

Obstacle Avoidance of Resilient UAV Swarm Formation with Active Sensing System in the Dense Environment

Peng Peng, Wei Dong*, Gang Chen and Xiangyang Zhu

Abstract—This paper proposes a perception-shared and swarm trajectory global optimal (STGO) algorithm fused UAVs formation motion planning framework aided by an active sensing system. First, the point cloud received by each UAV is fit by the gaussian mixture model (GMM) and transmitted in the swarm. Resampling from the received GMM contributes to a global map, which is used as the foundation for consensus. Second, to improve flight safety, an active sensing system is designed to plan the observation angle of each UAV considering the unknown field, overlap of the field of view (FOV), velocity direction and smoothness of yaw rotation, and this planning problem is solved by the distributed particle swarm optimization (DPSO) algorithm. Last, for the formation motion planning, to ensure obstacle avoidance, the formation structure is allowed for affine transformation and is treated as the soft constraint on the control points of the B-spline. Besides, the STGO is introduced to avoid local minima. The combination of GMM communication and STGO guarantees a safe and strict consensus between UAVs. Tests on different formations in the simulation show that our algorithm can contribute to a strict consensus and has a success rate of at least 80% for obstacle avoidance in a dense environment. Besides, the active sensing system can increase the success rate of obstacle avoidance from 50% to 100% in some scenarios.

I. INTRODUCTION

Compared to one single unmanned aerial vehicle (UAV), a swarm of UAVs can effectively increase payload capacity and have a broader sense of the environment. With such merits, swarms could be more efficiently fulfill tasks such as detection, search, mapping, and cooperative transportation [1]. In the aforementioned tasks, the swarm is commonly required to maintain a specified formation to traverse a dense field of obstacles safely. This requirement brings two challenges. First, for a distributed swarm, one main prerequisite is that all UAVs need to reach a safe consensus on trajectory planning and decision-making [2] while usually, different UAVs have a different understanding of the environment. Current work [3] uses the convex hull to express safe space for transmission, which builds a foundation for perception consensus. Nevertheless, perception consensus can not strictly ensure planning consensus due to the local minima existing in the optimization problem. Second, for the sake of the dense environment and the overall large volume of UAV formation, it is significant to have a complete perception of the environment to ensure safety [4]. Therefore, the FOV of each UAV in the swarm needs to be elaborately planned. Although there has been a lot of work [5], [6]

All authors are with the State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiaotong University, Shanghai, 200240, China (email: {yc.pengpeng, dr.dongwei, chg947089399, mexyzhu}@sjtu.edu.cn).

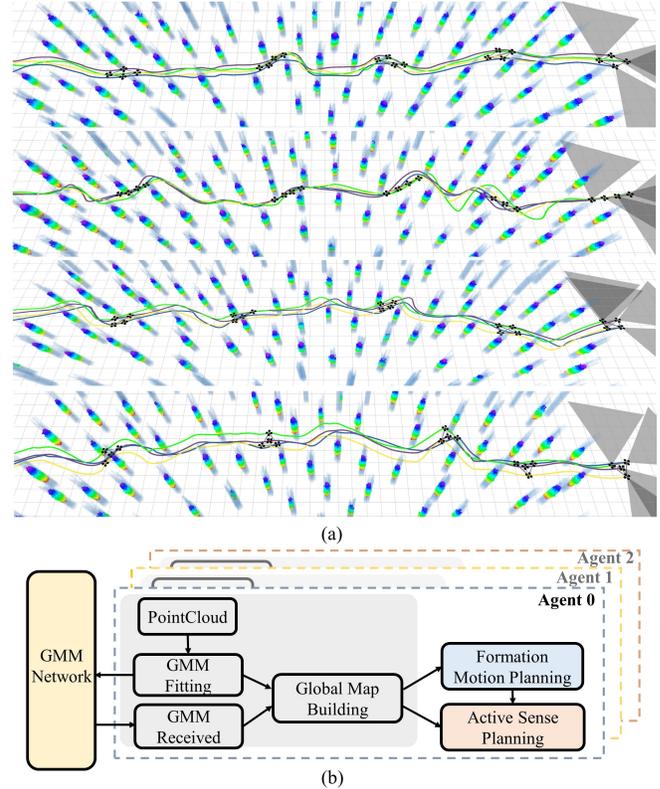


Fig. 1. (a) Diamond, straight line, rectangle and triangle formation traverse through the forest scenario from left to right. (b) An overview of our perception and planning system, including formation motion planning and active sense planning.

considering the field of view of one single UAV, planning FOVs for a UAV swarm in motion planning is still an open problem.

To bridge the gap of research, we propose a distributed and synchronous motion planning framework for multiple UAVs formation equipped with an active sensing system in the dense obstacle environment, as shown in Fig. 1b. The GMM is used to represent the distribution of obstacles in the environment compactly and transmitted in the swarm. This transmission contributes to a global map for each agent and lays a foundation for the consensus of subsequent planning. The STGO algorithm in the formation motion planning module, together with the GMM communication, guarantees strict consensus for all UAVs.

