in an onboard-built map is crucial for aerial robots operating in unknown cluttered environments. However, such capabilities may require the presence of heavy sensors on board the Unmanned Aerial Vehicles (UAVs), significantly increasing their dimensions. Minimizing such hardware requirements is critical when the UAVs need to be as small as possible to be able to operate in narrow passages.

In many applications, having some micro UAVs with only application-specific payload is desirable. In cooperative sensing tasks, the aim is to distribute UAVs with specific sensors to target positions to detect the sought phenomenon, such as a gas source [1] or radiation [2]. In a confined indoor environment, small UAVs with communication modules may be used to build a mesh communication network, as in the breadcrumb-deploying communication solutions of the DARPA SubT challenge [3]. In Global Navigation Satellite System (GNSS)-denied conditions, UAVs carrying localization anchors can be distributed to specific positions in the environment to provide an external localization system for other robots [4]. In sensory degraded environments, the minimalistic UAVs can serve as landmarks for improving the localization performance of the rest of the robot team [5]. All these tasks benefit from the ability to guide cheap miniature UAVs with only application-specific hardware to target positions or to places untraversable by the larger UAVs, either guiding them all the way or through areas too challenging for their own onboard sensors.

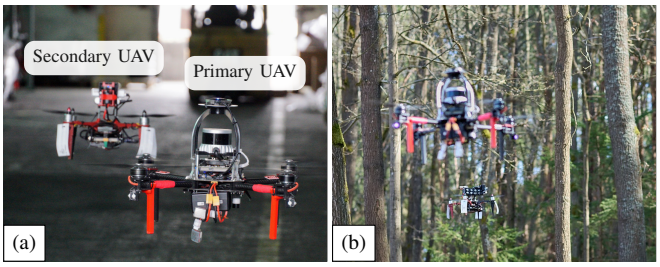

Fig. 1: The LiDAR-equipped primary UAV guides the secondary camera-equipped UAV through (a) narrow passages in an industrial complex, (b) a cluttered forest environment.

In this paper, we focus on offloading the sensing capability from the less-capable UAVs to a more-capable guiding UAV with superior sensors and processing power. Specifically, we deal with the case of a 3D Light Detection and Ranging (LiDAR)-equipped primary UAV guiding a smaller camera-equipped secondary UAV (see Fig. 1). 3D LiDARs are precise and robust to various environmental conditions but are generally heavy. Combining LiDAR-based and visual-based navigation reaps the benefits of both approaches while mitigating their drawbacks.

A. Problem statement

We tackle the problem of cooperative flight of two diverse UAVs in an unknown GNSS-denied cluttered environment. The primary UAV (pUAV) carries a 3D LiDAR sensor. The secondary UAV (sUAV) carries a visual camera for local self-localization only and has insufficient sensing capabilities for precise global localization and obstacle detection. Both UAVs are equipped with an onboard computer, an Inertial Measurement Unit (IMU), an embedded attitude controller, and a wireless communication module. We assume that the 3D LiDAR provides sufficient data for precise localization with low long-term drift w.r.t. the world reference frame,

This work was supported by CTU grant no SGS23/177/OHK3/3T/13, by the Czech Science Foundation (GAČR) under research project No. 23-07517S, and by the European Union under the project Robotics and advanced industrial production (reg. no. CZ.02.01.01/00/22_008/0004590).

*The authors are with the Multi-robot Systems Group, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic {vaclav.pritzl, matous.vrba, yurii.stasinchuk, vit.kratky, jiri.horyna, petr.stepan, martin.saska}@fel.cvut.cz

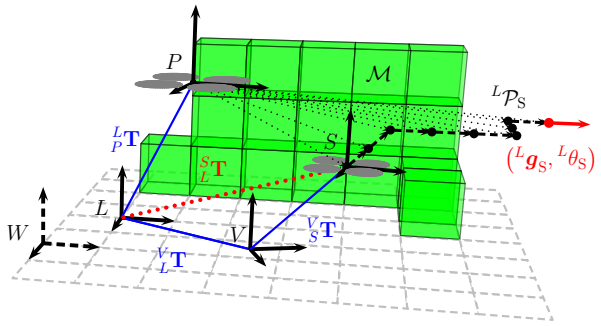

Fig. 2: The pUAV with body frame P is localized in its local frame L, builds dense occupancy map $\mathcal{M}$, and plans collision-free paths for both UAVs. The sUAV with body frame S is localized in its local frame V. W denotes the fixed world frame. All the reference frames are gravity-aligned. The pUAV periodically guides the sUAV to follow the planned path ${}^{L}\mathcal{P}_{\mathrm{S}}$. Black dotted lines mark the line of sight between the pUAV position and the sUAV waypoints.

while the visual camera data is sufficient for short-term localization only, exhibiting larger drift than the LiDAR-based localization. Relying on the LiDAR sensor for self-localization of the pUAV and relative localization between the UAVs enables precise localization for the entire team. The UAVs are capable of mutual communication with the pUAV sending planned paths to the sUAV and the sUAV transmitting its local odometry data to the pUAV. All software runs on board the UAVs with no external computational resources. Both UAVs use only their onboard sensors for localization, and no external localization system is available.

We denote vectors as bold lowercase letters, matrices as bold upright uppercase letters, and frames of reference as uppercase italics letters. Let ${}^{B}_{A}\mathbf{T} \in SE(3)$ be the transformation matrix describing the transformation from frame A to frame B. Let ${}^{A}\boldsymbol{x}$ be a 3D position vector in frame A. We denote sets and sequences by uppercase calligraphic letters. Let ${}^{A}\mathcal{P}_{\mathrm{U}}$ be a sequence of UAV reference poses $({}^{A}\boldsymbol{x}_i, {}^{A}\theta_i)$, consisting of position ${}^{A}\boldsymbol{x}_i \in \mathbb{R}^3$ and heading ${}^{A}\theta_i \in [-\pi, \pi]$, for UAV U in reference frame A, thus forming a path with a specific heading assigned to each position.

The addressed problem, depicted in Fig. 2, is to guide the sUAV to a goal pose $\left({}^{L}\boldsymbol{g}_{\mathrm{S}}, {}^{L}\theta_{\mathrm{S}}\right)$, consisting of position ${}^{L}\boldsymbol{g}_{\mathrm{S}} \in \mathbb{R}^3$ and heading ${}^{L}\theta_{\mathrm{S}} \in [-\pi, \pi]$. Specifically, the task is to plan collision-free paths ${}^{L}\mathcal{P}_{\mathrm{S}}$ and ${}^{L}\mathcal{P}_{\mathrm{P}}$ based on the 3D occupancy map $\mathcal{M}$ built by the pUAV and guide the sUAV along the planned path ${}^{L}\mathcal{P}_{\mathrm{S}}$, while ${}^{L}\mathcal{P}_{\mathrm{P}}$ is planned such that the pUAV tracks the sUAV throughout the environment.

