

# ColAG: A Collaborative Air-Ground Framework for Perception-Limited UGVs' Navigation

Zhehan Li<sup>† 1,2</sup>, Rui Mao<sup>† 2,3</sup>, Nanhe Chen<sup>1,2</sup>, Chao Xu<sup>1,2</sup>, Fei Gao<sup>1,2</sup> and Yanjun Cao<sup>1,2</sup>

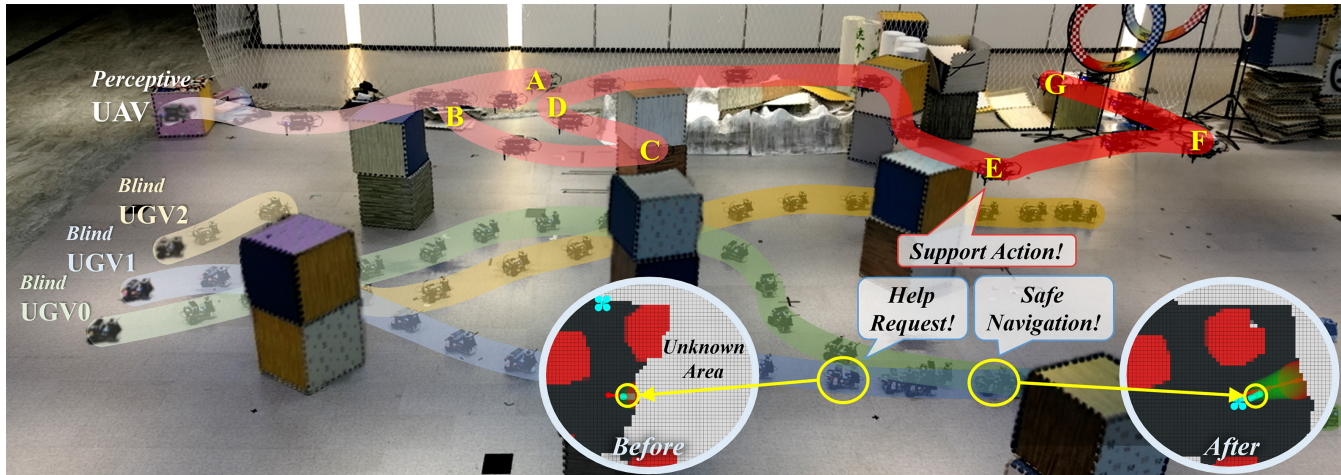


Fig. 1: Real-world experiment of ColAG with a group of perception-limited UGVs and one fully functional UAV. The ColAG system allows three blind UGVs to safely navigate through an area full of obstacles with help from the UAV. The UAV autonomously flies to points A-G to support the UGVs avoiding potential collisions by improving state estimation accuracy and providing maps. E for example, UGV1 requests help from the UAV because of its high state estimation uncertainty and potential to enter unknown terrains. After the arrival of the UAV, UGV1 can move forward safely.

**Abstract**— Perception is necessary for autonomous navigation in an unknown area crowded with obstacles. It's challenging for a robot to navigate safely without any sensors that can sense the environment, resulting in a *blind* robot, and becomes more difficult when comes to a group of robots. However, it could be costly to equip all robots with expensive perception or SLAM systems. In this paper, we propose a novel system named ColAG, to solve the problem of autonomous navigation for a group of *blind* UGVs by introducing cooperation with one UAV, which is the only robot that has full perception capabilities in the group. The UAV uses SLAM for its odometry and mapping while sharing this information with UGVs via limited relative pose estimation. The UGVs plan their trajectories in the received map and predict possible failures caused by the uncertainty of its wheel odometry and unknown risky areas. The UAV dynamically schedules waypoints to prevent UGVs from collisions, formulated as a Vehicle Routing Problem with Time Windows to optimize the UAV's trajectories and minimize time when UGVs have to wait to guarantee safety. We validate our system through extensive simulation with up to 7 UGVs and real-world experiments with 3 UGVs.

<sup>†</sup> Equal contribution.

This work was supported by National Nature Science Foundation of China under Grant No. 62103368. (Corresponding author: Yanjun Cao, Fei Gao.)

<sup>1</sup> State Key Laboratory of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, 310027, China.

<sup>2</sup> Huzhou Institute of Zhejiang University, Huzhou, 313000, China.

<sup>3</sup> Dalian University of Technology, Dalian, 116024, China.

E-mails: {zhehanli, nanhe\_chen, cxu, fgaoaa, yanjunhi}@zju.edu.cn, speedforce@mail.dlut.edu.cn

## I. INTRODUCTION

Nowadays, unmanned ground vehicles (UGVs) and unmanned aerial vehicles (UAVs) are widely used in many fields, such as surveillance [1], agriculture [2], search-and-rescue [3] [4], and transportation [5] [6], almost all relying on kinds of external sensors to sense the environment, such as LiDAR, camera, and radar etc. Perception ability is one of the key functions for autonomous navigation in an unknown area crowded with obstacles, which is also the condition for the following navigation modules such as mapping, planning, and high-level strategy. It's challenging for a perception-limited robot to navigate safely, especially if it lacks sensors to detect the environment, resulting in a *blind* robot discussed in this paper. Furthermore, it becomes more difficult when comes to a group of robots with potential inter-robot collision due to the lack of perception. When the size of the group increases, it can be costly to equip all robots with expensive perception or SLAM systems.