In addition, the FOVs of all UAVs are optimized considering the exploration and safety, as well as reducing the overlap of FOVs. This optimization is solved by the DPSO

algorithm.

Besides, in order to improve flight safety, our work allows the affine transformation of the formation structure and treats it as the soft constraint on the control points of the B-spline. The formation trajectory planning is based on the gradient-based method [7]. We adopt the synchronous planning manner and treat the trajectories and observation directions of all UAVs as optimization variables to ensure a feasible solution for all UAVs. Simulation is conducted to verify our algorithm for formation maintenance and obstacle avoidance.

We list the main contributions of our work as follows:

- 1) A perception-shared and STGO algorithm fused UAVs formation motion planning framework, ensuring a safe and strict consensus on trajectory planning.
- 2) A distributed active sensing system for the UAV swarm, and to the best of our knowledge, this is the first system considering FOVs in the UAV swarm motion planning task.

II. RELATED WORKS

A. Distributed Formation Obstacle Avoidance

Numbers of methods for UAV formation obstacle avoidance have been proposed. Among them, the lead-follower pattern is widely used due to its convenience of implementation [8]. This method highly depends on the leader agent, which makes the formation less robust. Paul et al. in [9] construct an artificial potential field to maintain the formation shape and avoid the obstacle. However, the potential field usually has many local minima, resulting in the robot being trapped.

Recently, motion planning for distributed formation obstacle avoidance has been extensively researched. In [10], Quan et al. use a graph-based metric to quantify the similarity between two formations. For each agent, the trajectory is optimized based on other agents' pre-planned trajectories. In some scenarios, the pre-planned trajectories may lead to the following UAVs failing in path finding. In other words, the optimized trajectories may not be optimal for the whole swarm. Alonso et al. [3] present a method for formation flight planning based on distributed consensus. The core of this method is to get the same safe convex hull for different robots by intersection, which is taken as the feasible area for planning. This same safe convex hull builds the prerequisite for consensus. Nevertheless, at the same time, this may also make obstacles and trajectories close because it treats the safe space equally [11]. Besides, the same convex hull can not strictly guarantee optimization consensus.

B. Active Sensing System

A large sensing range could improve flight safety and localization accuracy. Compared to the omnidirectional camera sensor [12], [13], active sensing system active vision has advantages in weight and computation consumption. In [4], Zhou et al. propose a yaw planning algorithm and define the quality of observation to improve flight safety. In [5], to help the object tracking, the heading of the UAV is planned to keep the dynamic obstacles in the FOV. Chen et al. [6]

use a multiple objectives function considering exploration, velocity, and dynamic obstacles to plan for an independent rotation camera with 1 degree of freedom. Besides, the active sensing system has also been applied to the UAV swarm. Researchers from [14] design the optimal trajectory and angular configuration for a UAV swarm to track a target of interest. Zhang et al. [15] use a graph-based active sensing method to improve relative localization for UAVs in the swarm.

III. GMM-BASED MAPPING AND TRANSMISSION

In this work, GMMs are used to compactly represent point cloud and transmitted in the swarm. By resampling from the models, we can reconstruct the obstacles, and build the occupancy grid map and the euclidean signed distance functions (ESDF) map. Compared to standard gaussian distribution, GMM can fit more complex point distributions [16] due to the linear combination of multiple gaussian models, which are called components. The probability density function of a GMM with K components is formed as

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (1)$$

where $\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the standard gaussian distribution with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$. π_k represents the proportion of each component in the GMMs with $\sum_{k=1}^K \pi_k = 1$. Our work uses the Expectation-Maximum (EM) algorithm [17] to calculate the optimal π_k , $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ of the GMM. EM algorithm solves the maximum-likelihood estimation in a iterative manner. For the E-step, the probability of x_n belonging to component z_k is calculated and the probability is called as the latent variable.

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \quad (2)$$

Then for the M-step, the optimal parameters are optimized through maximum-likelihood based on the latent variable

$$\begin{aligned} \boldsymbol{\mu}_k^{\text{new}} &= \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n \\ \boldsymbol{\Sigma}_k^{\text{new}} &= \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}})(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}})^T \\ \pi_k^{\text{new}} &= \frac{N_k}{N} \end{aligned} \quad (3)$$

where $N_k = \sum_{n=1}^N \gamma(z_{nk})$. By iteratively calculating Eq. 2 and Eq. 3, all parameters can converge to the optimal value.

One key point of the GMM is that the number of components K needs to be specified in advance. When faced with an unknown environment, it is difficult for a robot to select a suitable K . Commonly, with more components, the GMM can represent the environment more accurately, while this also costs the EM algorithm much more time. Many techniques have been raised to choose an appropriate K according to the complexity of the environment [16]. In our work, K is proportional to the number of points in the point

cloud. This criterion can adapt to most environments and does not require additional computations.

