

# Design and Analysis of Multi-robot grouping aggregation algorithm

Huailin Zhao\*

*School of electrical and Electronic Engineering, Shanghai Institute of technology, No. 100, Haiquan Road, Fengxian District, Shanghai 201418, China*

Zhen Nie

*School of electrical and Electronic Engineering, Shanghai Institute of technology, No. 100, Haiquan Road, Fengxian District, Shanghai 201418, China*

*E-mail: [zhao\\_huailin@yahoo.com](mailto:zhao_huailin@yahoo.com), [hbniezhen@foxmail.com](mailto:hbniezhen@foxmail.com)*

*[www.sit.edu.cn](http://www.sit.edu.cn)*

## Abstract:

In this paper, the grouping aggregation problem of multi-robot in simple environment is studied. The grouping algorithm and aggregation algorithm are mainly discussed. Based on two clustering algorithms, K-means and K-medoids, two grouping algorithms are designed. And the noise immunity and grouping rapidity of these two grouping algorithms are analyzed. Aiming at the problem of aggregation, three kinds of aggregation algorithms are proposed. Taking the center aggregation control as an example, the cooperation and control of each group of robots are studied. Finally, the simulation results of MATLAB shows that the multi-robot grouping and aggregation algorithm is effective.

*Keywords:* multi-robot; clustering; grouping aggregation; swarm intelligence

## 1. Introduction

The multi-robot collaboration system is a frontier topic in the field of artificial intelligence [1]. From the control point of view, the multi-robot collaboration system is a typical distributed control system [2-4], and its goal is to build large complex systems which included software and hardware systems into small, easy-to-manage systems that communicate and coordinate with each other. The research of multi-robot collaboration system involves the knowledge, goals, skills, planning of robots and how to make robots coordinate actions to solve problems, and the idea of swarm intelligence is

emphasized. The multi-robot collaboration system forms a complex system through communication, coordination, and cooperation among autonomous robots. The major applications include formation flying of multi-UAV, clusters of warehousing logistics robot, competitions of robot team, large-scale robotic rescue systems, etc. Some Examples in nature are dancing fish, flying birds, and so on.

Aggregation is the basic problem of multi-robot system consistency [5-6], and it is also an optimizing measure [7], and the grouping aggregation is a special case of aggregation problem. Group aggregation is a motion planning of the robots randomly distributed in space, which are divided into many

groups according to a specific grouping rule, and then each group of robots adjust the motion states through communication and cooperation between each other, finally they gather together, or in the desired area to complete the aggregation. In the end, all the robots formed a multi-group gathering situation, the idea of swarm intelligence is emphasized. After the comparative analysis of two clustering algorithms, K-means and K-medoids, an efficient grouping aggregation algorithm is designed based on K-means algorithm. The feasibility of multi-robot grouping aggregation is verified by MATLAB simulation experiment. And the multi-robot distributed coordination and grouping aggregation control are realized.

## 2. Analysis and design of grouping algorithm

Grouping aggregation is divided into two steps, grouping task and aggregation task. The grouping task is to complete the group division of each robot according to a specific grouping rule. This study designed and analyzed a grouping algorithm based on two clustering algorithms, K-means and K-medoids.

(1) The K-means algorithm is a clustering algorithm which is the simplest and most commonly used in unsupervised learning. It is a center-based clustering algorithm [8-9]. The K-means algorithm completes the grouping task by minimizing the distance between each sample and the center of the cluster. The initial centers of each cluster were randomly selected.

In the Euclid space, a set of n-dimensional samples  $X = \{x_1, x_2, x_3 \dots x_i \dots x_n\}$  (representing n robots) is given, where  $x_i \in R^n$ . The K-means algorithm divides all samples into k clusters  $C = \{c_i, i = 1, 2, \dots, k\}$ . The Euclidean distance is chosen as the criterion for the distance, the equation is expressed as follows.

$$d(a, b) = \|a - b\| \quad (1)$$

where  $a, b$  represents two points in the Euclid space, which are two n-dimensional data.

In determining the similarity of the samples, the samples are divided into the clusters with the highest similarity according to the following equation (the similarity is based on the distance between the samples and the centers, and the smaller the distance, the higher the similarity).

$$c_i = \arg \min_C \sum_{i=1}^k \sum_{x_j \in X} \|x_j - \mu_i\|^2 \quad (2)$$

Where  $\mu_i$  is the center of ith cluster.

