VG-Swarm: A Vision-Based Gene Regulation Network for UAVs Swarm Behavior Emergence

Huanlin Li, Yuwei Cai, Juncao Hong, Peng Xu, Hui Cheng, *Member, IEEE*, Xiaomin Zhu, Bingliang Hu, Zhifeng Hao, and Zhun Fan, *Senior Member, IEEE*

***Abstract*—We present VG-Swarm, a practical and effective method for aerial robots dynamic encirclement, which consists of a vision-based gene regulatory network (V-GRN) and a visual perception module. For each flying robot deployed with the proposed method, the relative spatial positions of the surrounding robots, targets, and obstacles are first obtained by omnidirectional monocular vision. Then the proposed method is used to generate the concentration field within its own perception range according to the obtained position information. The agent individually calculates and selects an optimal moving direction in its concentration field, and finally stays on its selected encirclement pattern (a closed concentration contour around the target). As a result, a swarm of flying robots can emerge adaptive pattern formations to entrap the targets even without any communication and global information. We verify the effectiveness and robustness of the proposed method in various simulations and real-world experiments.**

***Index Terms*—Aerial systems: Perception and autonomy, biologically-inspired robots, swarm robotics.**

I. Introduction

RECENTLY, the research of swarm unmanned aerial vehicles (UAVs) has attracted increasing attention from the research community due to its wide applicability. Representative applications include collaborative navigation [1], target tracking [2], flocking [3], autonomous search and rescue [4], and target encirclement [5], [6] etc. In particular, target encirclement includes entrapping dynamic targets [5], or protecting friend targets by using multiple UAVs [7].

Manuscript received 4 July 2022; accepted 22 December 2022. Date of publication 12 January 2023; date of current version 23 January 2023. This letter was recommended for publication by Associate Editor I. Sa and Editor P. Pounds upon evaluation of the reviewers' comments. This work was supported in part by the National Key R&D Program of China under Grant 2021ZD0111502 and in part by the National Natural Science Foundation of China under Grant 62176147. (*Huanlin Li and Yuwei Cai are co-first authors.*) (*Corresponding author: Zhun Fan.*)

Huanlin Li, Yuwei Cai, Juncao Hong, Peng Xu, Zhifeng Hao, and Zhun Fan are with the Key Lab of Digital Signal and Image Processing of Guangdong Province, College of Engineering, University of Shantou, Guangdong 515063, China (e-mail: 2558022243@qq.com; caiyuwei86@163.com; 644811588@qq.com; 21pxu@stu.edu.cn; haozhifeng@stu.edu.cn; zfan@stu.edu.cn).

Hui Cheng is with the School of Computer Science and Engineering, University of Sun Yat-Sen, Guangdong 510006, China (e-mail: chengh9@mail.sysu.edu.cn).

Xiaomin Zhu is with the College of Systems Engineering National University of Defense Technology Changsha, Hunan 410073, China (e-mail: xmzhu@nudt.edu.cn).

Bingliang Hu is with the Xi'an Institute of Optics and Precision Mechanics, Shanxi 710119, China (e-mail: hbl@opt.ac.cn).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2023.3236565, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3236565

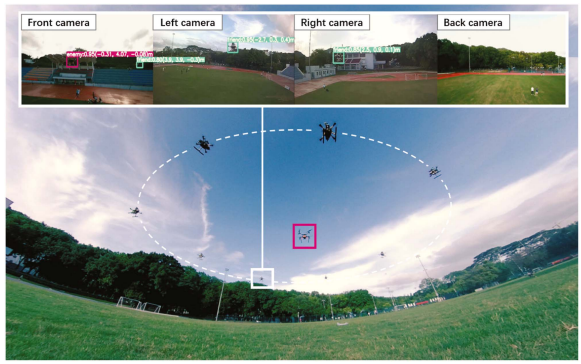

Fig. 1. The photo of 8 aerial robots encircling a target in a real-world experiment. Each captor drone detects and estimates the relative positions of the target (marked by a red square) and neighbors through omnidirectional vision, taking the captor (marked by a white square) at the bottom of the photo as an example.

The main purpose of target encirclement is to control a group of agents to emerge entrapping patterns automatically to accomplish entrapping or protecting tasks. In this letter, the encirclement formation is also referred to as pattern formation. Pattern formation initially involve biological morphogenesis, which occurs during biological development within multicellular organisms. Biological morphogenesis can be viewed as a self-organizing process. Populations of cells move autonomously to their destinations, governed by gene regulatory network (GRN) and cell-to-cell interactions. Among them, GRN is a model of genes and interactions of gene products that describes the gene expression dynamics [8]. The interaction between cells is reflected in the concentration gradient of morphogenetic substances, which can promote the autonomous migration of cells. The basic idea of applying the mechanism of GRN in biological morphogenesis to swarm robots is to establish a metaphor between cells and robots. In this analogy, each cell can be regarded as a robot, which can release proteins to the surroundings according to a certain rule. As a result, each point in the space has a corresponding protein concentration value, and the concentration field has consisted of all points in this concentration space area. Each robot establishes a concentration field around itself and performs motion control based on the gradient properties of concentration field.

The mainstream algorithms for encirclement formation can be largely divided into two categories: 1) behavior-based methods. Such as the classical leader-follower method [9] and the animal collective behavior-based method [10]. In which the robotics

2377-3766 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Tsinghua University. Downloaded on April 09,2023 at 05:33:18 UTC from IEEE Xplore. Restrictions apply.

are supposed to follow the leader or obey the simple interaction rules to achieve pattern formation. However, under this mechanism, the formed pattern cannot adapt well to the dynamic environment, especially in a scene with random obstacles. 2) field-based methods. In which the concentration filed-based method [11], artificial potentials field-based method [12], and vector field-based method [5] are commonly used ones. They can generate adaptively patterns in a dynamic environment. However, the vector field-based method requires a lot of human expert knowledge, and different rules need to be manually designed to complete the pattern generation. The concentration field established by the GRN is similar to the gravitational field established by the artificial potential field method in engineering implementation, but in fact, the source of ideas is different. The GRN model is inspired by biology and can be described by graphics, which is very human-friendly, and so it may be possible to use graphical description methods to find simpler but more effective structures.