Reconstructing the distribution of obstacles in space is implemented by resampling from the GMM. Meanwhile, an occupancy grid map is built through raycasting from the robot to the sampled points. The state update of each grid in the map follows the sensor measurement model in [18]. Moreover, a Euclidean Signed Distance Field (ESDF) map is maintained by Euclidean distance transform (EDT), and both maps are stored in the circular buffer data structure [19]. Illustrations for the GMM-based maps are shown in Fig. 2.

Thanks to the highly compact expression of point cloud with GMM, the message for communication only contains the parameters of GMM and the location of the agent. This expression greatly reduces the demand for communication bandwidth from 200 kbps to 5 kbps in simulation.

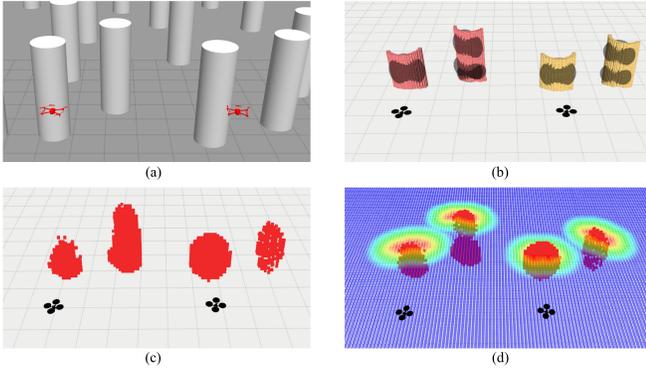


Fig. 2. The pipeline for two UAVs building a global map. (a) Two UAVs are equipped with depth cameras in the Gazebo simulation environment. The camera's FOV is limited. (b) The red grids are the point cloud seen by the UAV on the left, and the yellow grids belong to the UAV on the right. The black ellipses are plotted according to the GMM fitting result. Each ellipse represents a component. (c) The global map is constructed by the left UAV after GMM communication and resampling. Through communication, it can reconstruct the obstacles on the right side. (d) A cross-section of the ESDF map built from the global map maintained by the left UAV.

IV. ACTIVE-SENSING SYSTEM PLANNING

When conducting a traversal mission, the UAV swarm is required to have a full understanding of the environment and hence, we adopt the active sensing system for UAVs by planning the yaw angle. For a swarm of N UAVs, denote the angles of all UAV's yaw attitude in the swarm as $\Theta = [\theta_0, \theta_1, \dots, \theta_{N-1}]^T$, where θ_i represents the yaw angle of UAV i . The positions of UAVs for yaw angles optimization are selected from the pre-planned trajectory described in Sec.V with equal time interval, denoted as $P(t_0), P(t_1), \dots, P(t_{m-1})$. Yaw angles between this selected time point are calculated by linear interpolation. Considering the balance between safety and exploration, the cost function for yaw angles optimization at t_k is defined as

$$\min_{\Theta_k} J = \lambda_1 f_{ol} + \lambda_2 f_v - \lambda_3 f_{pv} + \lambda_4 f_s \quad (4)$$

$$\theta_i \in [-\pi, \pi), \quad i \in [0, N-1]$$

The f_{ol} function in the first term is defined as $f_{ol} = \sum_{i \neq j} Overlap(\theta_i, \theta_j)$, where $Overlap(\theta_i, \theta_j)$ function calculates the overlap area of two FOVs between UAV i and UAV j as shown in Fig. 3. In practice, to reduce computational complexity, we project the UAV's FOV onto the same height plane and use the Boost Geometry¹ library to get the size of the overlapping regions of the two triangles. Besides, to ensure the safety of the swarm, agents need to pay attention to obstacles in its direction of velocity. Thus, we set $f_v = \sum_i (\theta_i - \theta_i^v)^2$, where θ_i^v is the velocity direction of UAV i determined by the trajectory. The third term $f_{pv} = \sum_i PerceptionValue(\theta_i)$ in this cost function evaluates the perception value corresponding to a certain yaw angle. To be specific, in order to have a complete understanding of the whole environment and improve the success rate of path finding, agents tend to observe the field of higher uncertainty. Borrowing from the theory of information entropy, the perception value is defined as $PerceptionValue(\theta_i) = -\sum_{grid_l \in FOV} p_l \log p_l$ where p_l indicates the occupancy probability of grid l within the FOV. This function highlights those grids with occupancy probability around 0.5. The last term in the cost function considers the smoothness of the UAV's yaw rotation and is defined as $f_s = \|\Theta_k - \Theta_{k-1}\|_2^2$. Moreover, $\lambda_1, \lambda_2, \lambda_3$ and λ_4 determine the weight of each term. The optimization