The k-means algorithm needs to re-divide the cluster and update the centers iteratively. When the Square Error Sum is converged, the K-means algorithm ends. The equation of Square Fitting Error Sum is given by

$$J = \sum_{i=1}^k \sum_{j=1}^n d_{ij} \|x_j - \mu_i\|^2 \quad (3)$$

$$\text{Where } d_{ij} = \begin{cases} 1, & x_j \in c_i \\ 0, & x_j \notin c_i \end{cases} \circ$$

(2) The K-medoids algorithm uses the point closest to the cluster center as the center of the cluster [10-11]. There are many forms of K-medoids algorithm, the most classic one is the PAM algorithm. The basic idea is:

Firstly, k samples are randomly selected from n samples as k initial centers; then Euclidean distance is calculated according to the equation (1), and the initial clustering is performed according to the principle of minimum distance; Finally, center points are replaced by non-center points repeatedly to improve the clustering quality. The post-replacement clustering quality is estimated by a cost function (4), which measures the average dissimilarity degree between the object and its reference object. In the process of substitution, the least cost is selected for replacement, and cluster again until the cluster no longer changes. The cost function is expressed as follows:

$$TC_{ih} = \sum_{j=1}^n C_{jih} \quad (4)$$

Where  $TC_{ih}$  is the total cost after the center point  $o_i$  replaced by non-central point  $o_h$ ;  $C_{jih}$  is the cost of  $o_j$  after  $o_i$  is replaced by  $o_h$ . If the total cost is negative, it means that the average dissimilarity degree within the clusters becomes smaller after the replacement. So  $o_i$  can be replaced by  $o_h$ . Otherwise, it cannot be.

Under the requirement, to achieve the grouping aggregation, the grouping is the premise of the aggregation, and the quality of the grouping directly affects the aggregation of the robots. Therefore, the two grouping algorithms are further studied, mainly analyzing the noise immunity and the calculation rapidity.

(1) The analysis of the noise immunity

The noise immunity is the first factor to be considered in system design, and it is the key to system security under abnormal conditions. In order to test the noise immunity of the two algorithms, 20 samples were set to simulate 20 robots randomly placed in space, and they were divided into two groups by K-means and K-medoids respectively. Another abnormal sample was added as noise.

As mentioned earlier, the K-means algorithm is a center-based clustering algorithm, so the position of the center is an important criterion for the quality of the grouping. The experimental results are shown in Fig.1. Fig.1(a) shows the results of grouping without noise. The samples were well divided into two groups by the k-means algorithm, and the grouping effect was well. Fig.1(b) shows that when the noise point c appeared, although the grouping task was completed, the center was obviously deviated. If the noise is more intense, the center deviation would be larger. Due to the deviation of the center, the grouping of the two samples, a and b, has changed. If the noise is more intense, the grouping results of more samples would be affected. It can be seen that the K-means algorithm is sensitive to noise. If the sample contains abnormal points, it would seriously affect the grouping result.

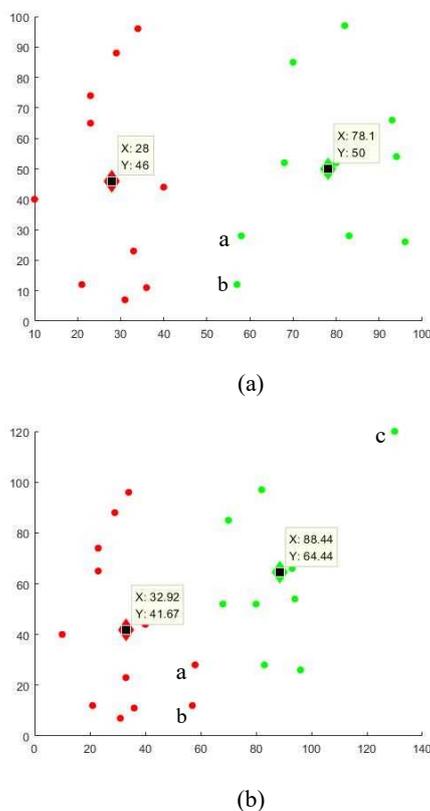


Fig.1. The test of noise immunity of K-means algorithm: (a) the grouping result without noise; (b) the grouping

© The 2019 International Conference on Artificial Life and Robotics (ICAROB2019), Jan. 10-13, B-Con Plaza, Beppu, Oita, Japan

result with noise

The K-medoids algorithm uses the point closest to the cluster center as the center point of the cluster, thus effectively avoiding the phenomenon that the center of the K-means algorithm changes due to the deviation of the noise. The simulation of grouping results is shown in Fig.2. Fig.2(a) shows the grouping result without noise, and the grouping result was good. Fig.2(b) shows the grouping result when the noise point c appeared, and the grouping task was still completed. Because the center was instead of the sample closest to the center. Therefore, there is no center deviation, and the grouping result was not affected. It can be seen that the K-medoids algorithm has strong noise immunity.

