Proceedings of 2013 IEEE
International Conference on Mechatronics and Automation
August 4 - 7, Takamatsu, Japan

3D Exploration Priority Based Flocking of UAVs

Abdullah Al Redwan Newaz, Geunho Lee, Ferdian Adi Pratama, Nak Young Chong
Robotics Laboratory
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa, Japan 923-1292
{*redwan, geun-lee, ferdian, nakyoung*}*@jaist.ac.jp*

***Abstract*—This paper presents a 3D flocking algorithm for a team of unmanned aerial vehicles (UAVs), where each member is equipped with limited range sensors and computational resources. A minimal leader-follower communication scheme is proposed for maneuvering huge swarm of UAVs. The proposed triangular formation compacts the overall group size. Even though UAVs are considered for tactical, remote monitoring, and surveillance purposes in both indoor and outdoor environments, it is very difficult to achieve autonomous aerial flocking in unknown cluttered environments. Specifically, greater attention is placed to reduce computational complexity for on-board implementation. We demonstrate the efficiency of our algorithm in a real world scenario with the V-REP simulator employing a group of five UAVs.**

***Index Terms*—aerial swarm, online path planning, flocking, triangular formation, computational complexity.**

I. Introduction

In recent decades, unmanned aerial vehicles (UAVs) have attracted much attention due to their wide range of applications and reasonable manufacturing cost. Among different types of UAVs, researchers increase their focus on rotor wing UAVs particularly Quadrotor UAVs, because their kinematics offers low speed maneuvering and hovering. Quadrotor UAVs appear in miniature form in contrast to typical aerial vehicles, whereby the possibility of aerial vehicle swarming becomes a reality. Flocking is one of the basic elements of aerial swarm behavior. Considerable effort has been directed toward understanding how a group of autonomous creatures creates a certain form of clusters. Similar problems have been studied in ecology and theoretical biology, in context of animal aggregation and social cohesion in animal groups [1] [2].

Reynolds [3] proposed the basic model that was later modified in different ways. Delgado-Mata *et al.* [4] introduced the effects of fear by observing the activities of Olfaction to transmit emotion between animals through pheromones modeled as particles in a free expansion gas. Hartman and Benes [5] incorporated a complementary force to the alignment for a leadership change, where the steer defines the chance of the boid to become a leader and try to escape. Hemerlijk and Hildenbrandt [6] used attraction, alignment and avoidance and extended the algorithm with a number of traits of starlings given by

1) birds fly according to the fixed-wing aerodynamics, while rolling and turning
2) they coordinate with a limited number of neighbors
3) staying above a sleeping site is given priority and when moving outwards the sleeping site, they return to it by turning,
4) fixed relative speed is proposed.

The authors claimed that the specifics of flying behavior as well as large flock size and low number of interaction partners were essential to the creation of the variable shape of flocks of starlings. Related problems have become a major thrust in system and control theory [7]–[11]. Vicsek *et al.* [12] proposed the leader following model, in which one agent acted as a group leader and others would just follow the aforementioned cohesion/separation/alignment rules.

Meanwhile, Lee and Chong [13] proposed the equilateral triangle lattice model in establishing selective local interactions among neighboring robots. They claimed that the equilateral triangle can reduce the number of robots in a given location, and improve the network connectivity and hole repair capability [14], [15].

Inspired by the results of [12] and [13], this paper introduces the *communication model* to the existing basic models of flocking behavior, where small intermediate *equilateral triangles* are considered to communicate with neighbors. Each intermediate group consists of three members *i.e., one leader and two followers*. We have previously proposed the exploration priority based heuristic approach (EPBHA) for UAV collision-free path planning with lesser computational complexity in cluttered environments. EPBHA reorients equilateral triangles into arbitrary triangular shapes depending on the type of obstacle and allows flexible path planning to avoid collisions. The communication model increases the overall team efficiency, since every robot is not required to find the position of obstacle that will be discovered by one of the team members.