In recent years, researchers have successfully applied GRN in designing coordination mechanisms for swarm robots [13]. Guo et al. [14] proposed a GRN model to enable multiple robots to autonomously self-organize into different predefined shapes. However, this method needs to use global coordinates system, which may be extravagant in many real-world applications. To address this problem, Guo et al. [15] extended the method by utilizing a reference robot to set a local coordinate system, through which each robot participates in forming part of a pattern of an uneven B-spline shape. Further, Jin et al. [11] proposed a hierarchical gene regulatory network (H-GRN) for adaptive pattern generation to entrap targets in dynamic environments. To deal with the problem of merging and separating patterns in the process of multi-target entrapment and the issue of obstacle avoidance, Oh et al. [16] proposed an EH-GRN structure that uses obstacles and targets as the inputs of an evolving GRN to generate a fused concentration field. And the above works rely on at least limited communication in their experiments.

We mainly consider a swarm of UAVs entrapping targets in communication-denied scenarios. In this case, each agent needs to perform its contribution task independently based on its own visual perception. In fact, vision is the main sensory mode of animal groups that allows collective behaviors [17]. Renaud Bastien et al. [18] demonstrated that organized collective behavior of multi-agent systems can be generated by a vision-based collective behavior model. Fabian Schilling et al. [19] proposed a target tracking algorithm to achieve vision-based UAVs flocking by using omnidirectional vision. Inspired by the above works, we propose a vision-based GRN model (V-GRN) for UAVs swarm to entrap dynamic targets. The main contributions of this letter are summarized as follows:

- We propose a V-GRN model for vision-based UAVs, which can generate adaptive candidate entrapping patterns in communication-denied environments.
- We improve the speed performance of the original YOLOv5s [20] network and propose a monocular position estimation algorithm to enable UAVs to estimate the position of detected objects without communication.

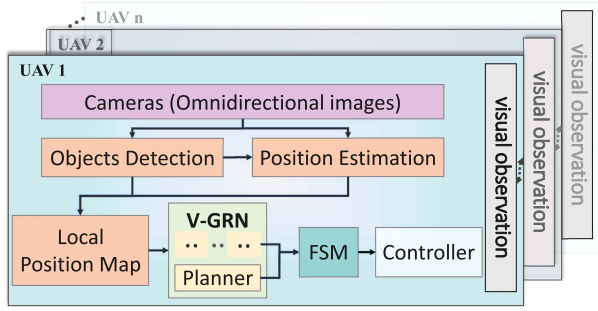

Fig. 2. *The system architecture of the proposed VG-Swarm.* The input, omnidirectional images captured from four cameras, are used for visual perception. In the module of local position map, it records the position information of the detected objects. V-GRN generates the concentration fields within the perception range using the environmental information provided by the local position map, from which the candidate entrapping patterns are formed. The planner in the bottom layer of V-GRN generates the velocity commands for the agent's movement control. The FSM module controls the behavior states switching and action output of the agent.

- A behavioral model of each UAV is designed and described in an FSM, enable swarm UAVs to accomplish search and entrapment in an integrated way.

II. Method

Fig. 2 illustrates the overall framework of the proposed method, which consists of three major parts as follows: visual perception (object detection and object position estimation), V-GRN, and design of the behavioral model of the UAVs.

A. Object Detection

Recently YOLO series have been widely used in detection tasks for their considerable accuracy and fast inference speed. However, directly applying YOLO models to tackle object detection tasks in drone-captured scenarios may not work well due to the usually small sizes of objects detected by drones and the limitation of onboard computing power. In this paper, to further improve the detection performance of tiny objects using onboard devices, we made some modifications to the original YOLOv5s network architecture [20].

In particular, we found that some prediction heads in the original YOLOv5 network are generated from high-level, low-resolution feature maps, which are less sensitive to tiny objects. Therefore, in the feature pyramids generation part, we added one more bottom-up and top-down path and replaced a prediction head for processing the lower-level, higher-resolution feature map. By doing so, the mAP performance of the model for tiny objects is improved about 0.7% (tested in our datasets), but with the parameters increasing 9.5%. Moreover, We observed that despite the use of a larger feature map, the feature diversity of tiny objects is still poor as image details become blurred, and many layers channels contribute little to detecting tiny objects. To reduce the parameters, we reduced the number of convolution kernels to around 40% of the YOLOv5s network.

Authorized licensed use limited to: Tsinghua University. Downloaded on April 09,2023 at 05:33:18 UTC from IEEE Xplore. Restrictions apply.

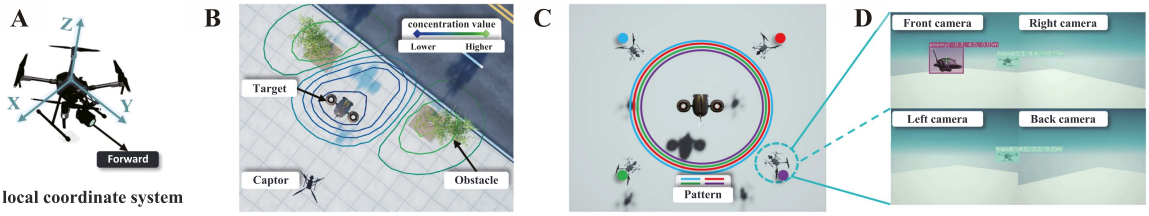

Fig. 3. *The schematic diagram illustrating the definition of a coordinate system and pattern formation in our work.* Each UAV establishes a right-handed local coordinate system (part A) with the forward direction as the Y axis. Part B presents the multiple concentration contour lines in a concentration field established by a captor in an obstacle scenario, in which the multiple contour lines around the target are candidate entrapping patterns, and the dark blue contour lines represents the lower concentration values and light green represents higher concentration values. In part C, 4 separate entrapping patterns generated by 4 UAVs individually in an open scenario. Ideally the four entrapping patterns would overlap each other. Part D shows the visual perception of the UAV in the lower right corner of part C.

