Unmanned Aerial Vehicles (UAVs), commonly known as drones, have recently been applied in choreographed light shows and thus evoked a new boom in technology. In 2016, Intel® used a cluster of its Shooting Star™ drones that were controlled by only one pilot and a single laptop to perform an aerial light show, resulting in a twinkling digital galaxy floating above the skyline.

Yet Intel is not the only company interested in aerial shows using drones since this newly formed technology has been brought under the spotlight and has remained to be a heated topic.

Other related researches and finding also focused on the control and orientation of drones in application.

In response to the Mayor’s ask of an outdoor aerial light show using drones, we carefully investigate the idea and would like to present our mathematical model, conclusions and recommendations for the aerial light show. In brief, we address the problem of optimizing the flight paths for each drone through mapping the locus function with three-dimensional system of coordinates, taking the number of drones required, safety concerns, launch area required and air space required into consideration.

We formulate a optimization model to account for the optimal flight path for each drone in order to form the shape of our three-dimensional design of display——a Ferris wheel, a dragon and a map of China——a mixture of both static and dynamic images. Based on the historical data from all the aerial light shows using drones that have already taken place so far, we determine the initial conditions, prerequisites and several basic parameters of drones for our model. To solve the model, the shortest path and its length of every two nodes in the incomplete undirected graph are calculated with Euclidean distance. In search of the optimal model, we run through the particular path for each drone and group those paths that share similar characteristics to minimize the number of functions controlling the drones using clustering analysis. To ground this model in reality, we animate the image and the whole display process through computer simulation to adjust and update colliding paths. Additionally, we use the Bessel function t0 fit the speed of the drones, making the transition between each image more vivid.

We show that this strategy is not optimal but can be improved by optimizing the coordinates and functions of the flight path for each drone. If the Mayor were to adopt this model and strategy for the aerial light show on the annual festival, the cost would be approximately......We modify the model to reflect the flight paths and generalize the model to other fields including but not limited to the combat drone operation as well as control and orientation system. We conclude with a series of recommendations for how best to design and distribute the particular path for each drone. The simulation examples validate the feasibility of our strategy.

Our suggested solution, which is easy to implement, includes a detailed aerial display program and flight paths for each drone. We firmly believe that our algorithm is broad and flexible enough to accommodate various local conditions, safety concerns and other unexpected incidents. Since our model is based on the control and orientation of UAVs, our strategy may also contribute to other technologies related to drones.

**Key words**: Euclidean distance, Bessel function

Letter to the Mayor

Dear Mayor,

Our team has carefully planned the light show on the night of the annual festival and succeeded in creating three possible sky displays including the pattern of a Ferris wheel, a dragon and a map of China. The whole performance will include 477 drones, and all the people in this city can enjoy this well-organized fantastic light show.

All the drones that will be used in this show should be ready about 20 minutes before the show starts and the place for the light show needs to be spacious, especially to avoid the city center in order to make sure the traffic won't be blocked, and after the careful examination, the square in the north of the city is a suitable place. Our team also suggests that the show should begin at 8:00 pm since it would be completely dark at that time and the audience can fully enjoy this visual feast. Therefore, the drones should be in place at about 7:40, and before that, we want to make sure that all the roads around the place should be cleared so that the potential accident can be avoided. The apron will cover approximately the space of 100m\*70m on the ground and the whole performance will occupy the total space of 200m\*70m\*200m. The total time of the light show will be no longer than 20 minutes according to limited time a drone can constantly fly in the sky.

The brightest shining point of our display is the third image designed. The broad territory of China fully unfolds the prosperity of our motherland and our patriotism, echoing the theme——what ethnic is what worldwide. Another great brilliance that lies thoroughly in our model is how we improve our algorithm to avoid crashes between any two drones during the overall display. Remarkable progress has been made through our adjustment for each drone since the total number of crashed drones is lowered down from 61 to 0. Since our model involves 477 drones, it’s clearly a great challenge to ensure that any two flying tracks do not possess intersections. Yet we manage to conquer the challenge and successfully present a structural model in response to the task.

Our model effectively achieves all of the goals we set initially. It is definitely a feasible solution and could handle large quantities of data. Admittedly, there remain several flaws in our robust and effective model. But we firmly believe that with a larger number of drones, more adjustment of the flight paths, and more factors being taken into consideration, the model can be improved to a higher and more realistic level. In addition, our model generalizes the algorithm used in the control and orientation of UAVs, flexible and broad enough to accommodate various local conditions, safety concerns and other unexpected incidents. We proudly declare that the application of our model maintains a vast potential for future development including combat drone control and orientation system.