II. Problem Statement

We categorize aerial swarm missions further into three cases: 1) navigate in obstacle-free environments 2) build/maintain a team via internal communications and 3) avoid obstacles and escape from a deadend passageway. Case 1) lies in more general context, which means if there is free space to move, the robot finds the minimum distance path toward a goal position. Case 2) and Case 3), however, must conform to several crucial conditions: how to build a large group, how to avoid unexpected obstacles that appear in the path of navigating robots. A complex and unpredictably changing environment makes it difficult to accomplish safe path

978-1-4673-5560-5/13/$31.00 ©2013 IEEE

planning. Moreover, vision sensors increase the computational complexity that makes it difficult to accommodate on-board implementation requirements. Therefore, without having any *a priori* knowledge of the environment, this paper proposes a new heuristic approach to allow a group of UAVs consisting of two intermidate small groups to navigate through complex terrains, only based on six infrared sensors, ensuring a near-constant computational complexity.

Now we address the flocking problem of a group of UAVs in unknown environments as follows:

> *Assuming a group of UAVs equipped with limited range sensors exploring an arena, where different types of unknown obstacles exist, how to make it reaches a goal position avoiding the obstacles with comparatively little computational cost?*

This problem can be decomposed into simpler problems:

- **Sub-problem 1** (*path planning for free space*) How do all the members of a group travel a minimum possible distance in an obstacle free environment?
- **Sub-problem 2** (*group formation*) How to build and maintain a group?
- **Sub-problem 3** (*path planning for obstacle avoidance*) How does a group re-plan its next position, while avoiding any obstacles in its path?

III. Algorithm Description

Definition 1 (Triangular Configuration)**.** *Given the leader robot r_l and neighbour robots r_{f1} and r_{f2}, a triangular configuration is defined as the set of their distinct positions $\{P_l, P_{f1}, P_{f2}\}$ denoted by*

$$T_i = \{P_l, P_{f1}, P_{f2}\}.$$

One half of the interior angle $\angle P_{f1}P_lP_{f2}$ is denoted by θ.

Definition 2 (Sensing Range)**.** *Each robot is equipped with 6 proximity sensors detecting up to S_d with a $45°$ angle of coverage*

Definition 3 (Inter-robot Distance)**.** *Given T_i, a safe distance is configured between the leader and follower robots, which must be greater than the sensing range (S_d) of individual robots.*

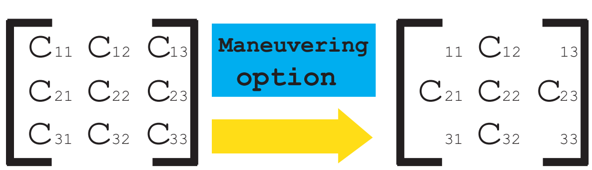


Fig. 1: Reduced cost assessment

Definition 4 (Input Description)**.** *The leader robot knows its own position and goal position but does not know* a priori *the obstacle position. The distance between one coordinate and the next coordinate is defined as step length d. The value of d is responsible for smooth motion planning which is propositional to the velocity of robot. The goal position is divided into one goal for the XY plane and another for the YZ plane. After reaching a goal in one plane, the goal is automatically shifted to the other plane.*

Definition 5 (Coordinate Cost)**.** *A coordinate cost is defined by the difference between the current position (x_1, y_1) and the next position (x_2, y_2) given by*

$$Cost := A \times (x_1 - x_2) + B \times (y_1 - y_2),$$

where A and B are arbitrary even constants for emphasizing the straight forward (X-axis) or straight sideward (Y-axis) movements instead of the diagonal movements travel. If $A > B$, then the robot moves forward or backward, while $A < B$ indicates left or right movements. Fig. 1 represents the cost reduction assessment where diagonal movement cost calculation is omitted by introducing maneuvering options.