B. Related work

Utilizing heterogeneous multi-robot teams consisting of more-capable robots guiding less-capable robots has been proposed for use with Unmanned Ground Vehicles (UGVs) in the past [6]–[9]. Works in the area of UGV-UAV cooperation have focused on the cooperation of a UGV with a UAV acting as an "eye in the sky" [10]. The UAV can map the surroundings of the UGV and plan a collision-free path for the UGV to follow [11]–[14]. The path planning can be performed directly on the image data [15], [16] or as a Partially Observable Markov Decision Process (POMDP) [17] without global map construction. In [18], the UAV sent only target positions to the UGV, which performed motion planning on its own sensory data. In [19], a terrain map was constructed on a ground station using a learning-based approach and sent to the UGV, which used it for path planning. In [20], the UAV shared its voxel map of the environment with a UGV, which converted the voxel map to an elevation map and utilized it for path planning. In [21], a UAV detected a UGV along with obstacles on the ground and shared the data with the UGV to aid the UGV's path planning. In [22], a framework for a single UAV guiding a group of blind UGVs among obstacles was proposed. In [10]–[13], [16]–[21], the authors either assume apriori alignment of the UAV-UGV reference frames or utilize camera-based relative localization. However, camera-based localization may suffer from inaccurate distance estimation of the detected teammate or obstacles, as noted in [21]. While its accuracy is sufficient for guiding a UGV among obstacles on the ground, guiding a UAV through 3D space requires more accurate localization. In [15], an Ultra-wideband (UWB)-based localization system was used, but such a system requires apriori-placed anchors in the environment. In [14], matching visual landmarks in shared maps was employed, but sharing map data requires significant communication bandwidth. The authors of [22] utilized relative pose estimation from detections of infrared LED markers, UWB, and IMU data, but such an approach requires a specialized localization module on each robot. All of the aforementioned UAV-UGV cooperation approaches assume that the UAV is moving above the obstacles and does not need to solve obstacle avoidance nor maintaining line-of-sight (LOS) of the teammates w.r.t. the obstacles.

In [23], a trajectory optimization method for a 3D LiDAR-carrying UGV guiding a camera-equipped UAV while keeping visibility between obstacles was proposed. The approach was evaluated mainly in simulations and localization details were not provided. The obstacle avoidance and occlusion mitigation were treated as 2D problems, while we solve the obstacle avoidance in 3D and occlusion mitigation in 2D. Finally, in a UAV-UGV system, the UAV can safely fly over a UGV. In a multi-UAV system, the UAVs need to maintain sufficient altitude separation to avoid propeller downwash.

Regarding multi-UAV cooperation, the closest methods [24]–[27] to our approach used a leader-follower scheme with one UAV moving through the environment and the other UAVs tracking the leader. In [24], the navigation of a main-agent was aided by a monocular camera-equipped sub-agent. The approach did not consider obstacles and was evaluated in simulations only. A leader-follower scheme was applied in trajectory generation methods for tracking a UAV among obstacles using a camera [25], [26], and in a decentralized system for tracking a target UAV by a UAV swarm among obstacles, using LiDAR-based relative localization [27].

The aforementioned leader-follower-based studies [24]–[27] dealt with UAVs that are equal in size. They would

fail in mixed-size UAV team scenarios requiring a miniature UAV with limited or absent obstacle-sensing capability to pass through a narrow gap untraversable for the leader. In such scenarios, which match inspection and data-gathering tasks in hard-to-reach areas, the obstacle avoidance and visibility-maintaining constraints conflict as the leading UAV cannot directly lead a secondary UAV through narrow openings. Thus, the leader must guide the UAV from a distance using communication and precise relative localization.

To obtain high-accuracy relative localization, necessary for our guiding approach, we utilize LiDAR-based relative localization, similarly as in [27], but focus on a heterogeneous UAV team where only the pUAV is carrying a 3D LiDAR while the minimalistic sUAV carries a camera providing data for Visual-Inertial Odometry (VIO). We build upon our previous works on the use of LiDAR-based UAV detections [28] and fusion of these detections with VIO [29] for guiding a cooperating UAV. To the best of our knowledge, our approach marks the first deployment of a heterogeneous UAV team offloading obstacle detection and motion planning from one UAV to another. The contributions of this paper are summarized as:

- A novel approach for flight of a heterogeneous UAV team in cluttered environments, offloading occupancy mapping and path planning requirements to a more-capable pUAV while preserving the obstacle avoidance capabilities and precise localization of the less-capable sUAV. The framework ensures accurate guiding of the sUAV with minimal communication requirements.
- A unique cooperative relative-localization-aware path planning method for a mixed-size UAV team. Collision-free paths are planned for the sUAV, while the pUAV motion is planned to maximize LOS visibility between the UAVs before a LOS break, which may be unavoidable due to their different sizes.

The proposed methodology was evaluated in realistic simulations as well as in demanding real-world deployments while running on board the UAVs with no external computational resources and no external localization system.

II. MULTI-UAV PLANNING AND GUIDANCE

A high-level diagram of the proposed multi-UAV framework is shown in Fig. 3. The *Multi-UAV path planner* module receives high-level sUAV goals $\left({}^{L}\boldsymbol{g}_{\mathrm{S}}, {}^{L}\theta_{\mathrm{S}}\right)$ and constructs collision-free paths for both the sUAV and the pUAV itself. The sUAV path is periodically transformed to the sUAV body frame S using the relative localization data and transmitted to the sUAV over the wireless network. During the guiding process, the pUAV-sUAV pair essentially forms a closed loop controlling the sUAV pose in the local reference frame L. The pUAV is localized using a 3D LiDAR-based Simultaneous Localization and Mapping (SLAM) and performs dense occupancy mapping using the Octomap [31] approach, producing an occupancy map $\mathcal{M}$ of the environment. To decrease the processing requirements, the map $\mathcal{M}$ represents a limited region of the environment, centered at the current pUAV position. The sUAV uses VIO for self-localization. Generating time-parametrized trajectories from planned paths and position control in the frames of the respective self-localization methods is provided by the MRS UAV System[1] [30] running on each UAV. The system times of the UAVs are synchronized over the wireless network using the Network Time Protocol (NTP).