Attached on the next page is our designed images for the aerial light show. We express our sincerest gratitude for your trust in us to organize this festive event, which we hope will develop into a worldwide carnival in which everyone enjoys and appreciated our drone light show.

Best Wishes

**Background**

Pilotless aircraft, often referred to as "unmanned aerial vehicle", is abbreviated as "UAV". It is a pilotless aircraft operated by radio remote control equipment, its own program control device, or operated entirely or intermittently by the on-board computers. Since the birth of the aircraft in the early twentieth century, people have proposed the idea of unmanned aircrafts because of the safety problem of the aircrafts. In 1930s, the British Ferrell company remade a double-fixed wing aircraft into an unmanned drone, which was the first time UVA had entered the history of aviation. Since then, UVA has been used in a lot of domains including aerial photography, news report, wildlife protecting and also performances despite the military use. In a recent UVA show performed by YiHang GHOSTDRONE 2.0, engineers designed a set of intelligent and efficient unmanned aerial vehicle remote control system, which realized the function of using only one computer as a ground control station to autonomously control, monitor the flight task of thousands of UAVs, and set the color change of aircrafts’ lights. They presented a large-scale visual feast in the form of fancy lighting show in only 15 minutes.

In our task, our main goal is to organize a beautifully performed light show by using approximately 480 drones and create 3 possible displays. The main challenge is how to minimize the total time the whole performance would take because of the limited time a drone could fly constantly in the sky and how to reduce the total distance that drones would move from one displayed pattern to another.

**Restatement of the problem**

We are asked to organize an outdoor aerial light show with the utilization of drones. The main challenge is depicting three possible images and the overall show process as well as determining the optimal flight paths of each drone device that would simulate our image on display. To be more specific, the required launch area, required number of drones, required air space and the transitioning flight paths between two images for our three images——the Ferris wheel, the dragon and the map of China. The great barrier lies upon us is how to avoid crashes of the flight paths during the display and the transitions between two images. The effect of changes in the total distance, average distance, and time required based on the consideration of safety concerns, limited space and limited flying speed will also have to be investigated to explore the advantages we may achieve in future events.

**Assumptions**

We make the following assumptions about the initial conditions and basic parameters of the drones in this paper. The functions and performance parameters of each drone may have a slight difference, but in order to simplify the model, we assume all of them to be the same. Through the reference of the Shooting Star™ drones used in the Intel® light show, some key parameters of the drones are assumed and set as following:

Table 1 Basic Parameters of the drones

|  |  |
| --- | --- |
| Size | 384mm×384mm×93 mm |
| Propeller diameter | 6 inches (15 cm) |
| Maximum take-off weight | 280 grams |
| Maximum time of flight | 20 minutes |
| Maximum distance of flight | 1.5 km |
| Maximum airspeed of flight | 10 m/s |
| Maximum airspeed of light show mode | 3 m/s |

Each drone can automatically measure its horizontal position and vertical height with the utilization of GPS and barometer. Each drone can determine the target location of the next moment according to the preset track. With the flight control system for navigation, the drones can complete the overall effect of structural formation. The flying path for each drone may vary from the preset track, but we approximated the flight path to be a linear function to simply the model.

Considering the convenience of formation, the take-off site and landing position of each drone is fixed to form a rectangle. To minimize the required launch space, we shape all the drones to remain a regular triangle distance with each other instead of simply fitting them into the points in a rectangle. In order to prevent mutual interference and ensure safety, the distance between each other during the whole display is further than the minimum safety distance by our definition.

Meteorological conditions can be complicated and unpredictable in reality; therefore, we ignore the factors related to meteorological conditions including the wind speed and non-ideal weather. Other unexpected incidents like breakdowns of drones and defaults are assumed to be impossible as well. Additional assumptions are made to simplify analysis for individual sections. These assumptions will be discussed at the appropriate locations.