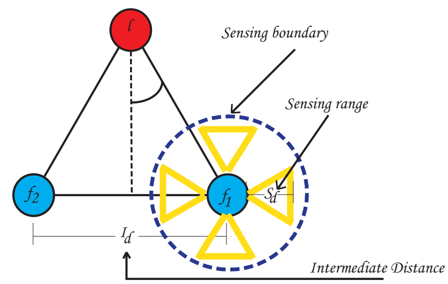


Fig. 2: Triangular group formation

Now the proposed exploration priority based heuristic aerial swarming algorithm is divided into three major functions:

1. Path planning in obstacle free environments
2. Communication to follower robots
3. Path planning in cluttered environments

A. Path planning in obstacle free environment

The leader robot moves forward to the goal position based on *Manhattan distance*. By introducing *maneuvering options*, we restrict diagonal movements. Therefore, each robot is capable to move only six directions, *i.e.*, forward, backward, left, right, upward, and downward, respectively. Fig. 2 represents the triangular configuration for group formation, where the red circle represents the leader robot and the blue represents followers, respectively. The leader robot determines the positions for adjacent follower robots with respect to its current position. A safe distance is ensured by maintaining inter-robot distances greater than their sensing range. The leader robot plans for the equilateral triangular configuration for its group. Every robot creates four square grids, while moving towards the goal position. Among the four grids indicating four different coordinates, robots choose the best coordinate for their next movement by calculating the minimum cost.

Grid making The incremental distance between the parent coordinates and next coordinates is termed as *d*. For the next set position of robots, one coordinate is chosen among four neighboring coordinates. The value of *d* neds to be kept as small as possible to ensure lesser probability of colliding with obstacles.

Algorithm III.1: GRID MAKING(x, y, d)

$$\textbf{for } i \leftarrow 1 \textbf{ to } 4 \\ \textbf{do} \begin{cases} \textbf{for } j \leftarrow 1 \textbf{ to } 3 \\ \textbf{do} \begin{cases} Grid[i][1] = i \\ \textbf{comment: } \text{indexing} \\ Grid[i][2] = x \pm d \\ \textbf{comment: } \text{next x coordinate} \\ Grid[i][3] = y \pm d \\ \textbf{comment: } \text{next y coordinate} \end{cases} \end{cases}$$

Maneuvering Option A sensor reports a certain range of numeric values, when it finds any obstacle within its sensing range. The available movement options are determined by counting the number of sensors that do not detect anything.

Algorithm III.2: MANEUVERING OPTION(g, h, i)

$$\textbf{for } i \leftarrow 1 \textbf{ to } 6 \\ \textbf{do} \begin{cases} \textbf{if } \text{value\_of\_sensor[i]} > \text{sensor\_range} \\ \textbf{then} \begin{cases} count+ = 1 \\ \textbf{comment: } \text{number of activated sensors} \\ \text{movement\_option} = 6 - count \end{cases} \end{cases}$$

Cost estimation The degree of freedom of robots is restricted by introducing maneuvering options, whereby straight or perpendicular movements are more emphasized than diagonal movements. Therefore, costs of diagonal movements are higher than straight or perpendicular movements. This cost estimation is valid when there is no obstacle around the robot.

Moving to minimum cost point The robot finds an optimal coordinate for its next set position and relocates its position to this coordinate.

B. Communication to follower robots

There are two kinds of goal for flocking, such as the user defined goal for overall team maneuver and the leader defined intermediate goals. The leader robot is assumed to know the user defined goal and it creates new goals for its followers equipped with a wireless transceiver while traveling every new grids. To reduce communication between robots, only the leader sends the position information to its followers and does not take any feedback from them. Similarly, a follower which is the leader of the next triangular group sends its position to its followers. Follower robots estimate a safe distance from the information of leader position.

$$Send = \begin{cases} \texttt{Position} & \textbf{if } obstacle = \textbf{false} \\ \texttt{NULL} & otherwise \end{cases} \tag{1}$$

C. Path planning in cluttered environment

One half the interior angle between the leader and follower robots is denoted by θ whose unit is $degree/100$. Let $P(f_1|l)$ indicate the probability of obstacle existence with respect to the leader robot sensing value, whereas $P(l)$ represent the probability of obstacle existence with respect to the position of leader. The value of $P(l)$ is always 1 since the leader only communicates when it finds an obstacle. Therefore the probability of existing obstacle with respect to the follower position can be given by

$$\begin{aligned} P(f_1) &= P(f_1|l) \times p(l) \\ &= (1 - \theta) \times 1 \end{aligned}$$