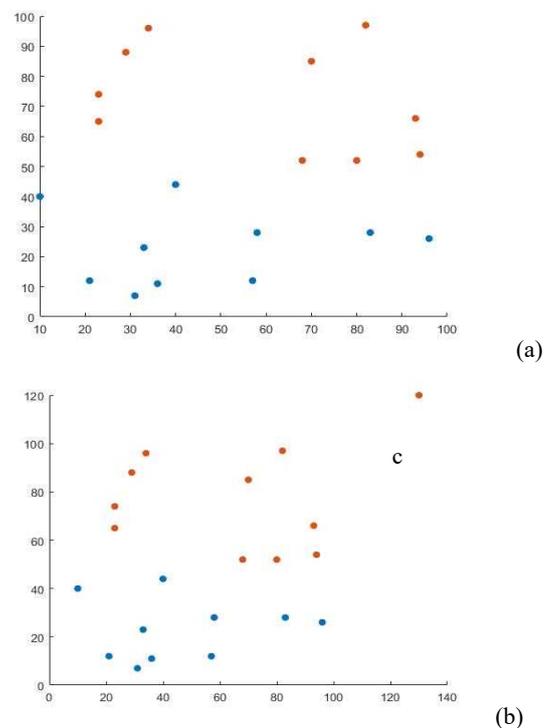


Fig.2. The test of noise immunity of K-medoids algorithm: (a) the grouping result without noise; (b) the grouping result with noise

(2) The analysis of the calculation rapidity

The K-means algorithm a center-based clustering algorithm. In the calculation process, only the average of the robot coordinates should be calculated. The time complexity is  $O(nkt)$  for  $k \ll n$ ,  $t \ll n$ . Where n is the number of samples, k is the number of clusters, and t is the number of iterations. The K-medoids algorithm uses the point closest to the cluster center as the center of the cluster. In the calculation process, the distance and total cost between any two points are constantly calculated. The time complexity is  $O(n^2kt)$  for  $k \ll n$ ,  $t \ll n$ . Where n is

the number of samples,  $k$  is the number of clusters, and  $t$  is the number of iterations. Obviously, the time complexity of the  $k$ -medoids algorithm is greater than  $k$ -means algorithm. When a large number of samples or a large  $k$  should be deal with, the calculation speed will be slow, which seriously affects the grouping efficiency.

Through the comparison and analysis, the  $K$ -means algorithm is simple and fast. Although the noise immunity is not as good as the  $K$ -medoids algorithm, when the sample is large, a few noise has little effect on the  $K$ -means algorithm. The  $K$ -medoids algorithm has strong noise immunity, but when there are a large number of samples, the calculation speed is slow due to the large time complexity. Since this grouping aggregation will process a large amount of data, the  $K$ -means algorithm was decided.

### 3. Analysis and design of aggregation algorithm

As mentioned earlier, the aggregation problem is that multi-robots randomly distributed in the space adjust the motion states through communication and cooperation between each other, finally they gather together, or in the desired area to complete the aggregation. Aiming at the aggregation task requirements of this topic, three aggregation control algorithms were designed and analyzed in this paper:

1) Selecting a robot from each group as leaders, as shown in Fig.3(a). The basic idea of this algorithm is to pre-set a leader robot, and other robots are called followers [12-13]. In the process of aggregation, the leader controls the motion path, and the followers keep a certain angle and distance with the leader, finally all robots aggregate at the position of the leader. After careful analysis, the aggregation has some defects. On the one hand, because the algorithm doesn't make full use of the advantages of distributed systems, when the leader robot goes wrong, it will cause all robots fail to aggregate. On the other hand, because the control of the system is relatively simple, the follower or leader robot may be lost in the process of motion in the complicated reality environment. The aggregation task failed.

2) Selecting the neighboring robot as the reference point, as shown in Fig.3(b). Each robot finds its own target position in the whole aggregation movement through the position information of neighboring robots. The limitation of this method is that a large amount of communications is required, and each robot needs to spend more time for com-

munication, and the result of the formation control is not satisfactory.