**Definition符号说明**

1. We define the flight from the take-off apron to the Ferris wheel as the **stage Ⅰ flying process**. Similarly, the flight from the Ferris wheel to the dragon, the flight from the dragon to the map of China, and the flight from the map of China to the landing apron are defined as **stage Ⅱ flying process**, **stage Ⅲ flying process**, and **stage Ⅳ flying process**.
2. To better build our model, we defined the minimum safety distance between the center of any two drones is A meters, and the maximum observing distance between the center of any two drones is B meters. As a result, the distance between any two drones would be

In our case, the drones we use possess a size of 384mm×384mm×93mm and the propeller of each drone possess a diameter of approximate 15cm. Therefore, we plug A=2 to be the minimum safety distance between any two drones by our assumption when taking both the lower and upper boundaries into consideration. In this way, our model is more flexible since the minimum safety distance can be defined to be any number that satisfies different parameters initial settings of drones under other circumstances. The analysis, calculation and computer animation appeared in the rest of our paper are all established and formed under this assumption.

1. In order to simplify our calculation and unify the coordinates in the three displays, we number all the drones as drone number 1-477. To clarify, the number of each drone does not change from one image to another. In this way, we can easily present the transforming track of each drone’s coordinates during the overall 5 stages flying process.

**Model**

1. **Modeling**

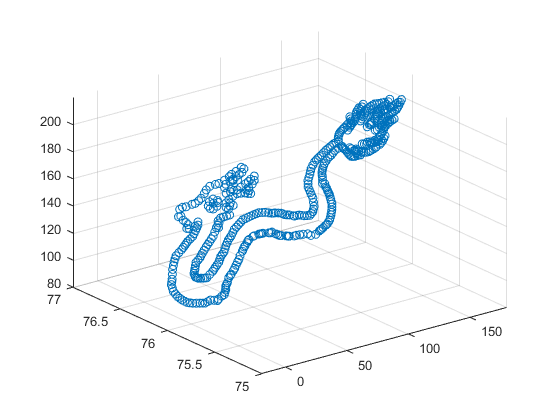
In response to the Mayor’s demand of investigating the idea of organizing an aerial light show, we start off our model by designing the three possible images that we would like to present. The first two images——the Ferris wheel and the dragon——are mandatory. With the help of reference images, we are able to design the stick figures of these two displays. With respect to the third one, we designed a map of China with a Chinese flag occupying the large territory in the west, representing the theme of “what ethnic is what worldwide”.

After establishing the images on display, we set up a left-handed three-dimensional coordinate system with the origin at the lower left corner of the parking apron. To be more specific, x-axis stands for the north direction; y-axis stands for the east direction; z-axis stands for the height. Then we pick out points from the images with the criterion that the coordinate difference between any two adjacent points remains no less than the minimum safety distance. Under this prerequisite, our minimum number of drones required would be 477, as each point selected out from the images represent a drone in reality. To validate the feasibility of the images in our model, we write a test program to examine whether the coordinate difference between every two drones is further than the minimum safety distance. The main theory in this program we write is to detect whether or not there is drone in a circumference with the radius of the minimum safety distance. More specifically speaking, for each drone in the Ferris wheel, if there is a drone detected within the range of the minimum safety distance, it means that the two drones would ultimately crash with each other. The test result for the Ferris wheel from our computer simulation is shown below:



From the diagram, we can naturally reach the conclusion that any two adjacent drones wouldn’t crash with each other. With respect to the other two images——the dragon and the map of China, the crashing test result is shown in the appendix. All the test reports clear the possibility of crashes between drones.

As for the design process of the dragon pattern, we first download a picture of dragon and erase the dragon’s claws

Lastly, we calculated the three-dimensional coordinate of each drone and use the coordinates to form matrixes for later steps in modeling. All the coordinates in form of the three images are attached to the appendix.

1. The stage Ⅰ flying process

The main challenge in this part of our model is how to distribute the flight path for each drone and make sure that there’s no intersection in any two flight paths. Since our design of the Ferris wheel is a tridimensional spatial structure which has 5 layers including the front, the back, the frame and the carriages. The following figure shows the three views of the Ferris wheel.