It is obvious that the path distortion (*i.e.*, probability of unexpected obstacle in the navigating path) depends on the angle θ, since the leader robot does not know the situation perpendicular to follower robots due to the triangular formation. We use the EPBHA algorithm for obstacle avoidance. In swarming purposes, robots do not communicate to others while avoiding any obstacle. This behavior enhances the efficiency of avoiding obstacles within a short period of time but restricts the minimum limit of interior angle. In Fig. 3, S_d and I_d represent the sensing range and inter-robot distance. Moreover, the blue dotted circle is the sensing boundary based on radius S_d. While the leader robot avoids an obstacle by changing its position to the left or right side, it should be longer than $(I_d \div 2)$, since the leader movements also affect the follower path plan. This function is further divided into two functions:

a) **Obstacle definition and avoidance:** Robots identify the type of obstacle from the number of active sensors as detailed below:
   1) **Easy: single-sided obstacles** The number of active sensors is one, *e.g.*, either front, left, or right. Robots avoid this type of obstacle by moving towards the goal direction.
   2) **Medium: partition-type obstacles** The number of active sensors is more than one except upward and bottom sensors, *e.g.*, either Π or Γ shape. Robots pass over the obstacle.
   3) **Hard: one-side open box shape obstacles** The number of active sensors is more than one, including upward or downward sensors. Robots move backward diagonally.

b) **Waiting for the leader instruction**: Since the leader does not communicate while avoiding obstacles, followers wait after arriving in their goal given by the leader. However, as soon as a new goal is informed, they boost their speed to achieve that goal position.

Below is a sketch of the proposed algorithm, incorporating the above-mentioned function modules: the common goal position is defined in terms of the leader position, therefore the given orientation of group is automatically adjusted for others.

Algorithm III.3: COMMON GOAL PLANNING(x, y)

repeat
GRIDMAKING()
read sensor value
if obstacle exist
then EPBHA()
else COSTESTIMATION()
FINDMINIMUMINDEX() **and** $send(Position)$
$compare(UAVPos(x,y), goalPos(x,y))$
if $goalPos(x,y) - UAVPos(x,y) ==$ desired accuracy
then xy search is finished
until xy search is not finished

Algorithm III.4: INDIVIDUAL GOAL PLANNING(y, z)

repeat
GRIDMAKING()
read sensor value
if obstacle exist
then EPBHA()
else COSTESTIMATION()
FINDMINIMUMINDEX()
$compare(UAVPos(y,z), goalPos(y,z))$
if $goalPos(y,z) - UAVPos(y,z) ==$ desired accuracy
then yz search is finished
until yz search is not finished

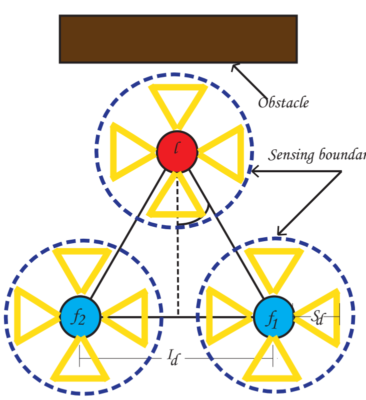


Fig. 3: Path planning for obstacle avoidance

Algorithm III.5: FOLLOWER ROBOT PATH PLANNING(x, y)

repeat
RECEIVE$(leaderPosition)$
ESTIMATE$(ownPosition)$
GRIDMAKING()
read sensor value
if obstacle exist
then EPBHA()
else COSTESTIMATION()
FINDMINIMUMINDEX() **and** $send(Position)$
$compare(UAVPos(y,z), goalPos(y,z))$
if $goalPos(y,z) - UAVPos(y,z) ==$ desired accuracy
then yz search is finished
until yz search is not finished