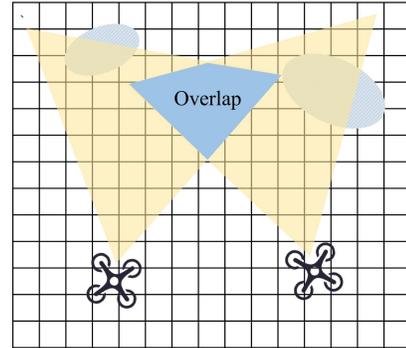


Fig. 3. The overlap area of two cameras' FOVs on the same height plane.

problem described above is a typical non-convex problem. In order to take advantage of the computing power of the UAV swarm and avoid local minima, we utilize a DPSO algorithm as shown in Alg. 1. Compared with standard PSO, DPSO adds the interaction between UAVs and thus increases the number of particles used for computation. Besides, DPSO can guarantee strict consensus between UAVs. Specifically, in the beginning, UAV i initializes a group of particles, and then each particle updates its state concerning the optimal particle in the particle group P_{gbest}^i and its own history optimal value $P_{pbest}^{i,k}$. All UAVs in the swarm broadcast the current optimal particle of their own particle group at a certain frequency. After receiving the information P_{gbest}^j , the UAV will compare it with its own group's optimal particle and select the better one. This algorithm achieves

¹<https://github.com/boostorg/geometry>

optimization consensus in practical and compared with some distributed optimization algorithms that are highly dependent on communication like DGT [20], the DPSO algorithm can obtain better optimization results in the case of unstable bandwidth.

Algorithm 1: DPSO algorithm

Input: max iteration times M

- 1 For agent i , initialize a group of particles S_i with random velocities and positions;
- 2 for $episode=1$ to M do
 - 3 for particle $P_k \in S_i$ do
 - 4 if $J(P_k) < J(P_{pbest}^{i,k})$ then
 - 5 $P_{pbest}^{i,k} \leftarrow P_k$;
 - 6 if $J(P_k) < J(P_{gbest}^i)$ then
 - 7 $P_{gbest}^i \leftarrow P_k$;
 - 8 for particle $P_k \in S_i$ do
 - 9 updateVelocityAndPosition(P_k);
 - 10 if receive P_{gbest}^j from other agent j then
 - 11 if $J(P_{gbest}^j) < J(P_{gbest}^i)$ then
 - 12 $P_{gbest}^i \leftarrow P_{gbest}^j$;
 - 13 broadcast P_{gbest}^i at a certain frequency;

V. FORMATION MOTION PLANNING

The swarm in our work is required to maintain a formation during the traversal task. To avoid obstacle, the formation allows for rotation, scaling and translation. We use a simplified affine transformation to describe formation at time t

$$P(t) = sRP_{ref} + P_{trans}(t) \quad (5)$$

where s represents the scale factor, R is the rotation matrix and $P_{trans}(t)$ describes the translation of the formation centre. P_{ref} is the reference formation structure such as diamond and straight line.

Motion planning for the formation can be divided into two steps. For the first step, the A* algorithm is used to find a path from the formation center to the goal, denoted as $Path_c$. Due to the limited sensing range of UAV, when the node in the A* algorithm reaches the map boundary, the algorithm is terminated, and the boundary point is regarded as the endpoint of the path. In our work, in order to ensure the connectivity of the swarm during obstacle avoidance, all UAVs are required to bypass the obstacles from the same side. Such requirement is reasonable for some situations, such as vision-based relative localization or cooperative transportation. Therefore, the path for each UAV is based on $Path_c$ as shown in Fig. 4. For each turning point along $Path_c$, a new formation is generated around it with a small scale factor for safety, and then each agent's path is found by connecting these new formations. In practice, these paths can be roughly used as the front-end paths for formation motion planning.

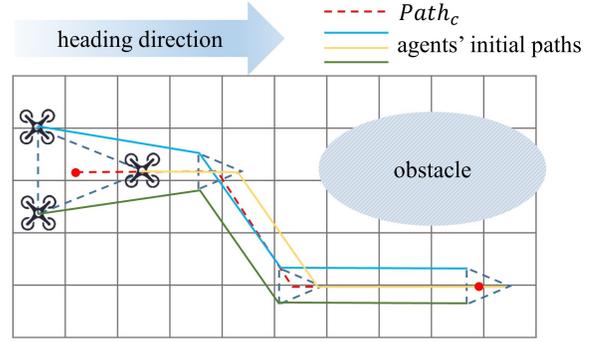


Fig. 4. The initial path of each agent. $Path_c$ starts from the formation centre and each agent's initial path is generated based on $Path_c$.