In this paper, we target a problem of autonomous navigation for a group of low-cost blind UGVs, referred to as *blind navigation*. Our motivation derives from the fact that UGVs possess distinct advantages such as low cost, high load capacity, and low energy consumption. UGVs are often used as mission execution units and work as a swarm due to these benefits, such as multi-robot collaborative transportation. More than expensive to equip all robots with sensors, the on-ground viewpoint also can not make the best

usage of these expensive sensors. Whereas, UAVs excel in terms of exceptional mobility in 3D space and therefore have a much bigger perception area. Combining their benefits to form a collaborative system comprising UGVs and UAVs can make an efficient system of demanding tasks at low cost. Therefore, we introduce an aerial robot equipped with full perception capabilities, working as a shared remote mobile eye, cooperatively guiding a group of blind UGVs from the aerial view to solve this blind navigation problem.

Following this concept, the main problem of this system is how to use a perceptive UAV to guide perception-limited UGVs safely and efficiently in an unknown environment. It is not trivial for one single UAV to guide multiple ground robots simultaneously. The challenge mainly comes from three aspects, limitation of relative pose estimation (RPE), UGV's planning in a received map, and the scheduling of the UAV for dynamically changed situations. In this system, the connections between UAV and UGVs are built from the relative state estimation system, such as AprilTag [7], specially designed LED board [8] or CREPES [9], etc. Note that we do not consider the RPE techniques from global localization systems (e.g. motion capture systems) because of the dependency on infrastructure. However, all these RPE techniques are direct observation and therefore limited by the field of view (FOV) and the occlusion of the environment. One UAV can't guarantee that all UGVs are constantly viewed in its FOV for stable RPE. The UGVs face the challenge of navigating in a shared map without direct environmental sensing. Enabling the UGVs to achieve optimal planning, in a received map including free, obstacle and unknown areas, is crucial. The treatment of unknown areas affects collision prediction as well as the planning quality dramatically. The control of blind UGVs with only the wheel odometer faces substantial challenges if no further action is taken but just trajectory tracking. The last point is about the planning of the UAV considering the dynamic nature of the UGVs group navigation. The UAV needs to find the best planning trajectory considering the UGVs' changing situations to take action efficiently without wasting energy.

To overcome these difficulties and solve the blind navigation problem, we propose a novel Collaborative Air-Ground Framework (ColAG), which employs a perceptive UAV to guide multiple blind UGVs simultaneously. The UAV is equipped with LiDAR and RPE device, while the UGVs are equipped with RPE device and wheel odometer only. The UAV uses SLAM to obtain its odometry and map and shares them with UGVs to recover global consistent maps. The UGVs receive the shared map and plan trajectory while using an EKF to estimate its odometry in the map. The EKF fuses RPE measurements from the UAV and its wheel odometers. The blind UGVs continuously predict possible collisions caused by the uncertainty of odometry estimation and unknown areas in the map. Then the information of possible collisions is sent to the UAV and the UAV plans its motion to provide the UGVs with updated measurements to save them from collisions. The contributions of this paper are summarized as follows:

- 1) We propose ColAG, a novel framework to solve the problem of *blind navigation* that a UAV guides a group of blind UGVs to achieve autonomous navigation in an unknown crowded environment.
- 2) We design a path planning strategy for the blind UGVs, which aggressively plans the trajectory and conservatively predicts the possible collision of the UGVs caused by the localization uncertainty, to improve the efficiency and guarantee the safety of the UGVs.
- 3) We formulate the scheduling of the UAV as dynamic vehicle routing problems with time windows (VRPTW) to provide the support order for UGVs' motion, which can minimize the total length of the UAV's trajectory and maximize the support before UGVs reach the predicted collision positions.
- 4) We release our implementation of ColAG for the reference and benefit of the community<sup>1</sup>.

## II. RELATED WORKS

Considerable research has studied heterogeneous robot systems, particularly in collaborative Air-Ground robotic systems. A major amount of the researches focus on combining two kinds of robots to collaboratively carry out tasks such as exploration [10] [11] and reconstruction [12] [13], leveraging the complementary strengths of each robot. These researches aim to address the limitations inherent in different types of robots for overall performance and capabilities.

Some studies involve utilizing UAVs to assist UGVs in their movement, leveraging the UAV's broader FOV to provide enhanced environmental information to the UGVs. Delmerico et al. [3] and Peterson et al. [14] assemble imagery into an ortho mosaic and then classify it using classical computer vision methods to generate an initial trajectory for UGV. Miller et al. [4] use semantic information between aerial and ground realize RPE and run in real-time with multi-UGV. This collaboration allows the UGVs to benefit from the UAV's aerial perspective and gather more comprehensive situational awareness.

The environment-aware sensors carried by the UGVs were removed in some studies, and the RPE between the UAV and the UGVs was used to obtain the localization of the UGVs. Cognetti et al. [5] use one UAV with a camera to provide localization for multiple UGVs and develop a cooperative control scheme to keep the UGVs inside the camera field of view. Guerin et al. [6] use a visual servo on UAV to keep a leader UGV in the center of the image plane for UGV's localization, and the navigation waypoints are given by a human operator. Mueggler et al. [15] localize the UGV with AprilTag which can be detected by the UAV's camera, to compensate for the drift of the UGV's wheel odometry. [5] [6] [15] [16] incorporate visual devices on UAVs to provide crucial localization information for the UGV, which need UAV's continuous observation. This is not only inefficient but also limits the UAV's mobility, yet without observation, the UGV's localization faces drifting and uncertainty.