Front view

side view

top view

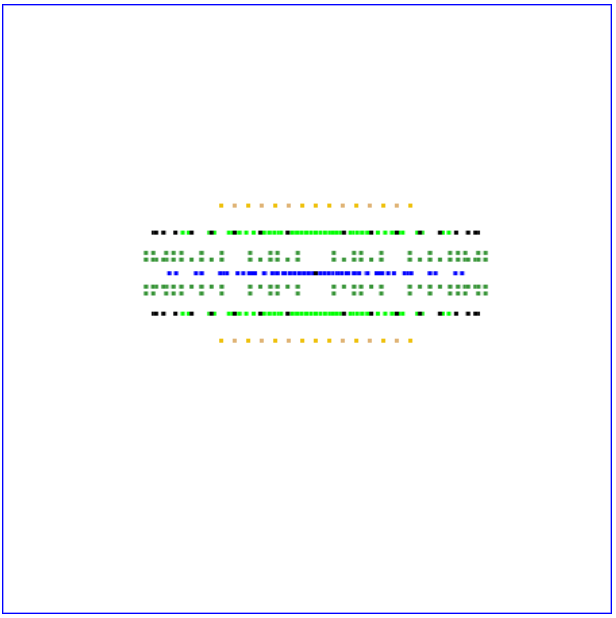
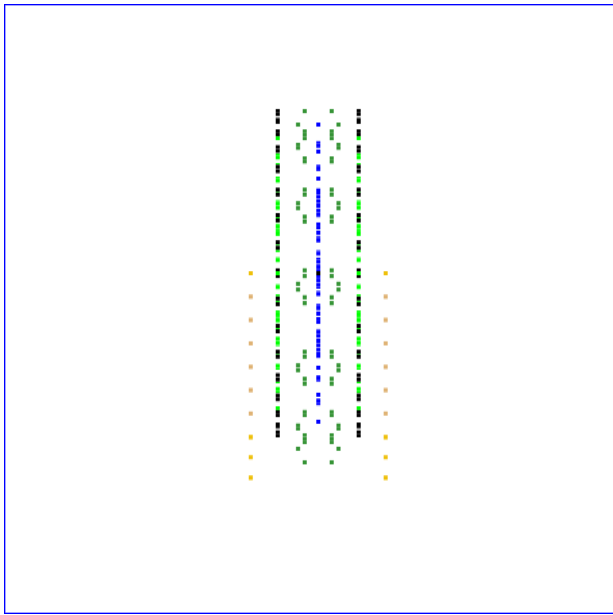
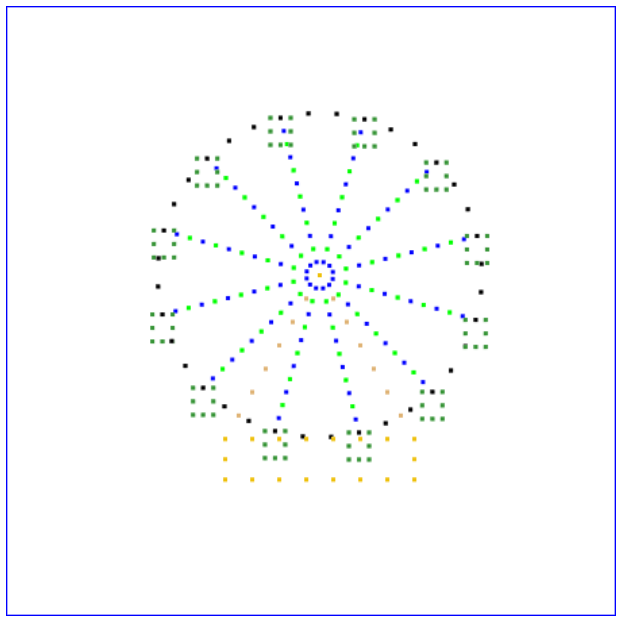


Figure1: the three views of the Ferris wheel

Our plan is first controlling the drones to take off from the apron and raise up in the midair to form a horizontal Ferris wheel, and then flipping it to the upstanding position. In order to accomplish this goal, we divide all the drones in the horizontal Ferris wheel into several flying groups using the clustering analysis. The cluster criterion is the z-axis parameters. In other words, by the standard of height, we group the drones into 5 levels in order to gain a better control. The clustering result is shown in figure2.