A. Proposed multi-UAV path planner

Algorithm 1 Multi-UAV path planning (state = *PLANNING*)

Input: Map $\mathcal{M}$, pUAV pose $\left({}^{L}\boldsymbol{x}_{\mathrm{P}}, {}^{L}\phi_{\mathrm{P}}\right)$, sUAV pose $\left({}^{L}\boldsymbol{x}_{\mathrm{S}}, {}^{L}\phi_{\mathrm{S}}\right)$, sUAV goal pose $\left({}^{L}\boldsymbol{g}_{\mathrm{S}}, {}^{L}\theta_{\mathrm{S}}\right)$
Output: Paths ${}^{L}\mathcal{P}_{\mathrm{P}}, {}^{L}\mathcal{P}_{\mathrm{S}}$ sent to the UAVs' control pipelines
Parameters: pUAV obstacle width w_{P} and height h_{P}, sUAV obstacle width w_{S} and height h_{S}, pUAV minimum obstacle distance d_{P}, sUAV minimum obstacle distance d_{S}

if $\left|\left|{}^{L}\boldsymbol{x}_{\mathrm{S}} - {}^{L}\boldsymbol{g}_{\mathrm{S}}\right|\right|_2 < \mathcal{M}.resolution$ **then**
  CHANGE_STATE(*GOAL REACHED*)
  return
$\mathcal{M}_{\mathrm{S}} \leftarrow$ COPY_MAP_AND_ADD_OBSTACLE($\mathcal{M}$, ${}^{L}\boldsymbol{x}_{\mathrm{P}}$, w_{P}, h_{P})
${}^{L}\mathcal{P}_{\mathrm{S}} \leftarrow$ FIND_PATH($\mathcal{M}_{\mathrm{S}}$, ${}^{L}\boldsymbol{x}_{\mathrm{S}}$, ${}^{L}\phi_{\mathrm{S}}$, ${}^{L}\boldsymbol{g}_{\mathrm{S}}$, ${}^{L}\theta_{\mathrm{S}}$, d_{S}) ▷ Using A* planner
if ${}^{L}\mathcal{P}_{\mathrm{S}} = \varnothing$ **then**
  CHANGE_STATE(*FAILURE*)
  return
${}^{L}\boldsymbol{g}_{\mathrm{P}} \leftarrow$ FIND_GUIDING_VIEWPOINT($\mathcal{M}$, ${}^{L}\mathcal{P}_{\mathrm{S}}$, ${}^{L}\boldsymbol{x}_{\mathrm{P}}$) ▷ Alg. 2
if ${}^{L}\boldsymbol{g}_{\mathrm{P}} = \varnothing$ **then**
  CHANGE_STATE(*FAILURE*)
  return
if $\left|\left|{}^{L}\boldsymbol{x}_{\mathrm{P}} - {}^{L}\boldsymbol{g}_{\mathrm{P}}\right|\right|_2 < \mathcal{M}.resolution$ **then**
  CHANGE_STATE(*SECONDARY MOVING*, ${}^{L}\mathcal{P}_{\mathrm{S}}$)
  return
$\mathcal{M}_{\mathrm{P}} \leftarrow$ COPY_MAP_AND_ADD_OBSTACLE($\mathcal{M}$, ${}^{L}\boldsymbol{x}_{\mathrm{S}}$, w_{S}, h_{S})
${}^{L}\mathcal{P}_{\mathrm{P}} \leftarrow$ FIND_PATH($\mathcal{M}_{\mathrm{P}}$, ${}^{L}\boldsymbol{x}_{\mathrm{P}}$, ${}^{L}\phi_{\mathrm{P}}$, ${}^{L}\boldsymbol{g}_{\mathrm{P}}$, ${}^{L}\theta_{\mathrm{P}}$, d_{P}) ▷ Using A* planner
if ${}^{L}\mathcal{P}_{\mathrm{P}} = \varnothing$ **then**
  CHANGE_STATE(*FAILURE*)
  return
CHANGE_STATE(*PRIMARY MOVING*, ${}^{L}\mathcal{P}_{\mathrm{P}}$)
return

The *Multi-UAV path planner* module consists of a Finite-State Machine (FSM) transferring between the *IDLE*, *PLANNING*, *PRIMARY MOVING*, *SECONDARY MOVING*, *GOAL REACHED*, and *FAILURE* states. When the module receives a new goal $\left({}^{L}\boldsymbol{g}_{\mathrm{S}}, {}^{L}\theta_{\mathrm{S}}\right)$ for the sUAV, the FSM transfers to the *PLANNING* state and constructs collision-free paths ${}^{L}\mathcal{P}_{\mathrm{P}}, {}^{L}\mathcal{P}_{\mathrm{S}}$ for both UAVs. In the *PRIMARY MOVING* state, the pUAV moves along its path to a viewpoint ${}^{L}\boldsymbol{g}_{\mathrm{P}}$, where it can observe and guide the sUAV along ${}^{L}\mathcal{P}_{\mathrm{S}}$. After reaching the viewpoint, the FSM transfers to the *SECONDARY MOVING* state. In the *SECONDARY MOVING* state, the pUAV guides the sUAV to follow the planned collision-free path ${}^{L}\mathcal{P}_{\mathrm{S}}$. When the goal is reached, the FSM transfers to the *GOAL REACHED* state.

The path planning process proceeds according to Alg. 1. The planner takes the current occupancy map $\mathcal{M}$ and inserts an occupied region at the current position ${}^{L}\boldsymbol{x}_{\mathrm{P}}$ of the

[1] https://github.com/ctu-mrs/mrs_uav_system

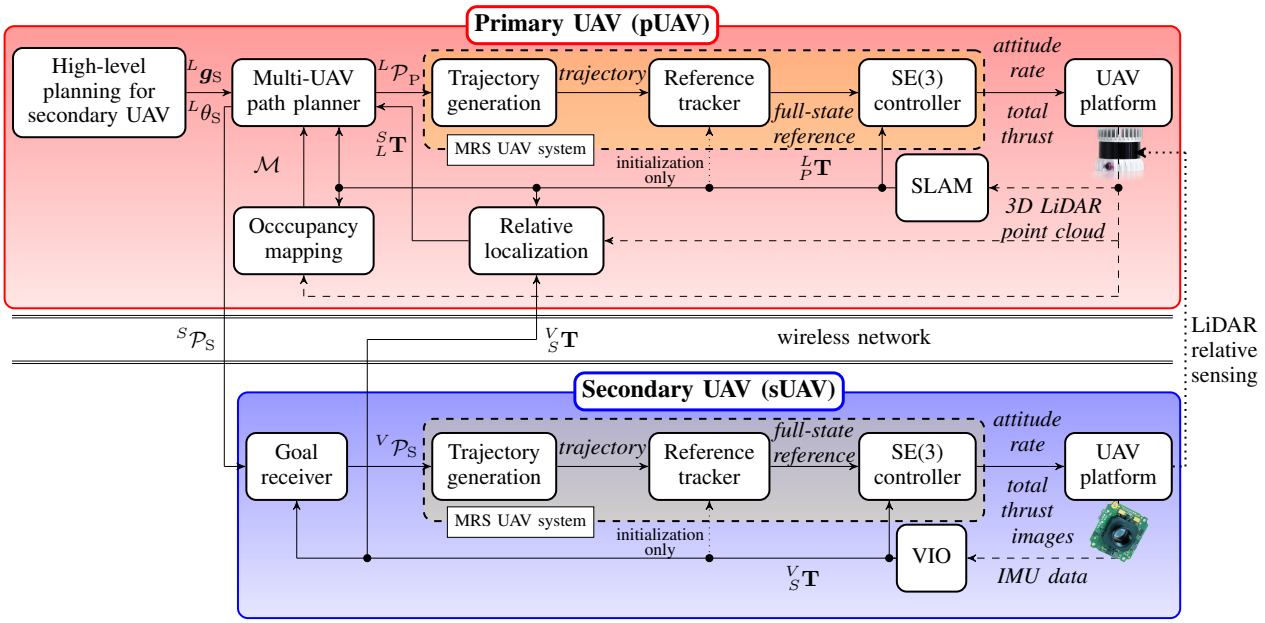

Fig. 3: The pUAV maps the surrounding environment using 3D LiDAR data, performs estimation of relative pose of the sUAV from LiDAR detections and received VIO data, and guides the sUAV to desired poses. The position control pipeline in the frame of each respective self-localization method is provided by the MRS UAV System [30].