The second step of motion planning is based on gradient-gated optimization. According to the closed-form solution presented by [21], we calculate a polynomial trajectory crossing all turning points of the path found by the first step following the principle of minimum acceleration. Then by sampling on the trajectory, each UAV's trajectory is parameterized as B-spline. Each UAV optimizes the trajectories of all UAVs simultaneously. For a formation with N agents and B-spline curves with n control points, p order, the problem is formulated as

$$\begin{aligned} & \arg \min \\ & q_j^i, i \in [0, N-1], j \in [3, n-1] \\ & J = \lambda_f J_f + \lambda_b J_b + \lambda_s J_s + \lambda_d J_d + \lambda_o J_o + \lambda_r J_r + \lambda_e J_e \end{aligned} \quad (6)$$

where q_j^i represents the j th control point of agent i 's trajectory. $\lambda_f, \lambda_b, \lambda_s, \lambda_d, \lambda_o, \lambda_r$ and λ_e determine the weight of each term. Because we specify the initial position, velocity, and acceleration of the trajectory, the q_0^i, q_1^i , and q_2^i are fixed.

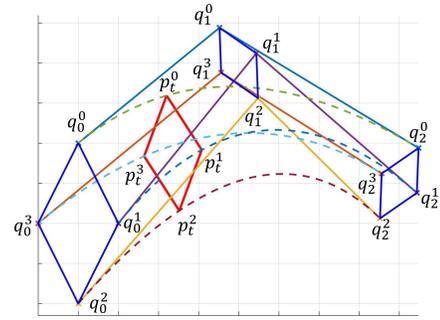


Fig. 5. Illustration for Theorem 1. Suppose the reference formation is the diamond. q_j^i is j th the control point of UAV i 's B-spline. The blue diamonds formed by q_j^i all satisfy the affine transformation relative to the reference formation. According to Theorem 1, at any time t , the red diamond formed by UAV i 's position p_t^i satisfies the affine transformation relative to the reference formation.

To ensure safe flight first, we treat the formation requirement as a soft constraint, and thus, the first term in the cost function measures the degree of formation distortion. The formation constraint is converted to the control points by the following theorem for B-spline curves. Illustration of

Theorem 1 is shown in Fig. 5.

Theorem 1: Suppose the formation formed by control points of all B-splines satisfies the affine transformation compared to the reference formation, then at any time on the trajectory, the formation formed by the agents satisfies the affine transformation compared to the reference formation.

Proof: At any time on the trajectory, the formation is determined by control points, $\mathbf{P}(t) = \mathbf{N}(t) \cdot \mathbf{Q}$, where $\mathbf{P}(t) = [\mathbf{P}_0(t), \mathbf{P}_1(t) \dots \mathbf{P}_{N-1}(t)]$ with $\mathbf{P}_i(t)$ denotes the position of agent i . $\mathbf{N}(t)$ is composed of the base function value which is only related to time t . \mathbf{Q} is a matrix of all the control points, $\mathbf{Q} = [\mathbf{q}_0^T, \mathbf{q}_1^T \dots \mathbf{q}_p^T]^T$ and $\mathbf{q}_j = [\mathbf{q}_j^0, \mathbf{q}_j^1 \dots \mathbf{q}_j^{N-1}]$. \mathbf{q}_j satisfies a general affine transformation compared to the reference formation, $\mathbf{q}_j = \mathbf{A}_j \cdot \mathbf{P}_{ref} + \mathbf{B}_j$. Then, $\mathbf{Q} = [\mathbf{A}_0, \mathbf{A}_1 \dots \mathbf{A}_p]^T \cdot \mathbf{P}_{ref} + [\mathbf{B}_0, \mathbf{B}_1 \dots \mathbf{B}_p]^T$ and thus, $\mathbf{P}(t) = \mathbf{N}(t) \cdot [\mathbf{A}_0, \mathbf{A}_1 \dots \mathbf{A}_p]^T \cdot \mathbf{P}_{ref} + \mathbf{N}(t) \cdot [\mathbf{B}_0, \mathbf{B}_1 \dots \mathbf{B}_p]^T$. ■

Thanks to Theorem 1, we can convert the formation constraint on trajectory points to the control points. According to [22], the in-plane affine transformation in Eq. 5 is determined by three agents, while in our work, the affine transformation is simplified to rotation, scaling and translation, and thus, it can be controlled by two agents, denoted as agent 0 and agent 1. In other word, other agents' desired position is a linear combination of agent 0 and agent 1 defined as $\hat{\mathbf{q}}_j^i = F(\mathbf{q}_j^0, \mathbf{q}_j^1)$, where $\hat{\mathbf{q}}_j^i$ is the desired position of the j th control point of agent i satisfying the formation structure. Therefore, the formation cost function is defined as

$$J_f = \sum_{i=2}^{N-1} \sum_{j=3}^{n-1} \|\mathbf{q}_j^i - \hat{\mathbf{q}}_j^i\|_2^2 \quad (7)$$

In order to prevent a large degree of scaling factor, we introduce J_b to limit the distance between agent 0 and agent 1,

$$J_b = \sum_{j=3}^{n-1} (\|\mathbf{q}_j^0 - \mathbf{q}_j^1\|_2^2 - d_{ref}^2)^2 \quad (8)$$

where d_{ref} is the distance between agent 0 and agent 1 in the reference formation.