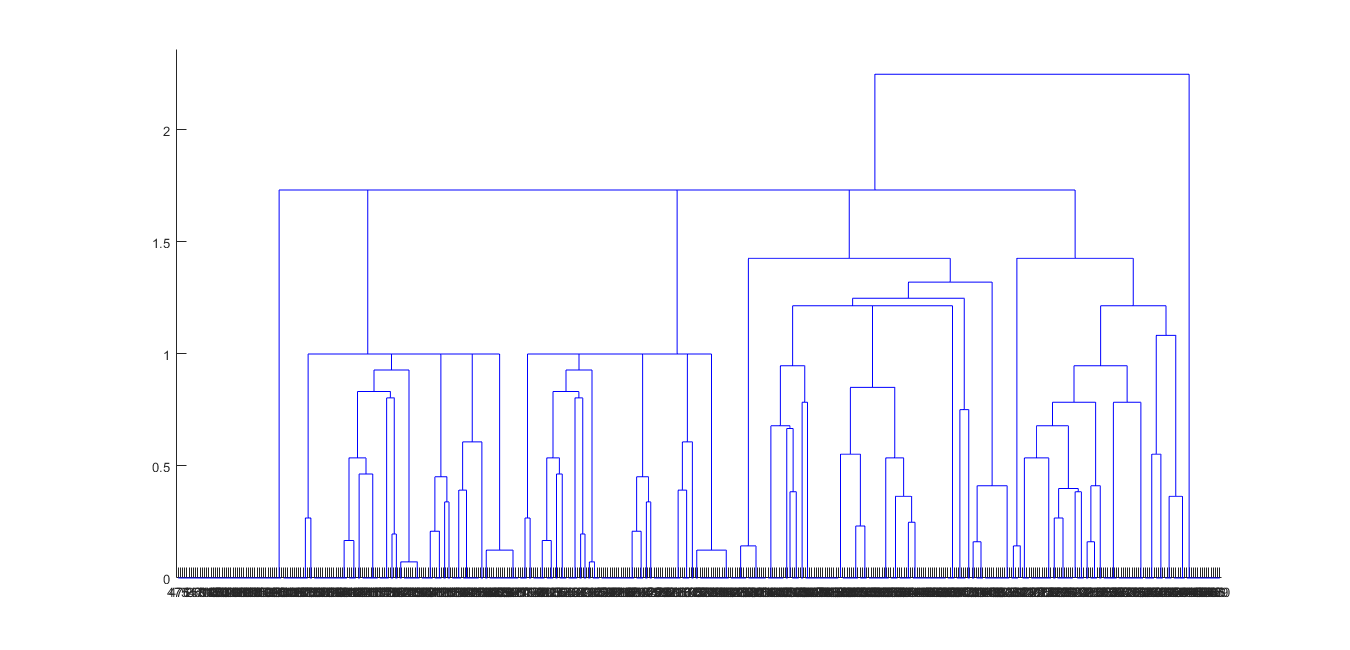


Figure2: the clustering analysis result of the Ferris wheel

Our next step is numbering the clusters from top to bottom as cluster 1-5 and determining the priority of being distributed to a coordinate on the take-off apron for each drone. According to the clustering analysis result, the priority order would be cluster 5, cluster 3, cluster 2, cluster 1, and cluster 4.

First, we match every drone in cluster 5 with a drone on the parking apron that has the least Euclidean distance, which is

Then we match every drone in cluster 3 with a drone from the set of drones remaining on the parking apron that has the least Euclidean distance. Similarly, we repeat the matching procedure for another three times with the order of cluster 2, cluster 1, and cluster 4, and successfully form an overall match from drones forming the Ferris wheel to drones on the apron.