3) Selecting the center of group as the reference point, as shown in Fig.3(c). This selection method is called center algorithm. During the aggregation process, multiple robots form a geometric figure. According to the geometric figure, the geometric center is calculated. Each robot uses takes the center as a reference point to adjust the direction and speed, and move to the reference point. During the aggregation movement, the system constantly updates the reference points, and finally all robots reach a same point. the aggregation task is end. This algorithm is simple and practical. The system is convergent. The advantages of the distributed control system are fully reflected.

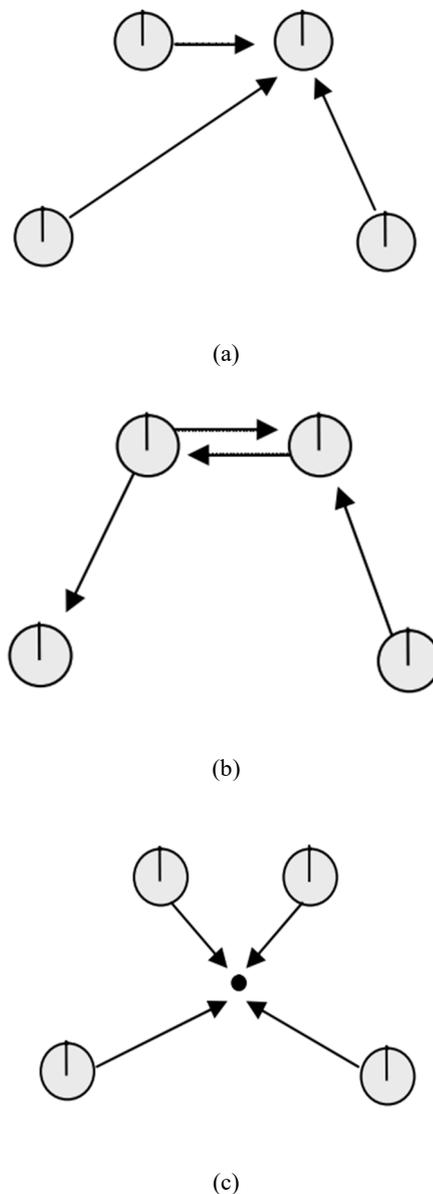


Fig.3. Three different aggregation algorithms:(a) Selecting a robot from each group as leaders; (b) Selecting the

neighboring robot as the reference point; (c) Selecting the center of group as the reference point.

Comparing the three aggregation algorithms, this study intends to use the center algorithm for aggregation control. The center calculation equations are expressed as follows:

$$x_c = \sum_{i=1}^n x_i / n, \quad y_c = \sum_{i=1}^n y_i / n \quad (5)$$

Where  $(x_i, y_i)$  is the 2-D coordinate of the current position for the  $i^{\text{th}}$  robot, and  $(x_c, y_c)$  is the center coordinate.

During the aggregation process, the system periodically updates the center position and notifies all robots. Each robot makes motion planning according to its current position and new center position. The motion planning equation is given by

$$X_i(k+1) = X_i(k) + v_i(k+1) \cdot T \quad (6)$$

Where  $T$  is the sampling period,  $X_i$  is the motion state of  $i^{\text{th}}$  robot,  $v_i$  is the velocity of  $i^{\text{th}}$  robot.

#### 4. Calculation process of grouping aggregation algorithm

As mentioned earlier, to complete the packet aggregation task, the first thing should be done is grouping operation, and the next is the aggregation operation. The specific process is as follows.

According to the grouping requirements, the robot updates the position of the center of the group in real time, and calculates the distance to the centroid, grouped according to the nearest distance principle. The specific algorithm flow is as follows:

- (1) choose  $k$  robots from all robots randomly as the initial group center;
- (2) The Euclidean distance between each robot and the group center are calculated, and assign them to the group with the closest center according to the nearest distance principle;
- (3) Calculate the mean of all objects in each group as the new center, and calculate the Square Error Sum. Repeat from step 2 until the Square Error Sum is converged.

After the grouping operation is completed, the aggregation operation is started. According to the requirements of center algorithm, the center position of each group should be real-time updated and notified each robot in the group. Each robot adjusts the speed and direction according to its current position

and new center position, then moves to the center. The specific algorithm flow is as follows:

- (1) Each robot initializes its position, and reads and saves the current position;
- (2) The robots in the group communicate with each other to inform the position information. The center coordinates are calculated according to the position coordinates of all robots. Then each robot calculates its own speed and direction of motion based on its own coordinate and center coordinate. Finally, the robots move to the center of the group.
- (3) Repeat from step 1 until all robots in each group enter a small range.