However, even with the lightweight model, it is still unable to achieve omnidirectional visual detection in real-time on embedded edge computing devices (about 5.7 FPS). To address this issue, we take the following steps:

- Using the TensorRT engine to optimize the inference speed of our custom model and preparing data for the detector using multi-processing (17.56 FPS).
- Stitching images from four cameras into one (1280×960), so the detector only needs to detect the combined frame once to achieved omnidirectional visual detection (45.46 FPS).

B. Object Position Estimation

In this letter, we obtain the relative spatial position of the detected objects by extending the aforementioned online object detection, which combines the detection results and the camera field of view (FOV) information. Haseeb et al. [21] found that the inverse of the detection bounding box (Bbox) size increases linearly with the object distance, and further, we found that the object distance and the area of the Bbox exhibit a power function relationship, which can be well-fitted by a power function form as in (1). Where D is the object distance, s represents the pixel area of the Bbox, α, and n can be obtained by using the least squares method from multiple sets of s and D data.

$$D = \alpha \cdot s^n \tag{1}$$

Griffin et al. [22] proposed to use the camera motion and Bbox height information to deduce D, however, we found the Bbox height changes little for tiny instances although the camera moved noticeably. Similarly, we introduced that the distance D can be obtained by using the detection Bbox area s and relative camera motion, as shown in (2).

$$D_i = \frac{C_{z_j} - C_{z_i}}{1 - \sqrt{\frac{s_i}{s_j}}} \tag{2}$$

Where D_i is the object distance in time index i, $(C_{z_j} - C_{z_i})$ is the relative camera motion in the direction towards to the target between the time moments j and i, and s_j, s_i are the areas of the Bbox in two moments, respectively. i and j are any time index only if the straight distances between the two corresponding observation points and the target are different.

We modeled the target used in the real-world experiments 1:1 in the simulation scene, and collect multiple sets of Bbox areas s and D (deduce from relative camera motion and s by (2)) data to fit the distance estimation function by (1) and finally use it directly on the real machine. For the captor used in the real world, we use the UWB system to get camera motion, and obtain corresponding sets of s by object detection to complete the above process. Therefore, a UAV can estimate the distance of the detected objects by providing s to well-fitted distance estimation power functions (each detection class corresponds to a power function), even if the UAV remains stationary.

Finally, the relative spatial position (x, y, z) of the object are calculated from the estimated distance D and the camera FOV of horizontal and vertical (FOV_h, FOV_v) information with reference to the local coordinate system (Fig. 3A).

$$\begin{aligned} x &= \frac{2 \cdot p_x \cdot D \cdot \sin \frac{FOV_h}{2}}{W} \\ y &= \sqrt{D^2 - x^2} \\ z &= \frac{2 \cdot p_y \cdot D \cdot \sin \frac{FOV_v}{2}}{H} \end{aligned} \tag{3}$$

Where p_x, p_y are pixel values that the center of the boundary box deviates from the center of the image, and W, H are pixel values for the width and height of the image.

C. Local Position Map Generation

Our local position map is used to preserve the relative spatial location information of objects within the perception range around the UAV itself. As can be seen in (4), it records the position information of the detected object, the object type OT and the detection lost count LT. Where OT denotes the object class detected by visual perception, which is used for the proposed V-GRN model to generate the corresponding type of concentration field. LT is a counter for recording the detection lost times of an object, which is used to remove map elements that previously existed in the local position map and were not detected for 50 consecutive frames. With a sufficiently fast visual perception, the spatial position of the same object in the context changes little in short interval, so we can update the local position map based on self-motion compensation to cope with

Authorized licensed use limited to: Tsinghua University. Downloaded on April 09,2023 at 05:33:18 UTC from IEEE Xplore. Restrictions apply.

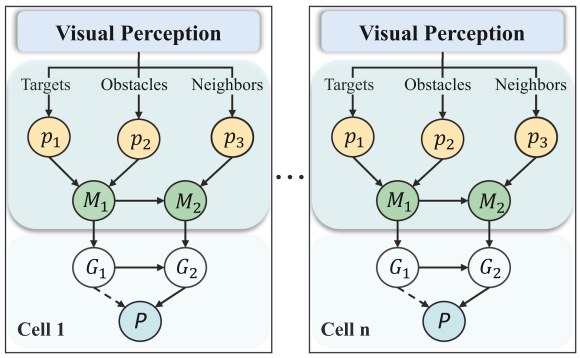

Fig. 4. A diagram of the V-GRN. A cell represents an individual UAV, which consists of a upper layer and a lower layer. In the upper layer, sensory proteins p_1, p_2 and p_3 receive the positions of targets, obstacles and neighbouring robots respectively to establish corresponding concentration fields. Protein M_1 fuses the concentration field from p_1, p_2, protein M_2 fuses the concentration field from M_1 and p_3, they affect the production of proteins G_1 and G_2, which are actuating proteins in the bottom layer representing entrapping patterns generation and the planner for movement calculating respectively, and they both affect the production of protein P, which ultimately determines the dynamic position of the UAV.

the problem that the object detector sometimes fails briefly.

$$object := [x, y, z, OT, LT] \tag{4}$$

D. Vision-Based Gene Regulatory Network

Different from commonly used GRN models [11], [16], the proposed V-GRN model (Fig. 4) removes the communication part and the access to get global information. All the inputs of p_1, p_2, and p_3 are derived from the aforementioned local position map. Establishing a local coordinate system with its own position as the center, p1, p2, and p3 generate concentration fields of targets, obstacles, and neighbors, respectively. These concentration fields are successively fused by protein M_1 (Fig. 5A) and protein M_2. The concentration field output from M_1 is used for G_1 to generate candidate entrapping patterns, which are the concentration contours around a target. The planner G_2 calculates the moving direction from the concentration field output by M_2, and generates the velocity commands for the flight controller. The adaptive entrapping pattern selected independently by each UAV should satisfy the following two conditions:

- The target is as close as possible to the centroid of the entrapping pattern.
- The minimum distance between the contour line and the target should surpass the predefined safe distance.