pUAV with safety margin above and below to avoid the downwash effect. A collision-free path ${}^{L}\mathcal{P}_{\mathrm{S}}$ for the sUAV is planned using an A* planner. We have utilized the A* planning algorithm with iterative path post-processing that we designed for the DARPA SubT challenge, described in [3], [32]. Based on the resulting path, a goal viewpoint ${}^{L}\boldsymbol{g}_{\mathrm{P}}$ for the pUAV is selected to maximize the uninterrupted sequence of waypoints, beginning at ${}^{L}\boldsymbol{x}_{\mathrm{S}}$, visible by the pUAV (see sec. II-A.1 for details). Occupied space at the position ${}^{L}\boldsymbol{x}_{\mathrm{S}}$ of the sUAV is inserted to the map and the A* planner constructs a collision-free path ${}^{L}\mathcal{P}_{\mathrm{P}}$ for the pUAV.

1) Finding a guiding viewpoint: To maximize LOS visibility throughout the guiding process, a viewpoint ${}^{L}\boldsymbol{g}_{\mathrm{P}}$ for the pUAV is computed (see Alg. 2 and Fig. 4), such that it fulfills the following conditions:

1) The pUAV has LOS visibility of as many sUAV waypoints from the path beginning as possible. The areas with LOS visibility of the specific sUAV waypoints are represented as a set of non-convex polygons $\mathcal{V}_{\mathrm{all}}$.
2) The closest obstacle distance is larger than a threshold d_{P}. This creates a safe region, represented as a multi-polygon (collection of non-convex polygons with holes and non-intersecting boundaries) $\mathcal{S}$.
3) The distance to the sUAV path ${}^{L}\mathcal{P}_{\mathrm{S}}$ is larger than a desired safety threshold d_{buffer}. The safety buffer around the sUAV path is represented as a non-convex polygon $\mathcal{B}$.

Alg. 2 constructs 2D polygons representing the visibility regions $\mathcal{V}_{\mathrm{all}}$ by performing raycasting from each waypoint $\boldsymbol{p}_i$ to the surrounding voxels at the same altitude. The algorithm calculates the difference of the visibility regions and the safety buffer $\mathcal{B}$, finds the closest safe region $\mathcal{S}_{\mathrm{closest}}$ within $\bigcup\mathcal{V}_{\mathrm{all}}$, and looks for the intersection of $\mathcal{S}_{\mathrm{closest}}$ with as

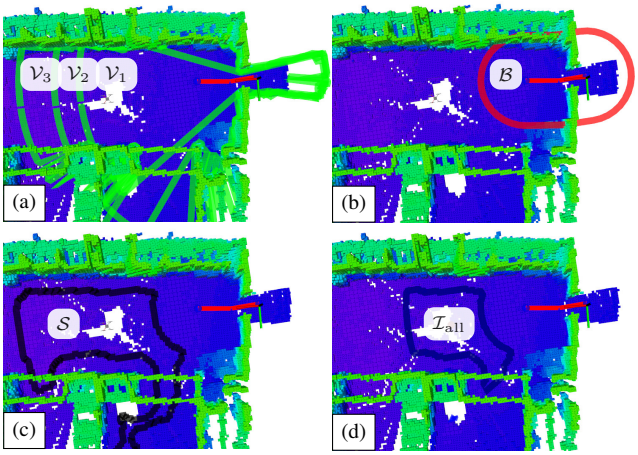

Fig. 4: Example of guiding viewpoint selection. sUAV path (red line) consists of 3 waypoints passing through the door in the upper right corner of the map. The algorithm constructs regions from which the waypoints are visible (a), a safety buffer around the sUAV path (b), safe space w.r.t. obstacles for the pUAV (c), and the final intersection $\mathcal{I}_{\mathrm{all}}$ (d).

many consecutive visibility regions $\mathcal{V}_{0\ldots k}$ as possible. If such an intersection exists, the viewpoint ${}^{L}\boldsymbol{g}_{\mathrm{P}}$ is selected as the closest one from the interior points furthest from the boundaries, i.e., the poles of inaccessibility (function POI() in Alg. 2), of the obtained intersection (the intersection may be a collection of polygons). The polygonal operations were implemented using the *boost::geometry* C++ library[2].

2) Guiding: The *Multi-UAV path planner* transforms the sUAV path ${}^{L}\mathcal{P}_{\mathrm{S}}$ to the sUAV body frame S and removes all points ${}^{S}\boldsymbol{p}_i \in {}^{S}\mathcal{P}_{\mathrm{S}}$ such that $\left|\left|{}^{S}\boldsymbol{p}_i\right|\right|_2 < \delta$, where δ

[2]https://www.boost.org/libs/geometry/

Algorithm 2 Finding a guiding viewpoint

Input: Map $\mathcal{M}$, pUAV position ${}^{L}\boldsymbol{x}_{\mathrm{P}}$, sUAV path ${}^{L}\mathcal{P}_{\mathrm{S}}$
Output: Goal ${}^{L}\boldsymbol{g}_{\mathrm{P}}$ for the pUAV
Parameters: raycasting sample count n_{samples}, raycasting max. ray length d_{ray}, min. safe distance d_{buffer} from the sUAV path, pUAV min. obstacle distance d_{P}