<sup>1</sup><https://github.com/fast-fire/ColAG>

Localization uncertainty brings collision risks to the system, which is studied in planning and control. EKF is the most used uncertainty propagation method. Zhu et al. [17], Kamel et al. [18] and Patil et al. [19] propagate uncertainties along the planned trajectory using EKF. Different methods are used to check collision with uncertainty. [17] calculate approximated tight bound of collision chance, [18] use the max eigenvalue of uncertainty matrix as  $\sigma$ , then use  $3\sigma$  as collision distance, and [19] truncate the a priori distribution to get collision probability.

Efficient scheduling of UAVs is necessary to guide these blind UGV systems, which has been studied in autonomous exploration problems. Zhou et al. [20] and Meng et al. [21] find a global tour for a UAV to visit viewpoints efficiently by formulating it as a Traveling Salesman Problem (TSP) with a little revision. Zhou et al. [22] and Gao et al. [23] utilize a Constrained Vehicle Routing Problem (CVRP) to minimize the overall lengths of UAVs' path and balance the workloads for UAVs. Stump et al. [24] consider a persistent surveillance problem, and cast it as a VRPTW to finding sequences of visits to discrete sites in a periodic fashion.

In ColAG, for UGV we design a strictly safe strategy for path planning, which includes trajectory generation and collision prediction based on uncertainty propagation, and for UAV we design a scheduling method formulated as VRPTW to guide UAV to support the UGVs.

### III. THE COLAG FRAMEWORK

In ColAG as shown in Fig. 3, the relationship between UAV and blind UGVs is like helper and recipient. The perceptive UAV runs SLAM for its map and odometry, supports UGVs' collaborative pose estimation (III-A) with RPE and shares the map (III-B). The blind UGVs plan trajectories in the received map and predict possible collisions (III-C), then send the collision information to the UAV to request support. The UAV utilizes this information to dynamically update VRPTW to schedule its waypoints for the optimal support order of the UGVs (III-D).

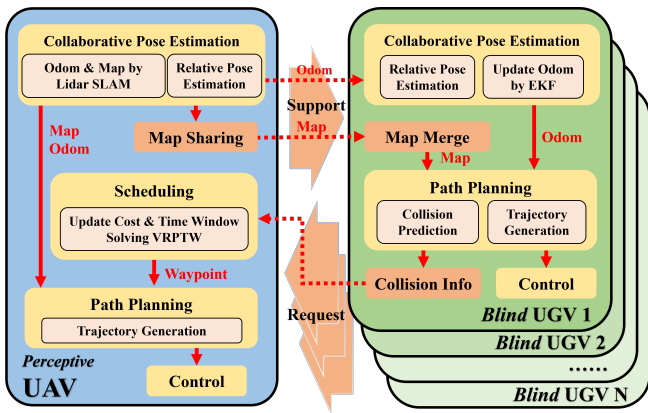


Fig. 3: The system architecture of ColAG.

#### A. Collaborative Pose Estimation

The UAV obtains its odometry and map by LiDAR SLAM directly, while the blind UGVs have to fuse their wheel

odometry and RPE with the UAV for accurate positioning. To obtain global consistent pose estimation, the UGVs employ EKF to fuse RPE, UAV's odometry, and wheel odometers. We write the state of the UGV at time  $k$  as  $\mathbf{x}_k$ , including 2D position and heading.

$$\mathbf{x}_k = [\mathbf{p}_k \ \theta_k]^T = [p_{x_k} \ p_{y_k} \ \theta_k]^T \quad (1)$$

1) *Prediction Model*: We use the wheel odometers of the UGV as the control input  $\mathbf{u}_k$  of the UGV, including linear velocity and angular velocity.

$$\mathbf{u}_k = [\mathbf{v}_k \ \omega_k]^T = [v_{x_k} \ v_{y_k} \ \omega_k]^T \quad (2)$$

The UGV state transition model can be written as

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) = \begin{bmatrix} \mathbf{p}_{k-1} + \mathbf{R}\{\theta_{k-1}\}\mathbf{v}_k\Delta t \\ \theta_{k-1} + \omega_k\Delta t \end{bmatrix}^T \quad (3)$$

$$\mathbf{R}\{\theta\} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Where  $\mathbf{p}$  is the position of the UGV in the world frame,  $\theta$  is the yaw angle of the UGV,  $\mathbf{v}$  is the velocity of the UGV,  $\omega$  is the angular velocity of the UGV,  $\Delta t$  is the time interval of the prediction step,  $\mathbf{R}\{\theta\}$  is the rotation matrix of the UGV.

The EKF prediction step is written as

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k) \\ \mathbf{P}_{k|k-1} &= \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{B}_k \mathbf{Q}_k \mathbf{B}_k^T \end{aligned} \quad (4)$$

2) *Measurement Model*: We use poses observed from the UAV as the measurement of the UGVs. This observed UGVs' poses can be obtained by multiplying the RPE and the odometry of the UAV. RPE can be obtained by the RPE device. The odometry of the UAV can be obtained by the SLAM algorithm running on the UAV. Formulated as

$$\begin{aligned} \mathbf{z}_k &= h(\mathbf{x}_k) = [\tilde{\mathbf{p}}_k \ \tilde{\theta}_k]^T \\ &= [x\{^W \mathbf{p}_{Gk}\} \ y\{^W \mathbf{p}_{Gk}\} \ yaw\{^W \mathbf{R}_{Gk}\}]^T \quad (5) \\ ^W \mathbf{R}_G &= ^W \mathbf{R}_A {}^A \mathbf{R}_G \\ ^W \mathbf{p}_G &= ^W \mathbf{p}_A + ^W \mathbf{R}_A {}^A \mathbf{p}_G \end{aligned} \quad (6)$$