Concretely, in the process of patterns generation (Fig. 5B), a UAV first generates the concentration fields of the targets and obstacles through (5), (6), and (9). Where γ_i and β_j represent the positions of the i^{th} target and j^{th} obstacle, T_i, O_j represent the concentration field components formed by the i^{th} target and j^{th} obstacle, respectively, θ and k are regulation parameters. $\bigtriangledown^2$ is a Laplace operator, which is defined as the second derivative of T_i, O_j. A fused concentration field M is obtained according to (8), (9) with $Flag = 1$ (ignore concentration fields of neighbors, Fig. 5(e)) and is used to generate candidate entrapping patterns (the concentration contours around a target, Fig. 5(f)), corresponding to G_1 in Fig. 4. Note that the first derivative terms in (5)–(7) are zero since we do not consider the spatiotemporal process of protein diffusion.

$$\frac{dT_i}{dt} = \bigtriangledown^2 T_i + \gamma_i - T_i \tag{5}$$

$$\frac{dO_j}{dt} = \bigtriangledown^2 O_j + \beta_j - O_j \tag{6}$$

$$\frac{dN_m}{dt} = \bigtriangledown^2 N_m + \eta_m - N_m \tag{7}$$

$$\begin{aligned} M = sig\left(1 - \left(\sum_{i=1}^{n_t} T_i\right)^2, \theta, k\right) \\ + sig\left(\left(\sum_{j=1}^{n_o} O_j\right)^2, \theta, k\right) \\ + (1 - Flag) \cdot sig\left(\left(\sum_{m=1}^{n_n} N_m\right)^2, \theta, k\right) \end{aligned} \tag{8}$$

$$sig(x, \theta, k) = \frac{1}{1 + e^{-k(x-\theta)}} \tag{9}$$

In the process of calculating moving orientation, a UAV then generates the concentration field of all neighbors through (7) and (9), where η_m represents the position of the m^{th} neighbor, N_m represents the concentration field component formed by the m^{th} neighbor. M_2 fuses the concentration fields of the output of M_1 and the concentration field of neighbors using (8) with $Flag = 0$. Notably, in this work, we regard the neighbouring robots as obstacles during calculating the moving direction. In the process of entrapping patterns generation, the concentration field of neighbors contributes nothing, so that the candidate entrapping patterns are only determined by targets and obstacles. In the concentration field formed by M_2, we use the sampling method to find an optimal moving direction. Specifically, the UAV chooses the points from five circles, starting from the one with a radius of 0.5 m from the center of its rigid body to another four with an interval of an incremental radius increase of 0.6 m outwards. To consider a more global optimality, the direction with the minimum sum of concentration values on the five circles is selected among the 180 directions, with an interval of 2°.

With the desired moving direction and the selected pattern, the planner G_2 generates velocity control commands for the UAV flight controller, the velocity vector is simplified to the vector $\vec{v}$ in (10). It should be noticed that the generated $\vec{v}$ is not an accurate value, and the ratio of the motion in each direction is the goal pursued by G_2.

$$\vec{v} = [k \cdot m_s \cdot \sin\theta, k \cdot m_s \cdot \cos\theta, k \cdot \Delta z] \tag{10}$$

Where θ is the angle between desired direction and the forward direction of a UAV, k is a scaling factor, m_s is the dynamic movement step length, and Δz is the height difference relative to the target entrapped. Dynamic m_s is positively related to distance from the chosen pattern ($m_s = 0$ if the selected pattern

Authorized licensed use limited to: Tsinghua University. Downloaded on April 09,2023 at 05:33:18 UTC from IEEE Xplore. Restrictions apply.

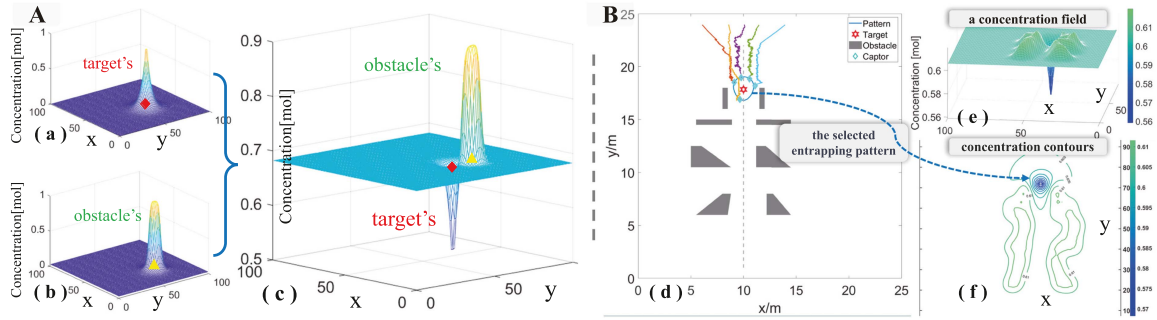

Fig. 5. Visualization of concentration fields and generation of candidate entrapping patterns. In part A, (a) is the concentration field formed by a target, (b) is the concentration field formed by an obstacle (or a neighboring UAV), and (c) is the fused concentration field. The concentration field of the target has an opposite sign compared with the concentration field of the obstacle, which is reflected in the negative sign preceding the T_i term in (8). In part B, (d) shows five robots dynamically entrapping a target in the obstacle scene, (e) presents a concentration field established by a robot within its perception range, (f) shows some contours of the concentration field in (e), where the contour lines around the target are the candidate entrapping patterns.