The J_s in Eq. 6 represents the cost function of smoothness. The elastic band cost function [7] is adopted to formulate it.

$$J_s = \sum_{i=0}^{N-1} \sum_{j=3}^{n-2} \|\mathbf{q}_j^i - \mathbf{q}_{j-1}^i - \mathbf{q}_{j+1}^i\|_2^2 \quad (9)$$

Due to the dynamic feasibility, we limit the speed and acceleration at the control points. The cost function to penalize excessive speed and acceleration is defined as $J_d = \sum_{i=0}^{N-1} (\sum_{j=2}^{n-2} g_v(\mathbf{v}_j^i) + \sum_{j=1}^{n-2} g_a(\mathbf{a}_j^i))$ with $g_v(\mathbf{v}_j^i)$ described below and $g_a(\mathbf{a}_j^i)$ follows the same form.

$$g_v(\mathbf{v}_j^i) = \begin{cases} (\|\mathbf{v}_j^i\|_2^2 - v_{max}^2)^2 & \|\mathbf{v}_j^i\|_2 > v_{max} \\ 0 & \|\mathbf{v}_j^i\|_2 \leq v_{max} \end{cases} \quad (10)$$

where \mathbf{v}_j^i is the control point of one order derivative of the B-spline trajectory.

J_o is used to keep trajectories away from obstacles. It can be quantified by the value in ESDF map. Due to the convex hull property of B-spline, we set the obstacle avoidance constraint on control points, $J_o = \sum_{i=0}^{N-1} \sum_{j=3}^{n-1} h(\mathbf{q}_j^i)$ with $h(\mathbf{q}_j^i)$ defined as

$$h(\mathbf{q}_j^i) = \begin{cases} (E(\mathbf{q}_j^i) - D)^2 & E(\mathbf{q}_j^i) < D \\ 0 & E(\mathbf{q}_j^i) \geq D \end{cases} \quad (11)$$

where $E(\mathbf{q}_j^i)$ represents the ESDF value and D is the safety distance for obstacle avoidance. The gradient of $E(\mathbf{q}_j^i)$ with respect to \mathbf{q}_j^i is calculated by trilinear interpolation.

J_r is the reciprocal collision cost for formation flight. Theoretically, it is difficult to guarantee that two UAVs will not collide at any time when only the control points are constrained, and thus, we select a series of points on the trajectory at equal time intervals $[t_0, t_1 \dots t_K]$ to apply constraints.

$$J_r = \sum_{i,j=0, i < j}^{N-1} \sum_{k=0}^K r(\mathbf{p}^i(t_k), \mathbf{p}^j(t_k))$$

$$r(\mathbf{p}^i, \mathbf{p}^j) = \begin{cases} (\|\mathbf{p}^i - \mathbf{p}^j\|_2^2 - D_r^2)^2 & \|\mathbf{p}^i - \mathbf{p}^j\|_2 < D_r \\ 0 & \|\mathbf{p}^i - \mathbf{p}^j\|_2 \geq D_r \end{cases} \quad (12)$$

where $\mathbf{p}^i(t_k)$ is the position of agent i at time t_k and D_r is the safety distance for reciprocal avoidance. Besides, $\mathbf{p}^i(t_k)$ is a linear combination of control points, so the gradient with J_r to \mathbf{q}^i can be easily calculated.

The endpoint of the path found in the first step $\hat{\mathbf{p}}_{end}^i$ is treated as a soft constraint. Due to the property of the Non-Uniform B-spline, the last control point coincides with the endpoint of the trajectory. Therefore, the cost function for endpoint constraint is defined as

$$J_e = \sum_{i=0}^{N-1} \|\mathbf{q}_{n-1}^i - \hat{\mathbf{p}}_{end}^i\|_2^2 \quad (13)$$

Each agent in the swarm uses the L-BFGS algorithm from NLOpt² to optimize the control points of all trajectories. Although all UAVs have a shared map due to the GMM communication in Sev.IV, they can not strictly reach a consensus on trajectories because of the local minima in Eq. 6. Therefore, we introduce the STGO algorithm. After the L-BFGS algorithm converges, every UAV will broadcast its optimization result and the corresponding cost. The result with the lowest cost will be chosen as the final trajectories for all UAVs. In practice, the STGO algorithm effectively avoids local minima and ensures a strict consensus.

VI. SIMULATION

We test our active sensing system and formation avoidance algorithm in simulation, and the environment is based on ego-planner-swarm³. The UAVs will perform GMM fitting on the point cloud information provided by the simulation environment and transmit it in the swarm. Similar to the

²<https://github.com/stevengj/nlopt>

³<https://github.com/ZJU-FAST-Lab/ego-planner-swarm>

real-world scenario, the camera’s FOV is set to 60 degrees, and the maximum distance of the point cloud is 5m. Unlike other works [10], [23], our reference formation tends to bring UAVs together for tasks such as cooperative transportation and relative localization. Typical formations include diamond, straight line, rectangle, and triangle.