Where  $^W \mathbf{R}_G$  and  $^W \mathbf{p}_G$  are the rotation matrix and position of the UGV in world frame,  $^W \mathbf{R}_A$  and  $^W \mathbf{p}_A$  are the rotation matrix and position of the UAV in world frame (obtained by SLAM),  $^A \mathbf{R}_G$  and  $^A \mathbf{p}_G$  are the rotation matrix and position of the UGV in UAV frame (obtained by RPE).

When a certain UGV is first observed by the UAV, the EKF can be initialized by the RPE. The EKF measurement step is written as

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_{k|k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \\ \hat{\mathbf{x}}_{k|k} &= \mathbf{x}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1})) \\ \mathbf{P}_{k|k} &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \end{aligned} \quad (7)$$

#### B. Map Sharing

We use the occupancy grid map as the map format. In UAV, the grid map is a 3D array, each cell in the array represents a grid in the environment, and the value of the cell represents the state of the grid. The state of the grid can

be free, occupied, or unknown, depending on the value of the cell. Note that these three states are treated differently on the UGVs for navigation, detailed in Sec. III-C. To reduce the bandwidth needed for map sharing, the UAV only sends the addresses of occupied cells and unknown cells in a specific region around the UGVs, and the free cells can be inferred as the remaining cells in the region. When the UGVs receive the map information, they store it in the same format as the UAV and merge the map according to the address of occupied cells and unknown cells.

### C. Path Planning of UGV

Due to the different viewpoints of the UAV and the UGVs, the environment surrounding the UGVs may be occluded in the UAV's map, presented as unknown cells in UGVs' map. These unknown cells in reality can be free or occupied for UGVs, causing the risk of collision between UGVs and obstacles, shown in Fig. 4.

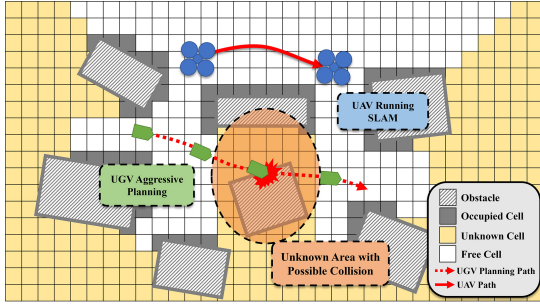


Fig. 4: The map constructed by the UAV is not suitable for UGVs' planning because of the occlusion of the environment. If the UGVs assume the unknown cells are free, possible collisions exist along the planning trajectory.

1) *Trajectory Generation in P-Map*: It is conservative to suggest that the UGVs can only move in the known-free environment when planning, which may reduce the efficiency of the system. We adopt an aggressive strategy that the UGVs plan in the planning map (P-Map) which assumes the unknown cells are free, as shown in Fig. 5(a). This approach aligns with conventional trajectory generation techniques utilized in autonomous navigation [25].

2) *Potential Risk of Collision*: This aggressive strategy brings collision risk to the UGVs, we consider two situations: First, when RPE observation from UAV is delayed than expected, the UGVs' odometers will shift, increasing the pose uncertainty. Second, obstacles might exist in unknown cells that the UGVs cannot perceive.

3) *Uncertainty Propagation of UGVs*: We use the covariance matrix of the EKF to represent the uncertainty of the pose estimation of UGVs. We calculate the prediction steps by Eq. 4 along the planning trajectory in a certain distance to propagate the UGV's uncertainty.

4) *Uncertainty-Based Collision Detect in CP-Map*: To deal with the two situations in III-C.2 uniformly, we propose an uncertainty-based collision prediction method in the collision predicting map (CP-Map) which suggests the unknown cells as occupied, as shown in Fig. 5(b).

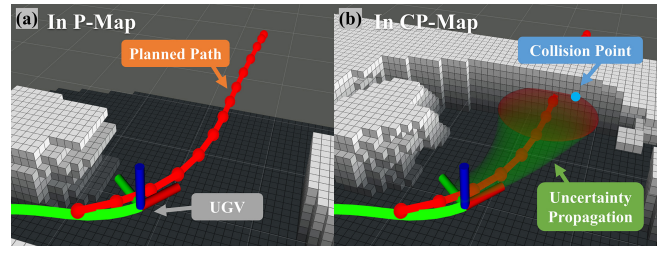


Fig. 5: The path planning of UGVs. (a) The UGV plans in P-Map, the red line is the planned path. (b) The UGV propagates uncertainty and predicts collision in CP-Map along the trajectory, the growing green area is the predicted covariance ellipses, the red ellipse is the collided ellipse, and the blue point is the collided point.

Since the covariance matrix of the EKF is an approximate Gaussian distribution, we can use it to find the possible collision of the UGVs in the CP-Map. Due to the  $3\sigma$ -rule in two-dimensional Gaussian distribution, that the probability of a random variable being within  $3\sigma$  range of its mean is 98.89%, we can assume that the UGVs will be safe if it keeps within  $3\sigma$  of its pose estimation along with the trajectory. The calculation is as follows:

- First, we calculate the eigen values and eigen vectors of the covariance matrix of the EKF

$$Q^{-1}P_{xy}Q = \Lambda = \text{diag}(\lambda_1, \lambda_2) \quad (8)$$

$$Q = [v_1 \ v_2]$$

Where  $P_{xy}$  is the covariance matrix of the EKF in XY plane,  $\Lambda$  is the eigen values of  $P_{xy}$ ,  $v_1$  and  $v_2$  are the eigen vectors of  $P_{xy}$ .