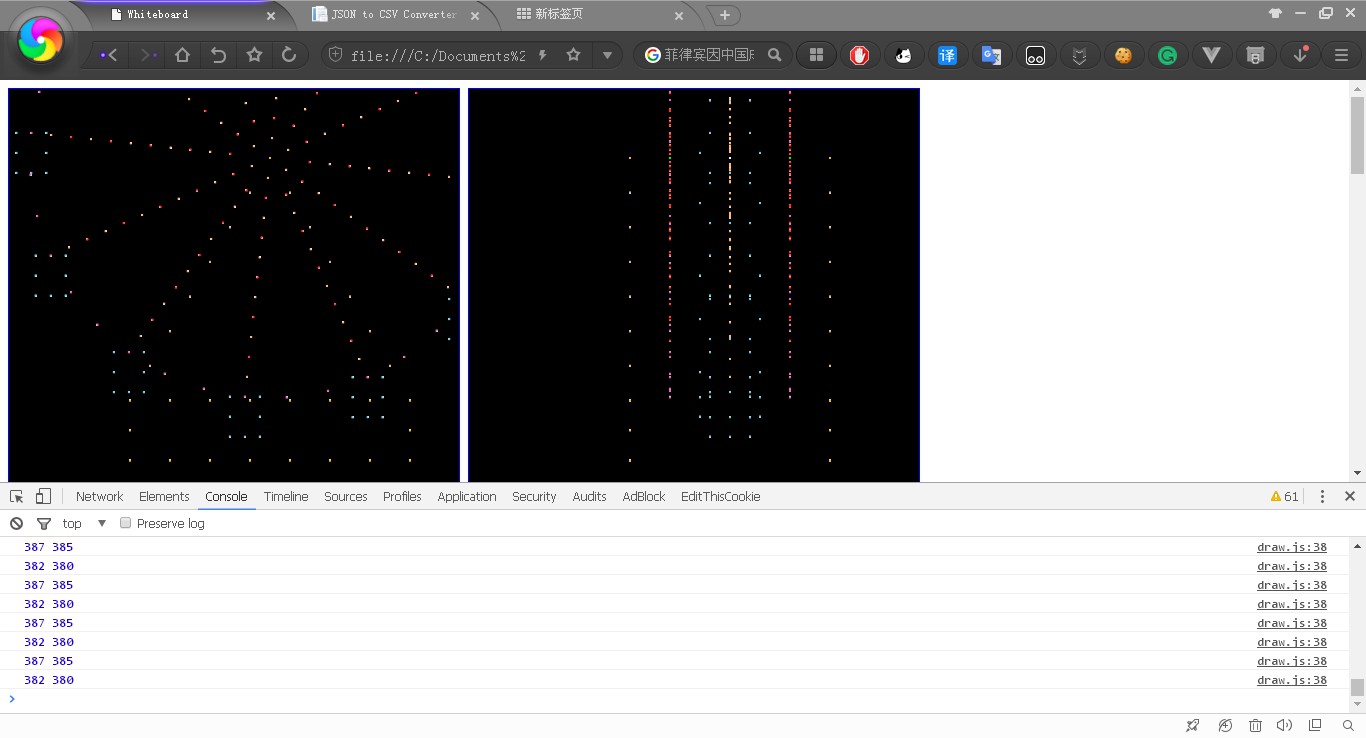
After affirming the initial coordinates and the final coordinates, we then shift our focus to the flight paths that connect the initials with the finals. What we need to deal with now is how to avoid the crashes between any two drones during the whole stage Ⅰ flying process. In the purpose of eliminating the possibility of crashes between any two drones, we adopt the method of forming the Ferris wheel on a horizontal plane with the parameter on the z-axis to be 150 first and then flipping the Ferris wheel around central axis of y=76, z=150 to shape the final image of the Ferris wheel that we desire to display.

As for the first part of the stage I flying process in which the drones fly to the horizontal plane, we connect the initial coordinate on the apron with the final coordinate in the horizontal Ferris wheel using a straight path for each drone. Hence, the general flight path for the drones can be described in a linear function, which is

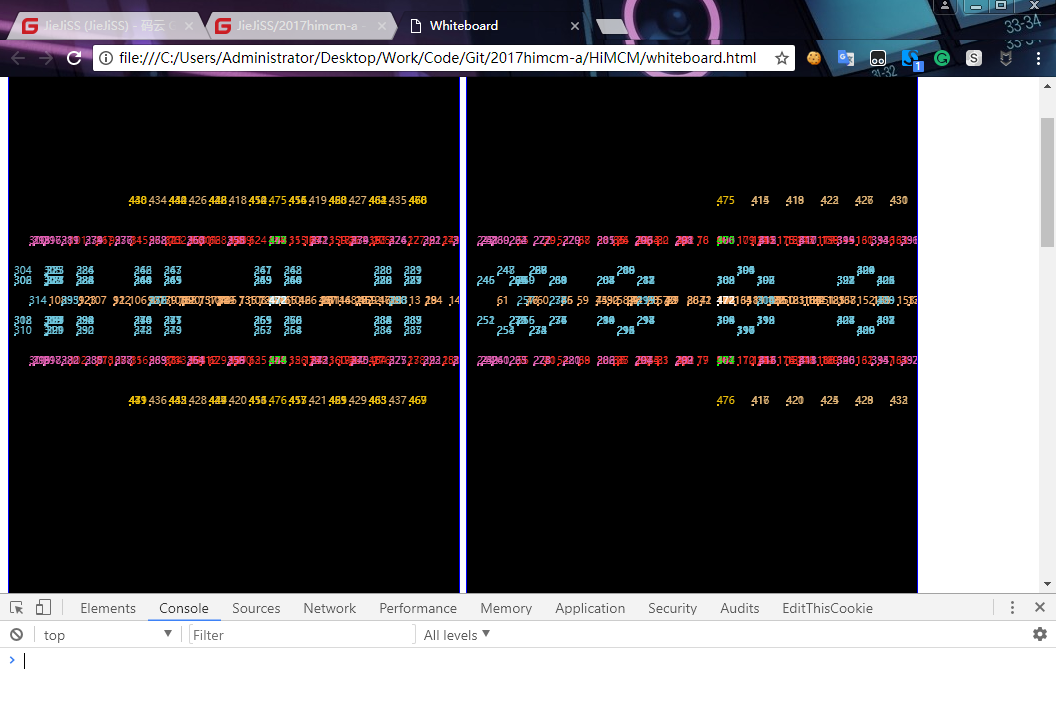
The notation in the function is defined and explained in the following table:

Table3: the definition of notations in the locus function

|  |  |
| --- | --- |
| notation | Definition |
|  | The matrix containing all the drones’ tridimensional coordinates |
|  | The take-off time for the drone |
|  | The actual flying time for the drone |
|  | The hovering time after arriving at the target position for the drone |

The next step after setting a general function for the flight paths of the drones is to validate the feasibility through computer animation. We begin our calculation by running a test program similar to the program mentioned earlier that is used to examine the distance between any two drones in the static initial state of each image to examine the flight locus of each drone under the condition that launch time interval between any two drones is second. The result of the examining test is shown in the following diagram:

From the diagram we can summarize that there are 61 drones in total that may crash with one another during the whole flying process. More specifically, the crashed pair of drones include drone (387, 385), drone (382, 380), etc. However, the crash of planes cannot be tolerated due to safety concerns. To optimize this result, we continue to adjust the launch time and time interval between each other for the crashed drones. With the utilization of this postponing algorithm, we manage to completely eradicate any remaining possibility of crashing and the computer simulation validates the feasibility of our model. The testing report on the crashed drones is shown below:

From the diagram, it can be concluded that none of the drones would crash during the flying process till shaping the horizontal Ferris wheel. Then we flip the horizontal Ferris wheel into the upstanding position. The operative procedure is to make all the drones rotate at the same angular velocity to the upstanding shape. Let be the matrix of all the coordinates in the horizontal level, in which x’, y’, and z’ stands for the matrixes of all the coordinates on x, y, and z axis. Accordingly, the after-flipped matrix of all the coordinates in the vertical level would be

At this moment, our model has accomplished the flight up until forming the Ferris wheel pattern. The stage I flying process is well concluded.

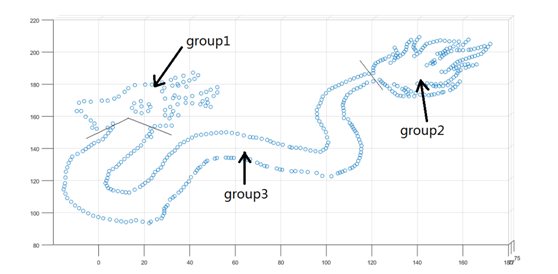
The stage II flying process

The main challenge of this process is how to minimize the total distance of drones flying in the air and making sure the flying pattern is optimal. In order to achieve this goal, we first divide the dragon into three groups according to the shape of the dragon and the distribution of the drones. In the next step, we utilize binary integer programming to find solution that guarantees the total distance to be the shortest in each group. Then we use the testing program to find if there are drones that may crash into one another and adjust the matching method accordingly.

Our team first flip the Ferris wheel around central axis of y=76, z=150 to the previous position mentioned in the former process in order to lay the foundation for the following practice. The dragon mainly consists of three parts including the head part, the body part and the tail part, and our team divides the dragon into three groups in accordance with this pattern. We here define the head part of the dragon as group1, the tail part as group2 and the main body part as group3. Then we calculate the distance between every drone in the Ferris wheel (the one after rotation) and the dragon by using Euclidean distance as mentioned in the former process, which is

In this process, we do not match the drones based on the minimum Euclidean distance. Instead, we utilize binary integer programming in order to achieve our goal. The binary integer programming is a kind of special integer programming that the conclusive variable includes only 0 and 1. It can be used to find the global optimal solution. In other words, the global optimal solution in the process being discussed now is the shortest distance drones fly in total. We will give the specific explanation in the text below.