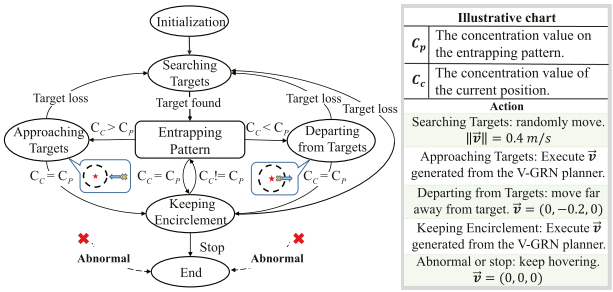

Fig. 6. The design of the FSM model. It receives data calculated from V-GRN and outputs the next behavior state of the UAV. The central part "Entrapping Pattern" involves the selected entrapping pattern from V-GRN. The calculated C_C and C_P are used for defining triggering conditions.

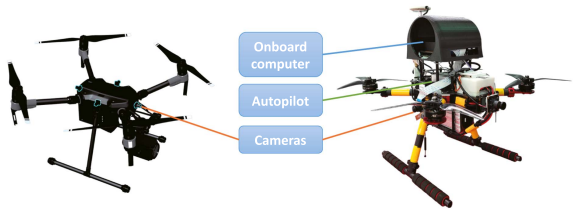

Fig. 7. They are captors used in simulation experiments (left one) and real-world experiments (right one). They both have four cameras installed in four directions: front, back, left, and right. Note that each realistic captor drone is equipped with a GPS module for obtaining its own ground speed to execute velocity commands, without sharing global position.

is reached), and we used a sigmoid form in this work to adjust the dynamic speed of the UAV.

E. Behavior Design of Swarm UAVs

We designed the same FSM model (Fig. 6) for each individual UAV to maneuver in dynamic environments. It is designed for the encirclement task and mainly contains searching and entrapping states, in which entrapping state includes behaviors of approaching targets, departing from targets, and keeping encirclement. Triggering conditions among states are also defined accordingly in the FSM model.

Each captor UAV starts its task at "Initialization," then enters the "Searching Targets" state. At this state, the captor will randomly select a lower concentration direction in the fused concentration field to search targets with velocity $\|\vec{v}\| = 0.4\,m/s$. It will keep this state until identifying at least one target. After discovering targets, the UAV starts generating and selecting an entrapping pattern by using V-GRN, and then the UAV executes the $\vec{v}$ command generated from the V-GRN planner to approach the target or keep encirclement. By comparing the calculated concentration value C_C of the current position and the concentration value C_P in the selected pattern, it chooses the next behavior state, including approaching targets, departing from targets, and keeping encirclement. If the target stops moving, the captor will finally stay in its selected entrapping pattern.

III. Experimental Validation

A. Hardware Setup

1) Simulation Setup: To achieve a more realistic verification, we opt for AirSim [23] and the Unreal Engine 4 simulation platform for most simulation experiments. We replaced the default quadcopter model in AirSim with our customized quadcopter model, which is modeled 1:1 according to DJI MATRICE 200, as shown in Fig. 7 (left one). It provides four camera frames in the front, back, left, and right to obtain surrounding environment information. We used the quadcopter as the captor and a dual-rotor drone as the target.

We built simulation environments and verified the proposed algorithm on two computers connected within a local area network, equipped with Intel i9-11900 k and NVIDIA Quadro RTX 4000, one for UE4 simulation rendering and the other for running the proposed algorithm for all UAVs.

2) Real-World Setup: We used a custom-built quadcopter named H350 as the captor and DJI MATRICE 200 quadcopter as the target. Each captor is equipped with four wide-angle RGB cameras (two CSI and two USB cameras), each of them providing a horizontal and vertical FOV of 120° and 90° respectively for obtaining omnidirectional visual inputs. We acquired each camera frame in a resolution of 640x480 at a frequency of 20 Hz and stitched the four frames into one image. We mounted a Jetson Xavier NX with SSD for each captor and a DJI N3 as an autopilot (Fig. 7, right one).

Authorized licensed use limited to: Tsinghua University. Downloaded on April 09,2023 at 05:33:18 UTC from IEEE Xplore. Restrictions apply.

TABLE I
NETWORKS COMPARISON RESULTS

Network model	mAP(%)	AP50(%)	Size(MB)	Inference speed (FPS)	
				no-stitching	stitching
YOLOv5m	**83.2**	**98.5**	42.5	4.01	8.57
YOLOv5m-TRT	/	/	51.6	11.86	13.44
YOLOv5s	80.8	98.1	14.4	5.25	17.65
YOLOv5s-TRT	/	/	18.7	13.83	31.25
Ours	80.8	97.6	**3.01**	5.71	24.17
Ours-TRT	/	/	7.89	**17.56**	**45.46**

B. Object Detection and Position Estimation

Our customized network model has 269 layers and 1.3 M parameters (about 17% of the YOLOv5s), with a size of only 3.01 MB. Without being pretrained, this network is used to train our detector model on our simulation and real-world datasets (200 epochs), which include 1849 images with more than 5300 bounding boxes and 5567 training images with more than 5700 bounding boxes, respectively.

In detection tasks, we set the IoU threshold to 0.5, the confidence threshold to 0.6, and the inference size to 1280 to detect the stitched image. Additionally, we use a 16-bit floating point TensorRT engine to optimize the network model, the networks comparing results of YOLOv5m, YOLOv5s, and ours tested in Jetson Xavier NX are shown in Table I, where TRT represents the TensorRT-optimized model.

We used the pose data provided by the UWB positioning system to calculate the visual positioning error. For each observation point, more than twenty visual position estimation data were taken to obtain the average localization error.