- Then we calculate the  $3\sigma$  ellipse

$$a = 3\sqrt{\lambda_1}, \quad b = 3\sqrt{\lambda_2}$$

$$1 \geq \frac{((p - p_G)v_1)^2}{a^2} + \frac{((p - p_G)v_2)^2}{b^2} \quad (9)$$

Where  $a$  and  $b$  are the length of the semi-major axis and semi-minor axis of the ellipse,  $p$  is the position of the cell in the ellipse,  $p_G$  is the position of the UGV.

- Finally, we traverse the cells occupied by the ellipses along the planned trajectory and find the center of collision ellipse  $p_{pc}$  and collision time  $t_{pc}$  of the UGVs. As LiDAR has some blind spots, we choose an expected support point  $p_{ps}$  with a certain distance back from the  $p_{pc}$  in the direction of planned velocity.  $p_{ps}$  and  $t_{pc}$  are combined as the **collision information**.

Note that UGV will stop and wait for UAV when collision time is less than a threshold  $t_c$ , which is 0.4 s in our work.

### D. Scheduling of UAV

Since we only have the UAV to sense the environment, we need to efficiently dispatch the UAV to help the UGVs maximize their efficiency. When the UAV comes to support, it can provide RPE measurement and nearby map information for the UGVs, so that the UGVs can update their pose estimation, obtain the information of potential collision cells, and replan the trajectory to avoid collision. To minimize the UGVs' waiting time because of possible collision, we need the UAV to support the UGVs before the UGVs reach the

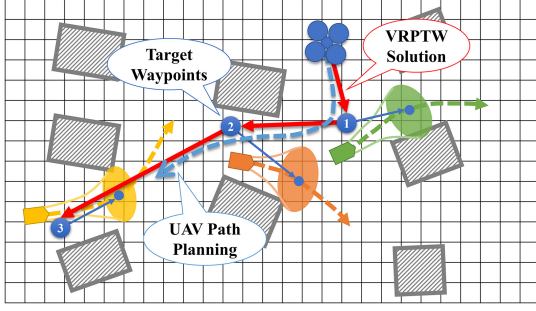


Fig. 6: The scheduling of UAV, the VRPTW takes the predicted collision information and gets the solution of the order to support the UGVs, which are used for UAV's trajectory planning.

potential collision cells, shown in Fig. 6.

We formulate the scheduling problem of UAV as a VRPTW by designing the time cost matrix  $M_{vrptw}$  and the time window matrix  $V_{vrptw}$ . Assume there are  $N$  collision points,  $M_{vrptw}$  corresponds to a  $N + 1$  dimensions square matrix and  $T_{vrptw}$  is a  $(N + 1) \times 2$  matrix, computed as:

$$\begin{aligned} M_{vrptw}(k_1, k_2) &= M_{vrptw}(k_2, k_1) \\ &= t_b(\mathbf{p}_{ps_{k_1}}, \mathbf{p}_{ps_{k_2}}), \quad k_1, k_2 \in \{1, 2, \dots, N\} \end{aligned} \quad (10)$$

$$M_{vrptw}(0, k) = t_b(\mathbf{p}_A, \mathbf{p}_{ps_k}), \quad k \in \{1, 2, \dots, N\}$$

$$M_{vrptw}(k, 0) = 0$$

$$T_{vrptw}(k, 0) = 0 \quad (11)$$

$$T_{vrptw}(k, 1) = t_{pc_k}, \quad k \in \{1, 2, \dots, N\}$$

Where  $\mathbf{p}_A$  is the position of the UAV.  $t_b(\mathbf{p}_{ps_{k_1}}, \mathbf{p}_{ps_{k_2}})$  is the time cost from  $\mathbf{p}_{ps_{k_1}}$  to  $\mathbf{p}_{ps_{k_2}}$ , calculated by

$$t_b(\mathbf{p}_{ps_{k_1}}, \mathbf{p}_{ps_{k_2}}) = \frac{\|\mathbf{p}_{ps_{k_1}} - \mathbf{p}_{ps_{k_2}}\|}{v_{max}} \quad (12)$$

We consider the continuity of the UAV's speed and calculate the time cost by

$$\begin{aligned} \mathbf{p}_{dis} &= \mathbf{p}_A - \mathbf{p}_{ps_k}, \quad v_{tan} = \mathbf{p}_{dis} / \|\mathbf{p}_{dis}\| \cdot \mathbf{v}_A \\ dis_a &= (v_{max}^2 - v_d^2) / (2a_{max}) \\ t_b(\mathbf{p}_A, \mathbf{p}_{ps_k}) &= \begin{cases} \frac{\sqrt{v_{tan}^2 + 2a_{max}dis_a} - v_{tan}}{a_{max}}, & dis_a > \|\mathbf{p}_{dis}\| \\ \frac{v_{max} - v_{tan}}{a_{max}} + \frac{\|\mathbf{p}_{dis}\| - dis_a}{v_{max}}, & dis_a \leq \|\mathbf{p}_{dis}\| \end{cases} \end{aligned} \quad (13)$$

We use OR-Tools [26] to solve the scheduling problem. The formulation of VRPTW is to ensure that the UAV can choose an optimal order to support the UGVs before the UGVs reach the possible collision position. However, there may be no solution to the VRPTW, which means the UAV cannot support all UGVs in time. In this case, we gradually increase the  $v_{max}$  until the VRPTW can find a solution.