Our team first use the method of binary integer programming in group1. We number the drones in the Ferris wheel from 1 to 477 and the drones in group1 from 1 to 78. Here we define dij as the distance between the drone numbered i in the Ferris wheel and the drone numbered j in group1. Then we define cij to judge the following situation, if the drone numbered i in the Ferris wheel match the drone numbered j in group1, then we define cij=1. Otherwise, we define cij=0.



The total number of the drones in the group1 is 78. Therefore, according to the definition above, we can define the objective function in group 1 as



Every drone in the Ferris wheel can only match up to one drone in group 1, besides, every drone in group 1 must match one drone in the Ferris wheel. Therefore, the constraint conditions are as the following





Next, we utilize LINGO to run the binary integer programming, the details of the code are in the attachment.

After matching the drones in group1, the total number of drones remain is 399. Then we apply the method of binary integer programming in group2. Similarly, we number the drones remain in the Ferris wheel from 1 to 399 and the drones in group2 from 1 to 167. Accordingly, we define dij as the distance between the drone numbered i remaining in the Ferris wheel and the drone numbered j in group2. Then we define cij to judge the following situation. To be more specific, if the drone numbered i remaining in the Ferris wheel matches the drone numbered j in group2, then we define cij=1. Otherwise, we define cij=0.

The total number of the drones in the group2 is 167. Therefore, according to the definition above, we can define the objective function in group 2 as



Every drone remaining in the Ferris wheel can only match up to one drone in group 2, besides, every drone in group 2 must match one drone in the Ferris wheel. Therefore, the constraint conditions are as the following





Then we again use LINGO to run the binary integer programming, the details of the code are in the attachment.

Then, there are only 232 drones left in group 3. We respectively number the drones in the Ferris wheel and group3 from 1 to 232. As mentioned above, we similarly define dij and cij.

The total number of the drones in the group3 is 232. Therefore, according to the definition above, we can define the objective function in group 3 as

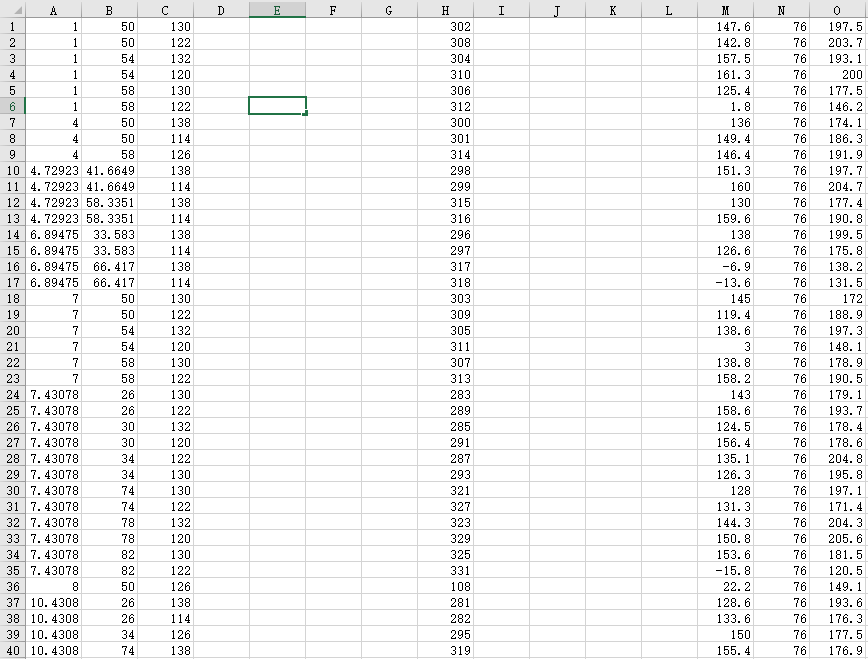


Every drone in group 2 must match one drone remaining in the Ferris wheel. Therefore, the constraint conditions are as the following



Then we again use LINGO to run the binary integer programming, the details of the code are in the attachment.

Now every drone in stage Ⅱ process is matched, part of the matching results are as the following



Column A, B, C respectively represents the x-coordinate, y-coordinate and z-coordinate of drones in the Ferris wheel. Column represents the drone's original number. Column M, N, O respectively represents the x-coordinate, y-coordinate and z-coordinate of drones in the dragon.

The complete results of the matching in the second process are in the attachment.