C. Target Entrapping Experiments

To verify that the proposed V-GRN method can guide the swarm of UAVs to emerge adaptive entrapping formations in dynamic environments, we set up a series of simulation scenarios on the AirSim simulator, including open scenarios, narrow road scenarios, and random obstacle scenarios.

The experimental process for target encirclement is the following. The captors were placed behind the target at the start, who constantly perceiving their surroundings and searching for targets using their visual sensors independently. Once detecting one or more targets, the captors who observe the target will switch from the search state to the entrapment state. The entrapping pattern obtained from the concentration field can be adaptively transformed in a dynamic environment. In the simulation experiments, the max velocity of captors is set to $5\,m/s$, while in the real-world experiments, we limit the maximum speed to $1\,m/s$ due to the large delay in acquiring raw image data from CSI cameras (500ms+).

We use the following quantitative evaluation metrics to evaluate the performance of the proposed method.

- Average entrapment distance error $\overline{d}$: the average distance between the swarm UAVs and the selected entrapping pattern (Fig. 8, part A), d_i is the current distance error of the i^{th} UAV from its selected pattern.

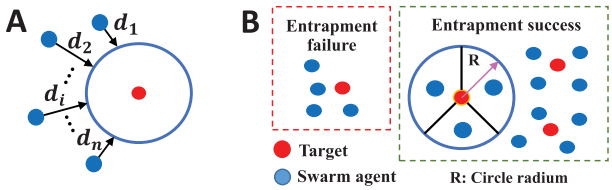

Fig. 8. The schematic diagram of the entrapment distance error and the success entrapment. Part A shows the entrapment distance error (d_i) between agents and their selected entrapping patterns. In part B, we divide the circle area around the target into three equal parts. Successful entrapment is defined to be achieved when more than one drone appears in each part.

$$\overline{d} = \frac{\sum_{i=1}^{n} d_i}{n} \tag{11}$$

- Average speed: The average speed of the captors from spotting the target to successful entrapment.
- Success rate: Success rate is defined as the number of successful entrapments (part B of Fig. 8) divided by the total number of experiments.

IV. RESULTS

A. Vision

We show the model comparison results in Table I. The prediction head we replaced is more sensitive to tiny instances. Therefore, the detector maintains the same mean Average Precision as YOLOv5s even if the number of convolution kernels is halved, and 45.5% faster in TensorRT-optimized (TRT) with images stitched mode. The detector we deployed in the real-world experiment achieves an average precision of 97.6% (AP_{50}) and 80.78% (AP) at a confidence threshold of $p^{conf} = 0.1\%$ on the hold-out validation set containing 650 images after 200 epochs of training.

The monocular visual relative localization error is mainly determined by the accuracy of bounding boxes, while the detector provides slightly different bounding box information from different perspectives. As a result, the monocular position estimation system presents a positioning error under 11% within 10 m, and 13% within 15 m.

B. Swarm

In the UE4 simulation, we validated the proposed algorithm through extensive experiments in three scenarios: open, narrow roads, and random obstacle scenarios. The experimental results are as follows.

1) In the Open Scenario: Captors entrap the target at a dynamic speed ranging from $0\,m/s$ to $5\,m/s$ after spotting the target. As an example shown in Fig. 9, it can be seen that captors can quickly surround a static target more than 10 meters ahead within 10 seconds. Then, the target moved at a low speed, and the captors remained in an encirclement mode. After a few seconds, the target attempted to escape from the encirclement pattern at high speed, and the captors started to catch up and entrap the target once again in a short period of time. Fig. 10 shows its entrapment distance error during the entire experimental

Authorized licensed use limited to: Tsinghua University. Downloaded on April 09,2023 at 05:33:18 UTC from IEEE Xplore. Restrictions apply.

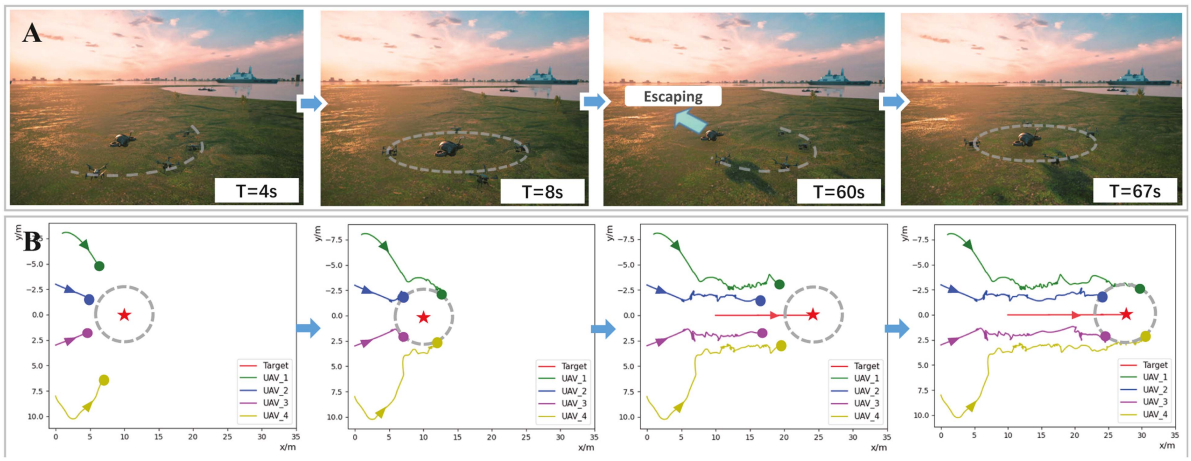

Fig. 9. The entire process of UAVs swarm entrapping a dynamic target. (A) Experimental process. (B) The trajectories of all UAVs, they are obtained by using GPS position information in the simulator. The corresponding entrapment distance error of all drones can be seen in Fig. 10.