A. Active Sensing System

The DPSO algorithm in our system takes 40 particles on each agent, and the maximum number of iterations is set as 20. The DPSO algorithm takes around 35ms, most of which is used to calculate perception value.

We compare our active sensing system with the conventional velocity-direction-consistency yaw planning method. As shown in Fig. 6, for some scenarios like sharp turn, the velocity-direction-consistency yaw planning method can not detect obstacles in time, and this leads to the crash. We conduct 20 tests, and the success rate for the velocity-direction-consistency yaw planning method to cross the corner is 50%, while with the help of our active sensing system, the success rate is 100%. Besides, for formations like the straight line, the velocity-direction-consistency yaw planning method causes a large overlap of FOVs as shown in Fig. 7a. Our algorithm can take advantage of the UAV swarm by planning the direction of observation, maintaining full awareness of the environment, as demonstrated in Fig. 7b. To be specific, the overlap rate of FOV in Fig. 7a is 48.0% while in Fig. 7b, it is 14.7%.

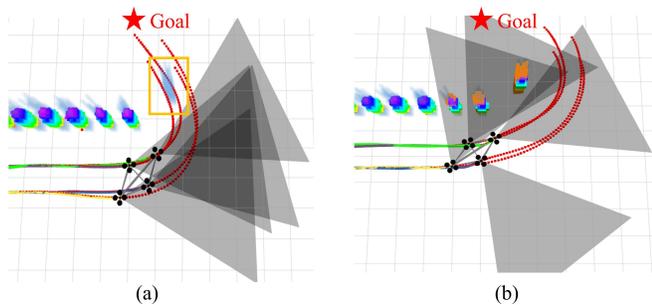


Fig. 6. Comparison between two perception planning algorithms when facing sharp turn scenario. Red curves are the planned trajectories, and black transparent triangles are the UAVs’ FOVs. Unknown obstacles are in blue. (a) The velocity-direction-consistency yaw planning algorithm cannot detect obstacles marked by the orange rectangle and lead to unsafe trajectories. The success rate to cross the corner is 50%. (b) Our algorithm enables the UAV to actively observe unknown areas to detect obstacles in time and improve flight safety. The success rate to cross the corner is 100%

TABLE I
SUCCESS RATE AND e_{dist} WITH AND WITHOUT GMM OR STGO

Algorithm	GMM+STGO	GMM	STGO
Success rate	85%	10%	0%
$e_{dist}(m)$	0.3984	1.6459	0.0725

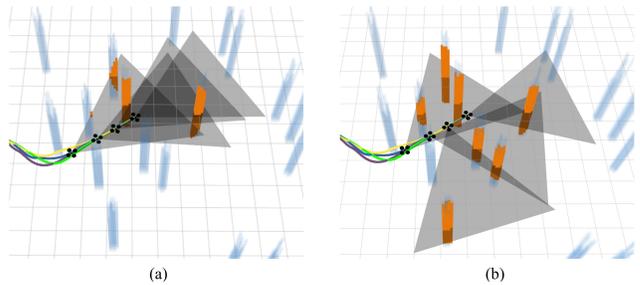


Fig. 7. The sense range of two perception planning algorithms. (a) is the velocity-direction-consistency yaw planning algorithm and the overlap rate of FOV is 48.0%. (b) is our proposed algorithm and the overlap rate of FOV is 14.7%.

TABLE II
SUCCESS RATE AND e_{dist} OF DIFFERENT FORMATION STRUCTURES IN THE FOREST SCENARIO

Formation structure	Diamond	Line	Rectangle	Triangle
Success rate	85%	90%	80%	80%
e_{dist}	0.3984	0.0234	1.0032	0.3920

B. Formation Motion Planning

During the traversal task, all UAVs rely on synchronous timestamps for synchronous planning, which in our system comes from ROS. When the planning timer is triggered, each UAV broadcasts its current position, and then the algorithm described in Sec.V is conducted. We solve the following optimization from [10] to describe the distortion degree of the current formation relative to the reference formation,

$$e_{dist} = \min_{R,s,t} \sum_{i=0}^{N-1} \left\| \mathbf{p}_i^{ref} - (s\mathbf{R}\mathbf{p}_i^c + \mathbf{t}) \right\|^2 \quad (14)$$

where \mathbf{p}_i^{ref} and \mathbf{p}_i^c is the position of UAV i in the reference and current formation respectively. A smaller e_{dist} represents a smaller distortion.

We demonstrate the success rate and e_{dist} of our system when traversing a forest scenario maintaining a diamond formation with and without GMM communication or STGO in Table I. As can be seen from the table, without STGO, the distortion degree is large, and there is much reciprocal collision, indicating a bad consensus between UAVs. Without GMM, UAVs can not maintain a full understanding of the environment and thus, evaluate the trajectories cost incorrectly, leading to a 0% success rate. Only under the joint help of GMM and STGO ensures a safe and strict consensus between UAVs.