▷ Construct a safety buffer around the UAV path
$\mathcal{B} \leftarrow \mathrm{BUFFER}({}^{L}\mathcal{P}_{\mathrm{S}}, d_{\mathrm{buffer}})$ ▷ Single non-convex polygon
▷ Get a visibility region for each point in the path
$\mathcal{V}_{\mathrm{all}} \leftarrow \varnothing$ ▷ Set of multi-polygons
for $(\boldsymbol{p}_i, \theta_i) \in {}^{L}\mathcal{P}_{\mathrm{S}}$ **do**
  $\mathcal{V}_i \leftarrow \varnothing$ ▷ Single non-convex polygon
  for $\alpha \in \left(k\frac{2\pi}{n_{\mathrm{samples}}}\right)_{k=0}^{n_{\mathrm{samples}}}$ **do**
    $\boldsymbol{r}_{\mathrm{end}} \leftarrow \boldsymbol{p}_i + [d_{\mathrm{ray}}\cos\alpha, d_{\mathrm{ray}}\sin\alpha, 0]^{\mathrm{T}}$
    ▷ get the first intersection with obstacle or $\boldsymbol{r}_{\mathrm{end}}$
    $\boldsymbol{r}_{\mathrm{int}} \leftarrow \mathrm{RAY\_OBSTACLE\_INTERSECT}(\mathcal{M}, \boldsymbol{p}_i, \boldsymbol{r}_{\mathrm{end}})$
    $\mathcal{V}_i.\mathrm{add}(\boldsymbol{r}_{\mathrm{int}})$
  $\mathcal{V}_{\mathrm{all}}.\mathrm{add}(\mathcal{V}_i \setminus \mathcal{B})$
▷ Generate safe space region
$\mathcal{S} \leftarrow \varnothing$ ▷ Multi-polygon (collection of polygons)
$r \leftarrow \mathcal{M}.resolution$
for $v \in \mathcal{M}[v_z = {}^{L}x_{\mathrm{P}z}]$ **do** ▷ v is a single voxel
  if $\mathrm{OBS\_DIST}(v) > d_{\mathrm{P}}$ **and** $v \in \bigcup \mathcal{V}_{\mathrm{all}}$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \mathrm{POLYGON}(v_x \pm \frac{r}{2}, v_y \pm \frac{r}{2})$ ▷ Do union with safe space region, adjacent polygons are joined
$\mathcal{S}_{\mathrm{closest}} \leftarrow \arg\min_{\mathcal{S}_i \in \mathcal{S}} \mathrm{DISTANCE}({}^{L}\boldsymbol{x}_{\mathrm{P}}, \mathcal{S}_i)$
▷ Find intersection of safe region and the visibility regions
$\mathcal{I}_{\mathrm{all}} \leftarrow \mathcal{S}_{\mathrm{closest}}$
$i_{\mathrm{found}} \leftarrow \mathbf{false}$
for $\mathcal{V}_i \in \mathcal{V}_{\mathrm{all}}$ **do** ▷ Ordered same as ${}^{L}\mathcal{P}_{\mathrm{S}}$, start from $\mathcal{V}_0$
  $\mathcal{I}_{\mathrm{new}} \leftarrow \mathcal{I}_{\mathrm{all}} \cap \mathcal{V}_i$ ▷ Intersection is a multi-polygon
  if $\mathcal{I}_{\mathrm{new}} = \varnothing$ **then** ▷ No more intersections found
    break
  else
    $\mathcal{I}_{\mathrm{all}} \leftarrow \mathcal{I}_{\mathrm{new}}$
    $i_{\mathrm{found}} \leftarrow \mathbf{true}$
if i_{found} **then**
▷ Select goal as the closest pole of inaccessibility
  return $\mathrm{POI}\left(\arg\min_{\mathcal{I}_i \in \mathcal{I}_{\mathrm{all}}} \left|\left|\mathrm{POI}(\mathcal{I}_i) - {}^{L}\boldsymbol{x}_{\mathrm{P}}\right|\right|_2\right)$
else
  return $\varnothing$

is a distance threshold for considering the path waypoint as visited. The remaining path is transmitted to the sUAV over the wireless network and saved for the next guiding step. It is worth mentioning that the planned path could instead be transmitted in the original frame L along with the transformation ${}^{V}_{L}\mathbf{T}$ and transformed on board the sUAV. Transforming the path on board the pUAV minimizes the requirements imposed on the sUAV, making the approach compatible with a wide range of UAVs. The only requirements on the sUAV are the capability to transmit its local odometry in the local frame V and follow paths specified relatively to its current pose.

B. Relative pose estimation

Accurate relative pose estimation between the UAVs is crucial for reliable performance of the guiding approach, and uncertainty in the relative pose represents one of the possible sources of guiding errors. The estimation approach needs to accurately provide the transformation ${}^{S}_{L}\mathbf{T}$ required for transforming the reference path from the local frame L to the sUAV body frame S. In the proposed approach, we apply relative pose estimation based on the fusion of LiDAR detections with VIO data received over the wireless network. The approach for UAV detection from LiDAR data is described in [33] in detail. The LiDAR detections provide precise 3D positions of the sUAV, and the fusion with VIO data provides orientation of the sUAV and keeps track of the sUAV when the detections are lost. The fusion approach is described in [29] in detail and does not require any apriori knowledge about the UAV positions nor orientations. The relative transformation ${}^{V}_{L}\mathbf{T}$ is obtained by solving a Non-linear Least Squares (NLS) problem aligning corresponding UAV trajectories observed in a sliding window of the LiDAR detections and VIO data. The periodically-updated transformation ${}^{V}_{L}\mathbf{T}$, LiDAR-based detections, and VIO data are utilized in a Kalman-filter-based approach to estimate the transformation ${}^{S}_{L}\mathbf{T}$. The UAV detection approach [33] was improved by utilizing reflective markers on the legs of the sUAV. The employed 3D LiDAR sensor provides the reflectivity of each measured data point, hence a thresholding-based filter was applied to filter out possible false detections, and thus increase the reliability of UAV detections in cluttered environments.

III. EXPERIMENTAL VERIFICATION

Two different quadrotor platforms (see Fig. 1) were utilized in the experiments. The pUAV was built upon the Holybro X500 frame and was $0.7\,\mathrm{m} \times 0.7\,\mathrm{m}$ wide including propellers. The sUAV was built upon the DJI F330 frame and was $0.45\,\mathrm{m} \times 0.45\,\mathrm{m}$ wide including propellers. Both UAVs carried the Intel NUC 10i7FNH onboard computer with the Intel Core i7 10710U CPU, 16 GB of RAM, and a Wi-Fi module. Both UAVs used the Pixhawk 4 flight controller. The pUAV was equipped with the Ouster OS0-128 Rev C 3D LiDAR. The LiDAR has a 360° horizontal and 90° vertical Field Of View (FOV) and produces scans with a resolution of 1024 × 128 beams at a rate of 10 Hz. For mapping, the LiDAR scans were downsampled by a factor of 4 to ensure the real-time performance of the entire software stack on board the UAVs. Detailed information about the UAV hardware is available in [34], [35].

The pUAV utilized the LOAM algorithm [36] for self-localization. The sUAV carried a front-facing RealSense T265 tracking camera and utilized its stereo fisheye image and IMU output for VIO using the OpenVINS algorithm [37]. The sUAV was equipped with reflective markers on its legs to aid the LiDAR-based relative localization. The software on board both UAVs was based on Ubuntu 20.04, Robot Operating System (ROS) 1, and the MRS UAV system [30]. Both UAVs were connected to a single Wi-Fi access point for mutual communication, and the Nimbro network was used for transporting the ROS topics over the wireless network. The system time of the UAVs was synchronized using the *chrony* implementation of the NTP.