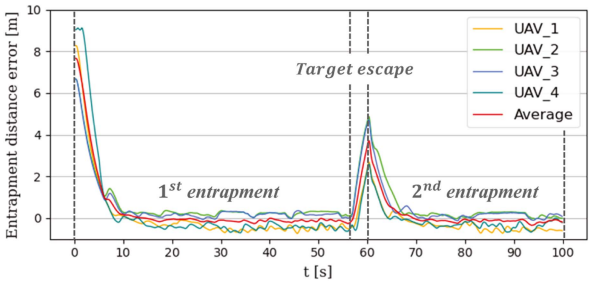

Fig. 10. The average entrapment distance error of UAVs.

TABLE II
EXPERIMENTAL RESULT OF SIMULATION

Initial average distance (m)	Success rate (%)			Average speed (m/s)
	6s	10s	14s	
6	0	93	100	0.76
10	80	100	100	2.01
14	73	100	100	2.53

process, which demonstrates the effectiveness of the proposed method in emerging entrapment behavior. We calculated the successful entrapment rates for more than 20 experiments at different average distances and different time consumption, as can be seen in Table II.

2) In the Narrow Road Scenario: Captors first enter the parallel obstacle lanes in an encirclement mode (Fig. 11A); the entrapping patterns around the target change from a circular to an elliptical shape due to the observed obstacles. Captors maintain an elliptical formation to dynamically entrap the target, guided by their own concentration field generated in real-time by V-GRN. When the target and the captors enter the conical obstacle lanes (Fig. 11B), the concentration contours around the target present a water droplet shape, with the pattern formation of the captors changed accordingly.

3) In the Random Obstacle Scenario: When the captors and the target enters a scene with randomly distributed obstacles

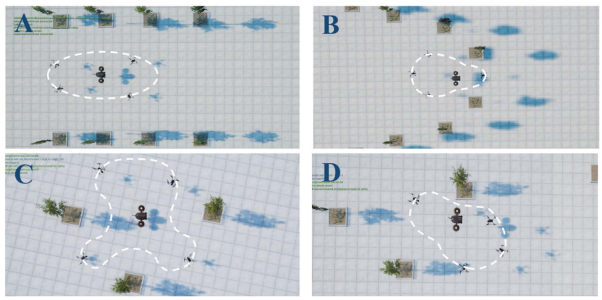

Fig. 11. The entrapping patterns generated by the swarm of UAVs under different obstacle scenes. Among them, part A is the scene with parallel obstacle lanes, part B is the scene with conical obstacle lanes, and parts C and D are the scenes with randomly distributed obstacles.

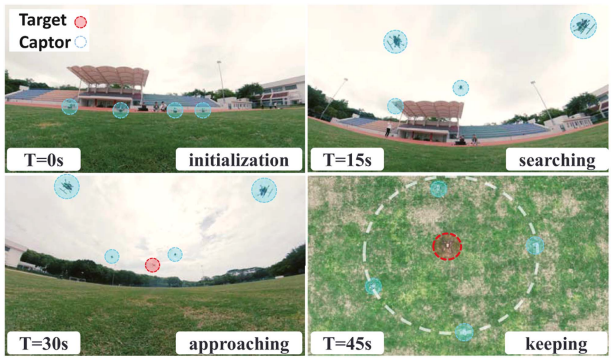

Fig. 12. The processing of UAVs swarm behavior emergence in outdoor experiments (4 captors from initialization to successful entrapping).

(Fig. 11C, D), the proposed V-GRN can still generate adaptive entrapping patterns to guide the captors to switch to different entrapping formations.

4) Real-World Experiments: As shown in Fig. 12, four captors were placed 15 meters behind the target where they can not spot the target at first. While the start instruction was given at 5 s, these captors switched to the searching state. At 20 seconds, the captors in front found the target and then switched

Authorized licensed use limited to: Tsinghua University. Downloaded on April 09,2023 at 05:33:18 UTC from IEEE Xplore. Restrictions apply.

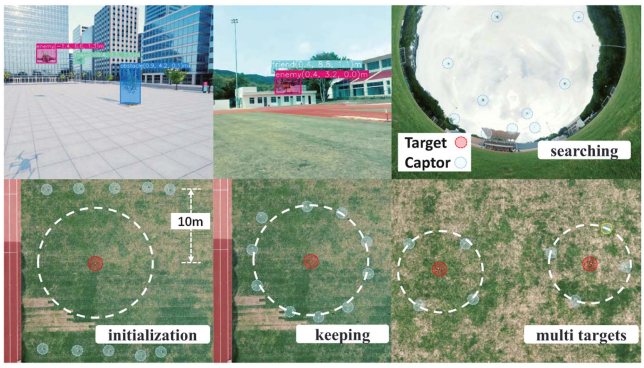

Fig. 13. More experiments. Visual perception and large-scale UAVs swarm behavior emergence in outdoor experiments.

to approaching state. The rest also completed state transition in their 30 s. The swarm of captors completed the entrapping task without collision at an average time of 50 s.

Besides, to demonstrate that the proposed approach is still effective in large-scale UAV swarms, we increase the number of captors to 10. The results can be seen in Fig. 13.

The details of the above experiments can be seen in the online video: https://youtu.be/R1SD1YlC-dc

C. FSM

The system status is monitored in real-time. Once an unknown situation or known failure occurs, the system will jump to the abnormal processing state, and then the UAV will keep hovering to wait for instructions. In more than 30 real-world experiments, the FSM can still retain the normal state transition in encirclement tasks, it can demonstrate that the behavioral-designed FSM is robust and effective.

V. Conclusion

We proposed a V-GRN model and an omnidirectional visual perception algorithm that enables multiple UAVs to complete the target entrapment task and emerge adaptive swarm formation in dynamic environments. The proposed method does not rely on inter-agent communication and external positioning information (note that in this work we used GPS to obtain ground speed and keep hovering, it can be replaced by visual-inertial odometry or optical flow module), which can still work in GNSS-denied environments or in situations where the existing communications are unreliable. Our simulation and real-world experiments demonstrate that the proposed approach is robust and effective, which can resist partial agent failures and still fulfill the entrapping task. Notably, we increased the number of captors to 10 in real-world entrapping experiments (Fig. 13), which to our best knowledge is the first work reported in this line.