#### IV. EXPERIMENT

To verify the effectiveness of ColAG, we conduct simulations with different configurations and real-world experiments. EGO-Swarm [25] is used to generate smooth trajectories of the UAV and UGVs.

#### A. Simulation Experiment

In the simulation experiment, we use the MARSIM [27] simulator to simulate the quadrotor UAV and LiDAR, and use numerical simulation to simulate the UGVs with differential chassis. We simulate the RPE by calculating true RPE and adding it with Gaussian noise.

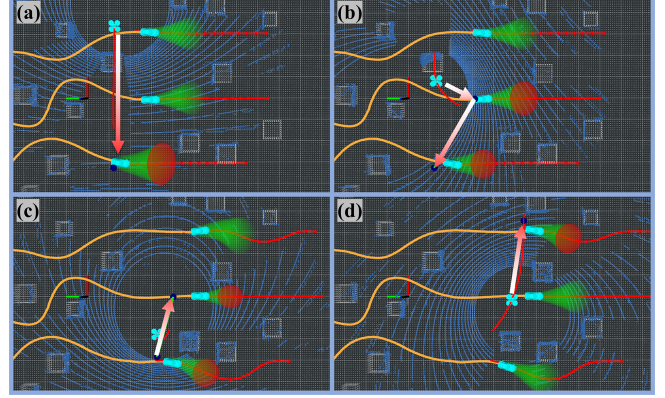


Fig. 7: The ColAG simulation. In (a)-(d), the UAV supports the UGVs which are predicted to collide in the order of the VRPTW solution, providing RPE and LiDAR perception of the environment.

The UGVs' dynamics limits are set to  $v_{max} = 0.5 \text{ m/s}$  and  $a_{max} = 1.0 \text{ m/s}^2$ , and the UAV's are set to  $v_{max} = 3.0 \text{ m/s}$  and  $a_{max} = 1.0 \text{ m/s}^2$ . The simulated Gaussian noise of UGV's wheel odometers is set to  $\sigma(v_x) = 0.0336 \text{ m/s}$ , and  $\sigma(\omega) = 0.0292 \text{ rad/s}$ , which is measured on our real platform, and noise of RPE is set to  $0.2 \text{ m}$  and  $0.05 \text{ rad}$ . The sensor range of the LiDAR and RPE is set to  $8 \text{ m}$  and  $5 \text{ m}$ , and the map-sharing range is set to  $2.5 \text{ m}$ .

We test the performance in sparse and dense environments with different numbers of UGVs. The environments are  $27 \times 27 \times 3 \text{ m}^3$  space with randomly generated obstacles, 40 in sparse environment and 80 in dense environment. We evaluate the performance by the following metrics:

- **UGV Reach Time (s):** The average time when UGV reaches the goal.
- **UGV Waiting Time (s):** The average time of UGV waiting because the predicted collision time is too small.
- **UGV Trajectory Length (m):** The average length of UGV's trajectory.
- **UAV Trajectory Length (m):** The average length of UAV's trajectory.

To prove our effectiveness, we also simulate UGVs with full perception, with no need for the UAV in the system.

- **Self-Perception UGV Reach Time (s):** The average time when UGV reaches the goal, where UGVs are equipped with LiDAR and run SLAM independently.
- **Self-Perception UGV Trajectory Length (m):** The average length of UGV's trajectory, where UGVs are equipped with LiDAR and run SLAM independently.

The simulation results are shown in Table I. According to the results, ColAG successfully guides the UGVs to cross sparse and dense environments, with different numbers of UGVs. When the number of UGVs is less than 5, the reach

TABLE I: ColAG Benchmark

Envi.	UGV Num.	UGV Reach Time	UGV Wait. Time	UGV Traj. Len.	UAV Traj. Len.	S.P. UGV Reach Time	S.P. UGV Traj. Len.
Spar.	1	106.78	0.00	33.02	36.35	106.20	32.92
	3	106.13	0.11	32.96	120.94	106.59	32.98
	5	116.62	3.87	33.65	214.07	107.59	33.32
	7	129.12	12.59	33.59	265.50	106.92	33.22
Dense	1	112.30	0.00	34.95	38.27	111.04	34.52
	3	112.38	0.07	34.73	133.08	110.69	34.53
	5	119.54	4.86	34.68	216.05	110.50	34.38
	7	143.18	24.65	34.98	304.34	111.88	34.94

time is close to the self-perception UGVs' reach time, and the UGVs' waiting time is small. When the number of UGVs is greater than 5, the waiting time of the UGVs and the length of the UAV's trajectory increase, which is acceptable as the limitation of the UAV.

To the best of our knowledge, there is no other work that solved the blind navigation problem. To verify the necessity of the parts in ColAG, we designed an ablation study with the following methods

- **without RPE Measurement:** The UGVs only use RPE to initialize the EKF and run on their wheel odometers.
- **without Uncertainty Propagation:** The UGVs predict collision without considering the uncertainty.
- **without Scheduling:** The UAV will support the UGV with the smallest predicted collision time first.
- **without Time Window:** The UAV doesn't consider the time window of the possible collision, and the VRPTW formation degenerates to VRP.