IV. SIMULATION RESULT

Six infrared sensors having $0.5m$ range and $45°$ angle of detection are mounted on top, front, right, left, back, and bottom of every robot, respectively. The sensor data is assumed to be accurate, noiseless, and achieved instantaneously. As this paper does not deal with a low level control system, dead reckoning and/or other aerodynamics errors are assumed to be negligible. There is no *a priori* information such as map or pre-specified path. The initial position of every robot is in a stable hovering position. Each robot is capable of avoiding collisions and re-planning its path in real time. If the path planner fails to generate a safe path within a bounded time, collisions may result. It is advantageous to form a small group to minimize communication delays between robots and expect fast responses from followers. The proposed algorithm accelerates the overall team speed, and followers require comparatively less time to find their path and effectively carry out other missions. Thus we can create a binary tree [16] structure to form a large team, where the node will be the leader and leaves will represent the followers. Once a huge triangular shape is formed, the members on the boundary have main responsibilities to make decisions for overall team maneuvering. The leader does not communicate to followers while avoiding obstacles. Therefore, followers wait until they receive obstacle free path coordinates from the leader. This will boost the speed of path planning for followers. As shown in Figure 4, followers exhibit faster translation compared to the leader.

In short, every follower decides its path either: 1) self avoidance using its sensors, or 2) prediction of obstacle position using the leader's heads up. Predicting obstacle positions offers fast path planning for followers, while self avoidance ensures safer path planning despite of sensing errors. Open loop communication also increases overall communication speed, since the leader does not need any feedback to be confirmed regarding the exact positions of its followers. We have transformed 3D path planning into two separate XY search and YZ search problems to reduce computational complexity and improve planning efficiency.

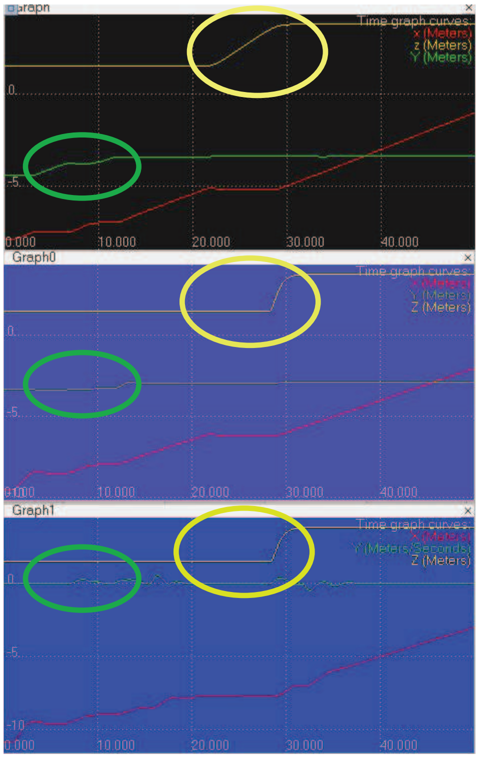
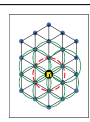
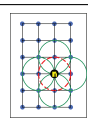
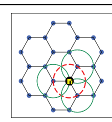


Fig. 4: Vehicle trajectory analysis. The top graph represents the leader trajectory, and the middle and bottom graphs represent the follower trajectories (robot no. 3 and 5 depicted in Fig. 5(c)), respectively.

TABLE I: Comparison of lattice typed network pattern

	Triangle	Square	Hexagon
Geometry			
Coverage Area	$\frac{\sqrt{3}}{2} \times n \times (d_u)^2$	$n \times (d_u)^2$	$\frac{3\times\sqrt{3}}{4} \times n \times (d_u)^2$
Coverage density	$\frac{1.2}{(d_u)^2}$	$\frac{1}{(d_u)^2}$	$\frac{0.78}{(d_u)^2}$
connectivity	6	4	3

As seen in Table 1 [13] , the triangle geometry offers higher converge density and better connectivity. While following the leader instructions, two neighboring robots may collide with each other in the same plane during obstacle avoidance. To cope with this problem, robots create a different layer while passing through a narrow passageway. With this layered formation, robots maintain the pre-defined triangular geometry but do not fly at the same height. This feature increases the volume flow rate of flocking, when passing through a narrow opening. By using [17] illustrated in Fig. 5, we have defined the common goal and individual goals and decided the common goal position based on the first leader robot position (robot no. 1), while followers maintain their geometric shape by maintaining the triangular configuration. After acquiring the common goal position, the leader robot moves toward its individual goal position that is located underneath the table.