Our work paves the way for the practical application of a swarm of UAVs in the vision-based encirclement. As a seminal work, the proposed work needs improvements in several directions. For instance, the monocular position estimation algorithm can be improved so that the system can estimate the position even if the object is not detected in the vision. Second, enhance the autonomous navigation ability of the UAV to improve the encirclement speed, in this work, the movement of the UAV is determined by the optimal direction of the concentration in the concentration field, this moving strategy may be inefficient or even fail if targets are tightly surrounded by a lot of obstacles.

References

[1] X. Zhou et al., "Swarm of micro flying robots in the wild," *Sci. Robot.*, vol. 7, no. 66, 2022, Art. no. eabm5954.
[2] L. Briñón-Arranz, A. Seuret, and A. Pascoal, "Circular formation control for cooperative target tracking with limited information," *J. Franklin Inst.*, vol. 356, no. 4, pp. 1771–1788, 2019.
[3] G. Vásárhelyi, C. Virágh, G. Somorjai, T. Nepusz, A. E. Eiben, and T. Vicsek, "Optimized flocking of autonomous drones in confined environments," *Sci. Robot.*, vol. 3, no. 20, Jul. 2018, Art. no. eaat3536.
[4] A. Macwan, J. Vilela, G. Nejat, and B. Benhabib, "A multirobot path-planning strategy for autonomous wilderness search and rescue," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1784–1797, Sep. 2015.
[5] Y. Gao, C. Bai, L. Zhang, and Q. Quan, "Multi-UAV cooperative target encirclement within an annular virtual tube," *Aerosp. Sci. Technol.*, vol. 128, 2022, Art. no. 107800.
[6] Y. Cai et al., "The behavior design of swarm robots based on a simplified gene regulatory network in communication-free environments," in *Proc. Int. Workshop Adv. Comput. Intell. Intell. Inform.*, 2021.
[7] A. Hafez, M. Iskandarani, S. Givigi, S. Yousefi, and A. Beaulieu, "UAVs in formation and dynamic encirclement via model predictive control," *IFAC Proc. Vol.*, vol. 47, no. 3, pp. 1241–1246, 2014.
[8] H. De Jong, "Modeling and simulation of genetic regulatory systems: A literature review," *J. Comput. Biol.*, vol. 9, no. 1, pp. 67–103, 2002.
[9] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "Leader–follower formation control of nonholonomic mobile robots with input constraints," *Automatica*, vol. 44, no. 5, pp. 1343–1349, 2008.
[10] D. J. Sumpter, *Collective Animal Behavior*. Princeton, NJ, USA: Princeton Univ. Press, 2010.
[11] Y. Jin, H. Guo, and Y. Meng, "A hierarchical gene regulatory network for adaptive multirobot pattern formation," *IEEE Trans. Syst., Man, Cybern. B*, vol. 42, no. 3, pp. 805–816, Jun. 2012.
[12] X. Fu, J. Pan, H. Wang, and X. Gao, "A formation maintenance and reconstruction method of UAV swarm based on distributed control," *Aerosp. Sci. Technol.*, vol. 104, 2020, Art. no. 105981.
[13] H. Oh, A. R. Shirazi, C. Sun, and Y. Jin, "Bio-inspired self-organising multi-robot pattern formation: A review," *Robot. Auton. Syst.*, vol. 91, pp. 83–100, 2017.
[14] H. Guo, Y. Meng, and Y. Jin, "A cellular mechanism for multi-robot construction via evolutionary multi-objective optimization of a gene regulatory network," *BioSystems*, vol. 98, no. 3, pp. 193–203, 2009.
[15] H. Guo,Y. Meng, and Y. Jin, "Swarm robot pattern formation using a morphogenetic multi-cellular based self-organizing algorithm," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 3205–3210.
[16] H. Oh and Y. Jin, "Evolving hierarchical gene regulatory networks for morphogenetic pattern formation of swarm robots," in *Proc. IEEE Congr. Evol. Comput.*, 2014, pp. 776–783.
[17] A. Strandburg-Peshkin et al., "Visual sensory networks and effective information transfer in animal groups," *Curr. Biol.*, vol. 23, no. 17, pp. R709–R711, 2013.
[18] R. Bastien and P. Romanczuk, "A model of collective behavior based purely on vision," *Sci. Adv.*, vol. 6, no. 6, 2020, Art. no. eaay0792.
[19] F. Schilling, F. Schiano, and D. Floreano, "Vision-based drone flocking in outdoor environments," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2954–2961, Apr. 2021.
[20] G. Jocher, A. Stoken, and J. Borovec, "ultralytics/yolov5: V5.0 - YOLOv5-p6 1280 models, AWS, supervise.ly and youtube integrations,"Apr. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4679653
[21] M. A. Haseeb, J. Guan, D. Ristic-Durrant, and A. Gräser, "DisNet: A novel method for distance estimation from monocular camera," in *Proc. 10th Plan., Percep. Navigation Intell. Veh., IROS*, 2018.
[22] B. A. Griffin and J. J. Corso, "Depth from camera motion and object detection," in *Proc. IEEE/CVF Conf. Comp. Vis. Pattern Recognit.*, 2021, pp. 1397–1406.
[23] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High fidelity visual and physical simulation for autonomous vehicles," in *Proc. Field Serv. Robot.*, 2017, pp. 621–635. [Online]. Available: https://arxiv.org/abs/1705.05065

Authorized licensed use limited to: Tsinghua University. Downloaded on April 09,2023 at 05:33:18 UTC from IEEE Xplore. Restrictions apply.