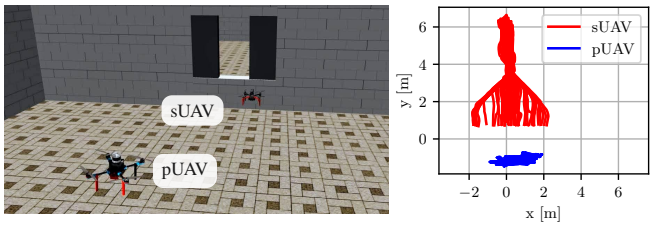

Fig. 5: The simulation with a gap of varying width and UAV trajectories from the test of the proposed guiding approach.

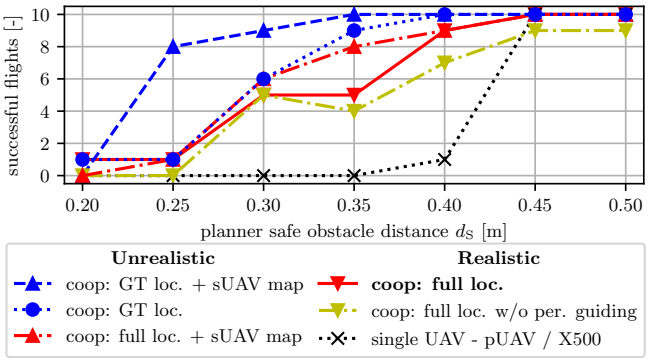

Fig. 6: The number of successful flights through the gap depending on the planner's distance d_S for each simulated configuration. The legend is ordered based on the total number of successful flights. *Single UAV* represents a UAV carrying a 3D LiDAR. *Coop* represents guiding sUAV by pUAV. *GT loc.* means ground-truth self-localization and ground-truth relative localization. *Full loc.* denotes localization from fully simulated onboard sensors. *sUAV map* means guiding the sUAV by pUAV using a map constructed from a 3D LiDAR on board the sUAV (in reality too heavy to be carried by the sUAV). *w/o per. guiding* means sending the path to the sUAV only once, without periodic guiding.

A. Simulated evaluation of guiding accuracy

The main factors influencing the guiding accuracy were identified in a simulation study using a two-room environment connected by a gap with varying width built in the Gazebo simulator (see Fig. 5). The gap width was set to

$$w_G = 2 \times d_S + 1.5 \times \mathcal{M}.resolution, \quad (1)$$

where d_S is the planner's safe distance for sUAV and $\mathcal{M}.resolution = 0.1\,\mathrm{m}$ is the voxel size of the map. Multiple configurations of the framework were compared. For each configuration, the ability to fly through the gap was tested for a set of planner's safe distances d_S. For each distance d_S, 10 runs of simulation with randomized initial UAV positions were done, randomizing the direction of the necessary UAV movement towards the gap and the alignment of the map grid with the environment. A simulation run was considered successful if the sUAV flew along planned paths through the gap and back to predefined goal position without colliding with the obstacles.

Fig. 6 compares the number of successful flights per safe distance d_S among the tested configurations. The effect of localization errors was compared on guiding with ground-truth localization and full localization using LOAM [36], OpenVINS [37], and LiDAR detections [29], identical to the real-world setup. The effect of mapping errors was compared on guiding using a map constructed on board sUAV (from onboard 3D LiDAR too heavy to be carried in reality), and a map constructed on board the pUAV. The configuration *coop: full loc.* in Fig. 6 corresponds to the real-world setup.

The guiding accuracy declines with increasing inaccuracy of mapping caused by the increase of the gap between adjacent LiDAR rays (*GT loc.* vs *GT loc. + sUAV map* in Fig. 6). The mutual UAV distance was approximately $5\,\mathrm{m}$ when the sUAV was passing through the gap. At the distance of $5\,\mathrm{m}$, the gap between adjacent LiDAR rays in the horizontal plane is $0.12\,\mathrm{m}$, assuming the 256 horizontal samples of the downsampled LiDAR scan. The LiDAR-based relative localization accuracy also depends on the mutual UAV distance. The Mean Absolute Error (MAE) of sUAV localization based on the LiDAR detections within $1\,\mathrm{m}$ from the center of the gap was $0.10\,\mathrm{m}$, and the mean error in the axis parallel to the gap was $0.054\,\mathrm{m}$. By comparing the *coop: GT loc. + sUAV map* configuration and the *coop: full loc.* configuration, we see that the distance d_S for which the sUAV was able to achieve 90% success rate increased by $0.1\,\mathrm{m}$ due to these mapping and localization errors.

The *coop: full loc. w/o per. guiding* configuration shows that sending the path to the sUAV only once without periodic guiding results in decreased performance due to localization errors. The *single UAV* configuration shows the gap width necessary for the pUAV to pass through. The simulations have proven the viability of guiding the sUAV by the pUAV, showing that the sUAV can be guided through much smaller gaps than the pUAV itself would be able to pass through.

B. Real-world cooperative flight through a forest

In the next set of experiments, the pUAV was tasked to guide the sUAV through a forest (see Fig. 7 and the video attachment). The guiding process was triggered repeatedly, and the sUAV goal pose was generated by incrementing the sUAV pose by $4\,\mathrm{m}$ in the direction of the x-axis each time the guiding process finished (switched to the *GOAL REACHED* state). Table I contains parameters of the Algorithms 1 and 2 utilized in the forest flights. Five separate flights were performed. Table II contains the distances traveled during these flights. The UAVs performed all of these flights successfully, reaching the border of the allowed operation space. Fig. 8 shows the occupancy map built during one of the flights along with the trajectories traversed by the two UAVs.

The median processing time of the entire Alg. 1 was $8.90\,\mathrm{s}$. The median processing times of the individual steps were $0.89\,\mathrm{s}$ for the guiding viewpoint selection (Alg. 2, Alg. 1 line 9), $0.60\,\mathrm{s}$ for pUAV path planning (Alg. 1 line 17), $5.33\,\mathrm{s}$ for sUAV path planning (Alg. 1 line 5), and the rest of the time was spent copying the occupancy maps and inserting occupied space into them. The processing time of pUAV path planning was much lower since the pUAV goal was always located in a safe space. In contrast, the sUAV goal was calculated by incrementing the sUAV pose and thus