TABLE II: ColAG Ablation

Methods	UGV Reach Time	UGV Avg. Wait. Time	UGV Max. Wait. Time	UGV Traj. Len.	UAV Traj. Len.
Proposed	119.54	4.86	11.08	34.68	216.05
w/o RPE Meas.	-	-	-	-	-
w/o Unce. Prop.	-	-	-	-	-
w/o Sche.	135.73	18.10	27.85	34.85	239.71
w/o Time Wind.	132.43	18.29	46.65	34.81	238.56

We conducted the ablation study in dense environment with 5 UGVs and the results are shown in Table II. Without RPE measurement and uncertainty propagation, the UGVs pose estimation drift and collide with the obstacles, resulting in mission failure. Without scheduling considering flight time cost, the UAV can't plan an optimal global trajectory, leading to a longer trajectory length, and then can't support UGVs in time. Without time window, the UAV always follows one UGV, leading to a much longer waiting time of other UGVs.

### B. Real-World Experiment

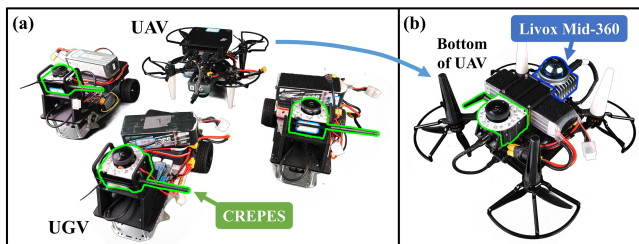


Fig. 8: The UAV and UGVs platform, the green edge device is CREPES and the blue edge device is Livox Mid-360. (a) The UAV and 3 UGVs. (b) The bottom of the UAV.

To validate the proposed method, we conduct real-world experiments. Fig. 8 shows a quadrotor and three simple

differential-driven UGVs we used. All robots are equipped with CREPES [9] as RPE and Intel NUC (i7-1165G7 CPU) as the computation platform. The UAV is equipped with a Livox Mid-360 and runs Fast-lio [28] as its SLAM module.

The UGVs' dynamics limits are set to  $v_{max} = 1.0 \text{ m/s}$  and  $a_{max} = 1.0 \text{ m/s}^2$ , and the UAV's are set to  $v_{max} = 1.5 \text{ m/s}$  and  $a_{max} = 0.5 \text{ m/s}^2$ . The range of the RPE is set to  $3\text{m}$ . The other parameters are the same as the simulation.

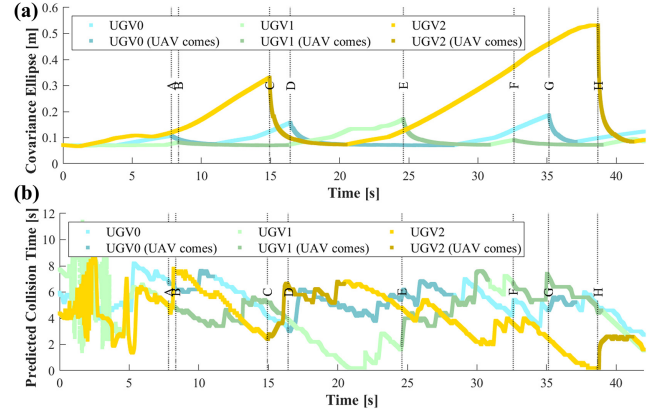


Fig. 9: Real world experiment. A-H are the points UAV came to help in Fig. 1. (a) The propagation of the UGVs' state estimation uncertainty (half-long axis  $\sqrt{\lambda_1}$  of ellipse). (b) The UGVs' predicted collision time.

The real-world experiment is shown in Fig. 1. The UAV and UGVs are placed in a  $20 \times 10 \times 3 \text{ m}^3$  arena and the goal is set to  $11\text{m}$  away from the start point. In Fig. 9, as long as the UAV comes to support, the uncertainty of the UGVs' pose estimation is decreased, and the predicted collision time is increased, guaranteeing the safety of the UGVs. The result in Table III shows that ColAG performed well with the direct RPE device CREPES, the UGVs reached the goal without collision and the UAV supported the UGVs in time.

TABLE III: Real World Experiment

UGV ID	UGV Reach Time	UGV Wait. Time	UGV Traj. Len.	UAV Traj. Len.
0	41.15	0.00	12.92	25.75
1	38.60	1.20	11.07	
2	42.75	0.40	13.52	
Avg.	40.83	0.53	12.50	

## V. CONCLUSION

In this work, we propose a novel cooperative air-ground framework to solve the *blind navigation* problem in which only one perceptive UAV guides a group of blind UGVs to navigate in unknown environments. A path planning strategy is introduced to consider the uncertainty of UGVs' pose estimation and the unknown parts of the map, guaranteeing the UGVs safety. Based on collision prediction, a customized VRPTW scheduling strategy is proposed for the UAV to support the UGVs, optimizing the UAV's trajectory and the UGVs' waiting time. Simulation and real-world experiments are conducted to verify the effectiveness of the proposed method. In the future, we will consider the initiative exploration of the UAV on the possible collision trajectory, and the cooperation of multiple UAVs to support the UGVs.