In addition, we use different reference formations to demonstrate our obstacle avoidance performance in an unknown environment. The average speed is about 1.8m/s, and the motion planning algorithm costs around 30ms. Fig. 1a illustrates the diamond, straight line, rectangle, and triangle formation in the forest scenario. As the figure shows, all UAVs avoid the obstacles on the same side, and the connectivity of the swarm is maintained during the flight. When encountering narrow corridors, the formation will shrink and

temporarily break for safety. The success rate and e_{dist} of different formation structures in the forest scenario for 20 tests are shown in Table II.

VII. CONCLUSIONS

This paper proposes a novel formation perception and motion planning framework. The simulation shows that the GMM and STGO fused method can guarantee a safe and strict consensus in a traversal task with a success rate of more than 80%. At the same time, GMM can substantially reduce the communication bandwidth from 200kbps to 5kbps. Besides, an active sensing system is embedded in this framework to improve the success rate from 50% to 100% in the sharp turn scenario. In the future, we plan to test this framework in the real-world experiment and develop it for a cooperative transportation task.

REFERENCES

- [1] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 837–855, 2018.
- [2] T. Nestmeyer, P. Robuffo Giordano, H. H. Bühlhoff, and A. Franchi, "Decentralized simultaneous multi-target exploration using a connected network of multiple robots," *Autonomous robots*, vol. 41, no. 4, pp. 989–1011, 2017.
- [3] J. Alonso-Mora, E. Montijano, T. Nägele, O. Hilliges, M. Schwager, and D. Rus, "Distributed multi-robot formation control in dynamic environments," *Autonomous Robots*, vol. 43, no. 5, pp. 1079–1100, 2019.
- [4] B. Zhou, J. Pan, F. Gao, and S. Shen, "Raptor: Robust and perception-aware trajectory replanning for quadrotor fast flight," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1992–2009, 2021.
- [5] J. Tordesillas and J. P. How, "Panther: perception-aware trajectory planner in dynamic environments," *arXiv preprint arXiv:2103.06372*, 2021.
- [6] G. Chen, W. Dong, X. Sheng, X. Zhu, and H. Ding, "An active sense and avoid system for flying robots in dynamic environments," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 2, pp. 668–678, 2021.
- [7] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3529–3536, 2019.
- [8] D. Panagou and V. Kumar, "Cooperative visibility maintenance for leader–follower formations in obstacle environments," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 831–844, 2014.
- [9] T. Paul, T. R. Krogstad, and J. T. Gravdahl, "Modelling of uav formation flight using 3d potential field," *Simulation Modelling Practice and Theory*, vol. 16, no. 9, pp. 1453–1462, 2008.
- [10] L. Quan, L. Yin, C. Xu, and F. Gao, "Distributed swarm trajectory optimization for formation flight in dense environments," *arXiv preprint arXiv:2109.07682*, 2021.
- [11] L. Quan, L. Han, B. Zhou, S. Shen, and F. Gao, "Survey of uav motion planning," *IET Cyber-systems and Robotics*, vol. 2, no. 1, pp. 14–21, 2020.
- [12] H. Xu, Y. Zhang, B. Zhou, L. Wang, X. Yao, G. Meng, and S. Shen, "Omni-swarm: A decentralized omnidirectional visual-inertial-uwB state estimation system for aerial swarm," *arXiv preprint arXiv:2103.04131*, 2021.
- [13] F. Schilling, F. Schiano, and D. Floreano, "Vision-based drone flocking in outdoor environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2954–2961, 2021.
- [14] R. Tallamraju, E. Price, R. Ludwig, K. Karlapalem, H. H. Bühlhoff, M. J. Black, and A. Ahmad, "Active perception based formation control for multiple aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4491–4498, 2019.
- [15] P. Zhang, G. Chen, and W. Dong, "Agile formation control of drone flocking enhanced with active vision-based relative localization," *arXiv preprint arXiv:2108.05505*, 2021.
- [16] M. Corah, C. O'Meadhra, K. Goel, and N. Michael, "Communication-efficient planning and mapping for multi-robot exploration in large environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1715–1721, 2019.
- [17] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.
- [18] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics.*, ser. Intelligent robotics and autonomous agents. MIT Press, 2005.
- [19] V. Usenko, L. Von Stumberg, A. Pangercic, and D. Cremers, "Real-time trajectory replanning for mavs using uniform b-splines and a 3d circular buffer," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 215–222.
- [20] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 2055–2060.
- [21] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics research*. Springer, 2016, pp. 649–666.
- [22] S. Zhao, "Affine formation maneuver control of multiagent systems," *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4140–4155, 2018.
- [23] 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.