To recapitulate all, we propose a heuristic approach to aerial flocking which ensures *lesser computational complexity, high volume flow rate, and lesser communication delay.*

V. Conclusion

Aerial flocking in cluttered environments is challenging due to limited hardware resources and proper swarm behaviors. As sticking to neighbor robots is not efficient in terms of overall team maneuvering, a minimal internal communication scheme was proposed to increase the team efficiency, where the triangular geometry offered better network connectivity and coverage density. Furthermore, to cope with computational intractability for on-board real-time computation, we implemented the exploration priority based approach yielding a new aerial flocking controller for low-cost UAVs.

References

[1] C. M. Breder, "Equations descriptive of fish schools and other animal aggregations," 1954.

[2] H. Tanner, A. Jadbabaie, and G. Pappas, "Stable flocking of mobile agents, part i: fixed topology," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 2, 2003, pp. 2010–2015 Vol.2.

[3] O. O'Loan and M. Evans, "Alternating steady state in one-dimensional flocking," *Journal of Physics A: Mathematical and General*, vol. 32, no. 8, p. L99, 1999.

[4] C. Delgado-Mata, J. I. Martinez, S. Bee, R. Ruiz-Rodarte, and R. Aylett, "On the use of virtual animals with artificial fear in virtual environments," *New Generation Computing*, vol. 25, no. 2, pp. 145–169, 2007.

[5] C. Hartman and B. Benes, "Autonomous boids," *Computer Animation and Virtual Worlds*, vol. 17, no. 3-4, pp. 199–206, 2006.

[6] C. K. Hemelrijk and H. Hildenbrandt, "Some causes of the variable shape of flocks of birds," *PLoS One*, vol. 6, no. 8, p. e22479, 2011.

[7] J. R. Spletzer and C. J. Taylor, "Dynamic sensor planning and control for optimally tracking targets," *The International Journal of Robotics Research*, vol. 22, no. 1, pp. 7–20, 2003.

[8] B. Jung and G. S. Sukhatme, "Tracking targets using multiple robots: The effect of environment occlusion," *Autonomous robots*, vol. 13, no. 3, pp. 191–205, 2002.

[9] K. Krishnanand, P. Amruth, M. Guruprasad, S. V. Bidargaddi, and D. Ghose, "Glowworm-inspired robot swarm for simultaneous taxis towards multiple radiation sources," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, 2006, pp. 958–963.

[10] S. Kamath, E. Meisner, and V. Isler, "Triangulation based multi target tracking with mobile sensor networks," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 3283–3288.

[11] X. Cui, C. T. Hardin, R. K. Ragade, and A. S. Elmaghraby, "A swarm approach for emission sources localization," in *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*. IEEE, 2004, pp. 424–430.