## REFERENCES

- [1] Z. Lin and H. H. Liu, "Topology-based distributed optimization for multi-uav cooperative wildfire monitoring," *Optimal Control Applications and Methods*, vol. 39, no. 4, pp. 1530–1548, 2018.
- [2] A. Pretto, S. Aravecchia, W. Burgard, N. Chebrolu, C. Dornhege, T. Falck, F. Fleckenstein, A. Fontenla, M. Imperoli, R. Khanna *et al.*, "Building an aerial-ground robotics system for precision farming: an adaptable solution," *IEEE Robotics & Automation Magazine*, vol. 28, no. 3, pp. 29–49, 2020.
- [3] J. Delmerico, E. Mueggler, J. Nitsch, and D. Scaramuzza, "Active autonomous aerial exploration for ground robot path planning," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 664–671, 2017.
- [4] I. D. Miller, F. Cladera, T. Smith, C. J. Taylor, and V. Kumar, "Stronger together: Air-ground robotic collaboration using semantics," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9643–9650, 2022.
- [5] M. Cagnetti, G. Oriolo, P. Peliti, L. Rosa, and P. Stegagno, "Cooperative control of a heterogeneous multi-robot system based on relative localization," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 350–356.
- [6] F. Guérin, F. Guinand, J.-F. Brethé, H. Pelvillain *et al.*, "Uav-ugv cooperation for objects transportation in an industrial area," in *2015 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2015, pp. 547–552.
- [7] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 3400–3407.
- [8] X. Yan, H. Deng, and Q. Quan, "Active infrared coded target design and pose estimation for multiple objects," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6885–6890.
- [9] Z. Xun, J. Huang, Z. Li, C. Xu, F. Gao, and Y. Cao, "Crepes: Cooperative relative pose estimation towards real-world multi-robot systems," *arXiv preprint arXiv:2302.01036*, 2023.
- [10] L. Wang, D. Cheng, F. Gao, F. Cai, J. Guo, M. Lin, and S. Shen, "A collaborative aerial-ground robotic system for fast exploration," in *Proceedings of the 2018 International Symposium on Experimental Robotics*. Springer, 2020, pp. 59–71.
- [11] J. Butzke, A. Dornbush, and M. Likhachev, "3-d exploration with an air-ground robotic system," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3241–3248.
- [12] H. Qin, Z. Meng, W. Meng, X. Chen, H. Sun, F. Lin, and M. H. Ang, "Autonomous exploration and mapping system using heterogeneous uavs and ugvs in gps-denied environments," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1339–1350, 2019.
- [13] J. He, Y. Zhou, L. Huang, Y. Kong, and H. Cheng, "Ground and aerial collaborative mapping in urban environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 95–102, 2020.
- [14] J. Peterson, H. Chaudhry, K. Abdelatty, J. Bird, and K. Kochersberger, "Online aerial terrain mapping for ground robot navigation," *Sensors*, vol. 18, no. 2, p. 630, 2018.
- [15] E. Mueggler, M. Faessler, F. Fontana, and D. Scaramuzza, "Aerial-guided navigation of a ground robot among movable obstacles," in *2014 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2014, pp. 1–8.
- [16] C. Shen, Y. Zhang, Z. Li, F. Gao, and S. Shen, "Collaborative air-ground target searching in complex environments," in *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, 2017, pp. 230–237.
- [17] H. Zhu and J. Alonso-Mora, "Chance-constrained collision avoidance for mavs in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 776–783, 2019.
- [18] M. Kamel, J. Alonso-Mora, R. Siegwart, and J. Nieto, "Robust collision avoidance for multiple micro aerial vehicles using nonlinear model predictive control," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 236–243.
- [19] S. Patil, J. Van Den Berg, and R. Alterovitz, "Estimating probability of collision for safe motion planning under gaussian motion and sensing uncertainty," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 3238–3244.
- [20] B. Zhou, Y. Zhang, X. Chen, and S. Shen, "Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 779–786, 2021.
- [21] Z. Meng, H. Qin, Z. Chen, X. Chen, H. Sun, F. Lin, and M. H. Ang, "A two-stage optimized next-view planning framework for 3-d unknown environment exploration, and structural reconstruction," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1680–1687, 2017.
- [22] B. Zhou, H. Xu, and S. Shen, "Racer: Rapid collaborative exploration with a decentralized multi-uav system," *IEEE Transactions on Robotics*, 2023.
- [23] Y. Gao, Y. Wang, X. Zhong, T. Yang, M. Wang, Z. Xu, Y. Wang, Y. Lin, C. Xu, and F. Gao, "Meeting-merging-mission: A multi-robot coordinate framework for large-scale communication-limited exploration," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 13 700–13 707.
- [24] E. Stump and N. Michael, "Multi-robot persistent surveillance planning as a vehicle routing problem," in *2011 IEEE International Conference on Automation Science and Engineering*. IEEE, 2011, pp. 569–575.
- [25] X. Zhou, J. Zhu, H. Zhou, C. Xu, and F. Gao, "Ego-swarm: A fully autonomous and decentralized quadrotor swarm system in cluttered environments," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 4101–4107.
- [26] L. Perron and V. Furnon, "Or-tools," Google. [Online]. Available: <https://developers.google.com/optimization/>
- [27] F. Kong, X. Liu, B. Tang, J. Lin, Y. Ren, Y. Cai, F. Zhu, N. Chen, and F. Zhang, "Marsim: A light-weight point-realistic simulator for lidar-based uavs," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2954–2961, 2023.
- [28] W. Xu and F. Zhang, "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317–3324, 2021.