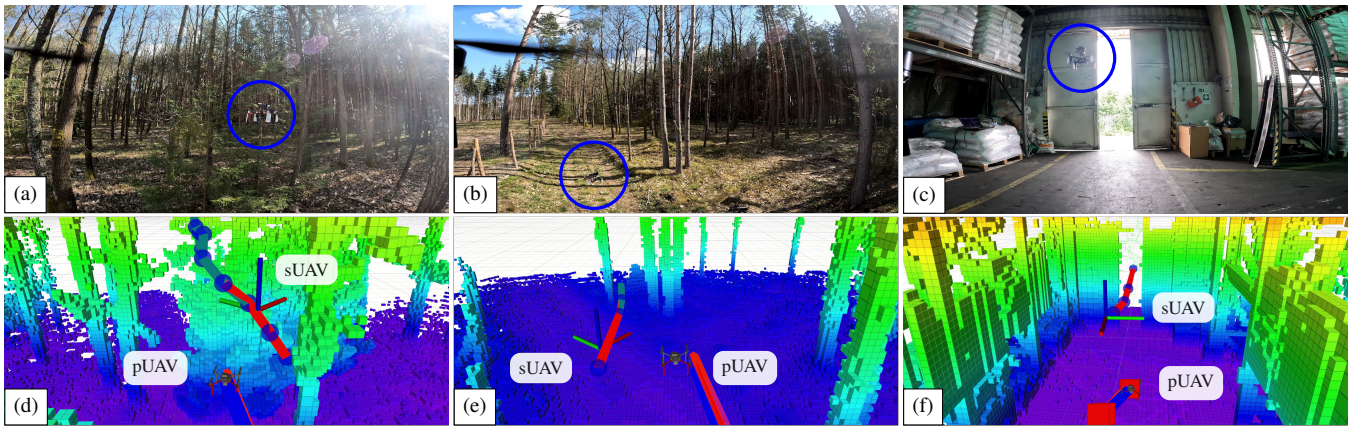

Fig. 7: Real-world experiments: images from the pUAV onboard camera (a-c) and visualization of the occupancy map and planned paths at the corresponding moments from similar views behind the pUAV (d-f). The sUAV is marked by blue circles.

TABLE I: Parameters from the forest flight experiments.

Parameter	Symbol	Value
voxel size	$\mathcal{M}.resolution$	0.1 m
pUAV safe distance	d_{P}	0.9 m
sUAV safe distance	d_{S}	0.8 m
pUAV occ. space width	w_{P}	1.5 m
pUAV occ. space height	h_{P}	10 m
sUAV occ. space width	w_{S}	1.3 m
sUAV occ. space height	h_{S}	10 m
raycasting sample count	n_{samples}	500
raycasting max length	d_{ray}	6 m
min. safe distance from sUAV path	d_{buffer}	2 m

TABLE II: 2D straight-line distances traveled from the start of the guiding process by the sUAV in the forest. The distances were calculated from the relative localization data.

Flight number	1	2	3	4	5
Distance traveled [m]	31.6	36.5	35.1	41.6	35.0

often laid too close to an obstacle, causing the planner to search for a path until a timeout threshold was reached. In a practical application, a feasible goal can be pre-selected, or the planning timeout can be decreased to reduce the planning time. For safety reasons, the planning process was performed only while the UAVs were hovering, but it can be easily triggered while the sUAV is still moving.

C. Real-world guiding through a narrow gap

The last set of experiments, performed in an industrial warehouse environment, aims to evaluate the performance of the proposed framework in navigation of the sUAV through extremely narrow openings. The pUAV was tasked to guide the sUAV through a narrow gap whose width was gradually decreased together with the safety distance of the path planner. The algorithms utilized the same parameters as in Table I, except for the sUAV safe distance d_{S}, which was decreased according to the width of the gap. Both UAVs started flying inside the warehouse, and the sUAV was guided through the narrow gap out of the building and back multiple times. Table III shows the results of this experiment. The proposed guiding approach reached its limit when passing through a $0.9\,\mathrm{m}$ wide gap when the sUAV collided with the door during the fifth attempted pass due to an inaccuracy in the relative localization. The results correspond to the performance observed in the simulations (see Fig. 6).

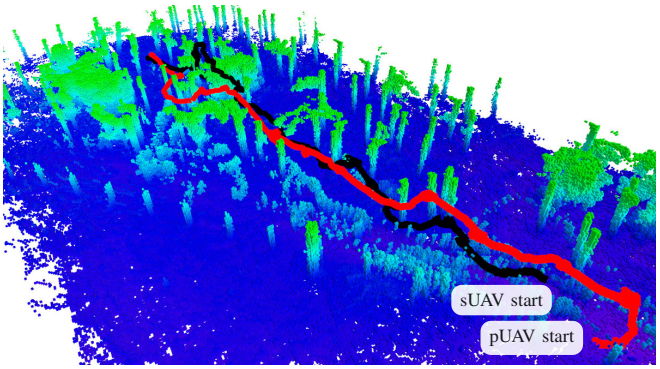

Fig. 8: Occupancy map from the forest flight and the UAV trajectories (pUAV - red, sUAV - black). For clarity, the map was sliced at the maximum height of $4\,\mathrm{m}$ above the ground.

TABLE III: Results of guiding the sUAV through narrow gaps with varying width. A single pass represents flight through the gap either out of the building or into the building.

Gap width [m]	1.2	1.1	1.0	0.9
sUAV safe distance d_{S} [m]	0.5	0.45	0.4	0.4
Successful / attempted passes [-]	2/2	2/2	6/6	4/5

D. Communication analysis

Table IV analyzes the wireless communication bandwidth required by the guiding approach. The table shows message sizes recorded during the real-world experiments and the required maximal and minimal bandwidth calculated based on the transmission rates. The data transmitted from the sUAV to pUAV consisted of the VIO output ${}^{V}_{S}\mathbf{T}$. The data sent from the sUAV to pUAV were the desired paths ${}^{S}\mathcal{P}_{\mathrm{S}}$.

The sizes of the path messages depend on the number of contained points. Fig. 9 shows histograms of the message sizes from real-world experiments. The communication bandwidth utilized by the guiding approach is minimal and leaves enough capacity for application-specific sensory data or commands for guiding multiple sUAVs.

TABLE IV: Analysis of the wireless communication requirements during the real-world experiments.

		sUAV→pUAV odometry	pUAV→sUAV path
Rate [Hz]		2	5
Message size [KB]	min	0.725	0.132
	max	0.725	0.484
Bandwidth [KBps]	min	1.45	0.66
	max	1.45	2.42

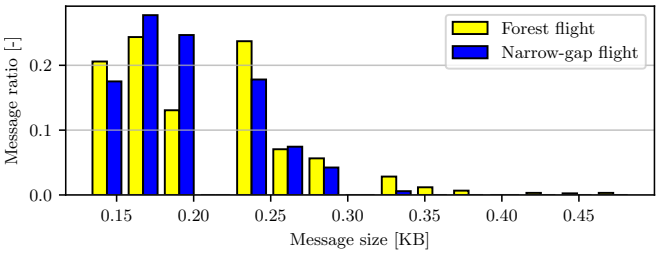

Fig. 9: Path message sizes during the real-world experiments.