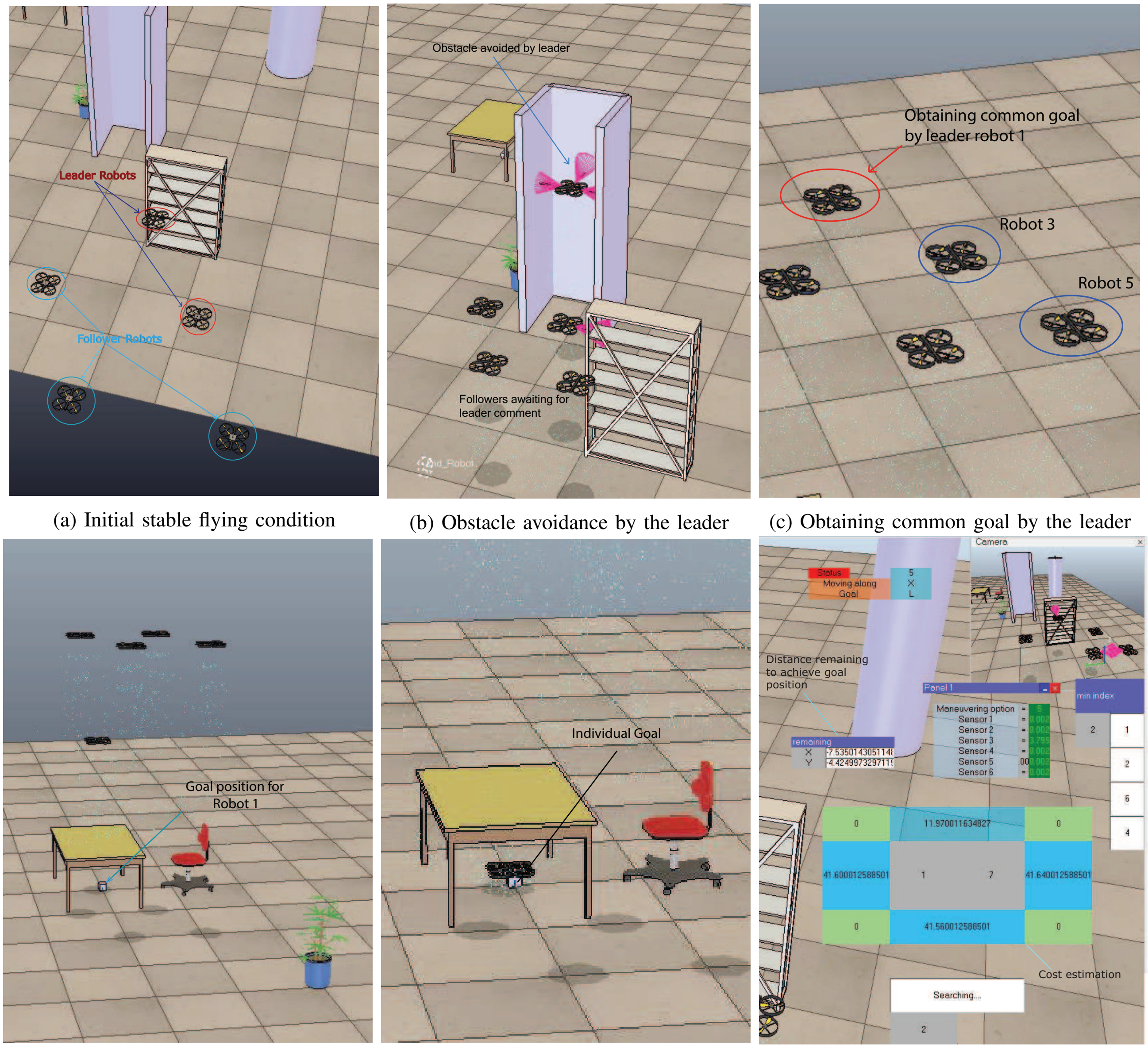
[12] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Physical Review Letters*, vol. 75, no. 6, pp. 1226–1229, 1995.

[13] G. Lee and N. Y. Chong, "A geometric approach to deploying robot swarms," *Annals of Mathematics and Artificial Intelligence*, vol. 52, no. 2-4, pp. 257–280, 2008.

[14] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer, "Pattern formation by autonomous robots without chirality," in *Proc. 8th Int. Colloquium on Structural Information and Communication Complexity*. Citeseer, 2001, pp. 147–162.

[15] Z. Cao, M. Tan, S. Wang, Y. Fan, and B. Zhang, "The optimization research of formation control for multiple mobile robots," in *Intelligent Control and Automation, 2002. Proceedings of the 4th World Congress on*, vol. 2. IEEE, 2002, pp. 1270–1274.

[16] *Data Structure and Algorithms*. World Scintific Publishing Co. Pte. Ltd., 2003.



(a) Initial stable flying condition

(b) Obstacle avoidance by the leader

(c) Obtaining common goal by the leader

(d) Moving towards goal position

(e) Obtaining individual goal position

(f) User interface

Fig. 5: Aerial flocking simulation snapshots

[17] M. Freese, S. Singh, F. Ozaki, and N. Matsuhira, "Virtual robot experimentation platform v-rep: a versatile 3d robot simulator," in *Simulation, Modeling, and Programming for Autonomous Robots*. Springer, 2010, pp. 51–62.