IV. CONCLUSIONS

A novel cooperative guiding approach enabling offloading the necessity to carry accurate but large and heavy sensors from a miniature UAV to another member of a UAV team while preserving the desired obstacle avoidance capabilities and reliable localization was proposed in this paper. To achieve such tight cooperation, the more-capable primary UAV performs occupancy mapping of the environment, relative localization of the less-capable secondary UAV, planning of collision-free paths for both the secondary UAV and itself, and guidance of the secondary UAV through the environment. To achieve reliable cooperative navigation in narrow passages, a reliable relative-localization-aware planner was proposed and integrated into the complex system. The feasibility of the proposed approach was demonstrated in multiple real-world experiments in unknown cluttered GNSS-denied environments with all the algorithms running on board the UAVs with no external localization system nor external computational resources utilized.

REFERENCES

[1] B. P. Duisterhof *et al.*, "Sniffy Bug: A Fully Autonomous Swarm of Gas-Seeking Nano Quadcopters in Cluttered Environments," in *IEEE IROS*, 2021.

[2] P. Štibinger *et al.*, "Localization of ionizing radiation sources by cooperating micro aerial vehicles with pixel detectors in real-time," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3634–3641, 2020.

[3] M. Petrlik *et al.*, "UAVs Beneath the Surface: Cooperative Autonomy for Subterranean Search and Rescue in DARPA SubT," *Field Robotics*, vol. 3, pp. 1–68, 2023.

[4] D. Natter *et al.*, "An Incrementally Deployed Swarm of MAVs for Localization Using Ultra-Wideband," in *IEEE ICUAS*, 2022.

[5] I. Spasojevic *et al.*, "Active Collaborative Localization in Heterogeneous Robot Teams," in *RSS*, 2023.

[6] L. Parker *et al.*, "Tightly-coupled navigation assistance in heterogeneous multi-robot teams," in *IEEE IROS*, 2004.

[7] J. Huang *et al.*, "Localization and follow-the-leader control of a heterogeneous group of mobile robots," *IEEE/ASME Transactions on Mechatronics*, vol. 11, no. 2, pp. 205–215, 2006.

[8] A. Howard *et al.*, "Experiments with a Large Heterogeneous Mobile Robot Team: Exploration, Mapping, Deployment and Detection," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 431–447, 2006.

[9] M. Hofmeister *et al.*, "Cooperative visual mapping in a heterogeneous team of mobile robots," in *IEEE ICRA*, 2011.

[10] M. A. Hsieh *et al.*, "Adaptive teams of autonomous aerial and ground robots for situational awareness," *Journal of Field Robotics*, vol. 24, no. 11-12, pp. 991–1014, 2007.

[11] S. Zhang *et al.*, "Fast Active Aerial Exploration for Traversable Path Finding of Ground Robots in Unknown Environments," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–13, 2022.

[12] E. Mueggler *et al.*, "Aerial-guided navigation of a ground robot among movable obstacles," in *IEEE SSRR*, 2014.

[13] J. Delmerico *et al.*, "Active Autonomous Aerial Exploration for Ground Robot Path Planning," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 664–671, 2017.

[14] P. Fankhauser *et al.*, "Collaborative navigation for flying and walking robots," in *IEEE IROS*, 2016.

[15] A. Lakas *et al.*, "A Framework for a Cooperative UAV-UGV System for Path Discovery and Planning," in *IIT*, 2018.

[16] B. R. Chang *et al.*, "Drone-Aided Path Planning for Unmanned Ground Vehicle Rapid Traversing Obstacle Area," *Electronics*, vol. 11, no. 8, p. 1228, 2022.

[17] C. Chen *et al.*, "Motion Planning for Heterogeneous Unmanned Systems under Partial Observation from UAV," in *IEEE IROS*, 2020.

[18] C. Shen *et al.*, "Collaborative air-ground target searching in complex environments," in *IEEE SSRR*, 2017.

[19] J. Chen *et al.*, "A GAN-based Active Terrain Mapping for Collaborative Air-Ground Robotic System," in *IEEE ICARM*, 2019.

[20] T. Miki *et al.*, "UAV/UGV Autonomous Cooperation: UAV assists UGV to climb a cliff by attaching a tether," in *IEEE ICRA*, 2019.

[21] J. Lee *et al.*, "Aerial online mapping on-board system by real-time object detection for UGV path generation in unstructured outdoor environments," *Journal of Field Robotics*, vol. 40, no. 7, pp. 1754–1765, 2023.

[22] Z. Li *et al.*, "ColAG: A Collaborative Air-Ground Framework for Perception-Limited UGVs' Navigation," 2023, arXiv:2310.13324 [cs].

[23] H. Masnavi *et al.*, "VACNA: Visibility-Aware Cooperative Navigation With Application in Inventory Management," *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 7114–7121, 2023.

[24] Y. Jang *et al.*, "Navigation-Assistant Path Planning within a MAV team," in *IEEE IROS*, 2020.

[25] Q. Wang *et al.*, "Visibility-aware Trajectory Optimization with Application to Aerial Tracking," in *IEEE IROS*, 2021.

[26] Y. Lee *et al.*, "Target-visible Polynomial Trajectory Generation within an MAV Team," in *IEEE IROS*, 2021.

[27] L. Yin *et al.*, "Decentralized Swarm Trajectory Generation for LiDAR-based Aerial Tracking in Cluttered Environments," in *IEEE IROS*, 2023.

[28] V. Pritzl *et al.*, "Cooperative Navigation and Guidance of a Micro-Scale Aerial Vehicle by an Accompanying UAV using 3D LiDAR Relative Localization," in *IEEE ICUAS*, 2022.

[29] V. Pritzl *et al.*, "Fusion of Visual-Inertial Odometry with LiDAR Relative Localization for Cooperative Guidance of a Micro-Scale Aerial Vehicle," 2023, arXiv:2306.17544 [cs].

[30] T. Baca *et al.*, "The MRS UAV System: Pushing the Frontiers of Reproducible Research, Real-world Deployment, and Education with Autonomous Unmanned Aerial Vehicles," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 1, p. 26, 2021.

[31] A. Hornung *et al.*, "OctoMap: an efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[32] V. Krátký *et al.*, "An autonomous unmanned aerial vehicle system for fast exploration of large complex indoor environments," *Journal of Field Robotics*, vol. 38, no. 8, pp. 1036–1058, 2021.

[33] M. Vrba *et al.*, "On Onboard LiDAR-based Flying Object Detection," 2023, arXiv:2303.05404 [cs].

[34] D. Hert *et al.*, "MRS Modular UAV Hardware Platforms for Supporting Research in Real-World Outdoor and Indoor Environments," in *IEEE ICUAS*, 2022.

[35] ——, "MRS Drone: A Modular Platform for Real-World Deployment of Aerial Multi-Robot Systems," *Journal of Intelligent & Robotic Systems*, vol. 108, pp. 1–34, 2023.

[36] J. Zhang *et al.*, "LOAM: Lidar Odometry and Mapping in Real-time," in *RSS*, 2014.

[37] P. Geneva *et al.*, "OpenVINS: A Research Platform for Visual-Inertial Estimation," in *IEEE ICRA*, 2020.