#### 5. Simulation analysis

According to the grouping aggregation control algorithm, the research group carried out simulation analysis on the MATLAB. In order to facilitate the observation of the simulation results and analysis, the simulation environment which is a two-dimensional with a length of 100 and a width of 100 was first configured. 60 robots were randomly set up in the grid. It was intended to divide them into 4 groups and achieve group aggregation. The specific two-dimensional grid and the initial position of the robot are shown in Fig.4.

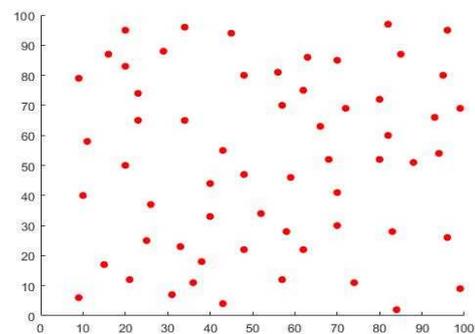
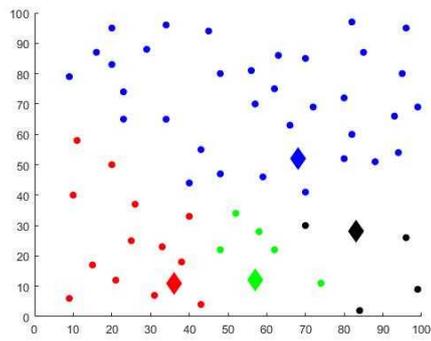


Fig.4. The specific two-dimensional grid and the initial position of the robot

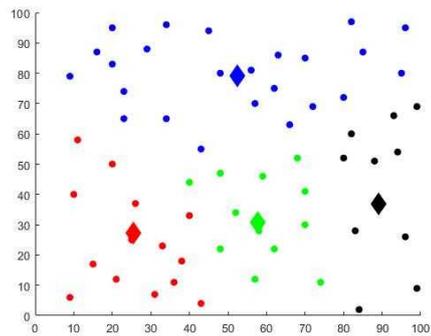
The multi-robot system is first divided into four groups according to the K-means algorithm, and the aggregation motion starts after the grouping is completed.

During the grouping process, the centers are updated after each grouping and grouped again according to the nearest distance principle. Fig. 5(a) shows that the four sample points were randomly selected as the initial centroid, and the first grouping was completed. Fig.5(b) and Fig. 5(c) were the repeated grouping process of the robot, and the diamonds showed the real-time position of the centers. The grouping result was as shown in Fig. 5(d), and there

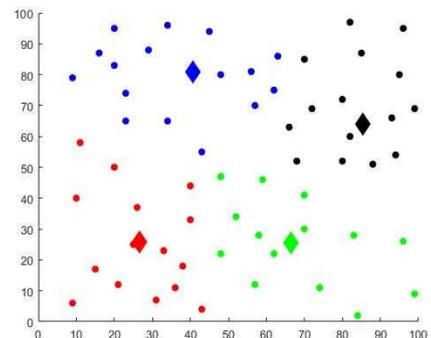
was a good effect



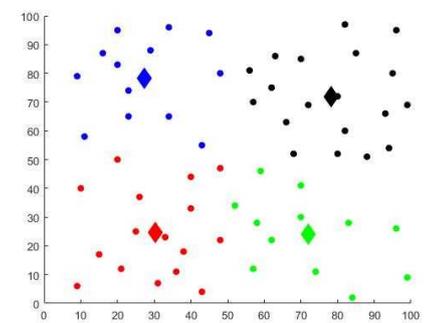
(a)



(b)



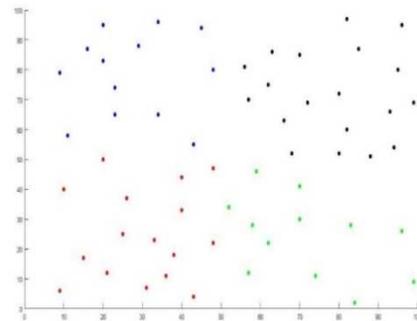
(c)



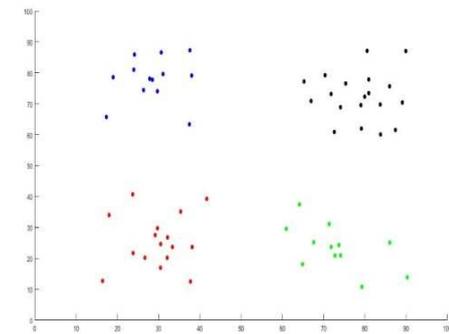
(d)

Fig.5. The simulation of grouping process

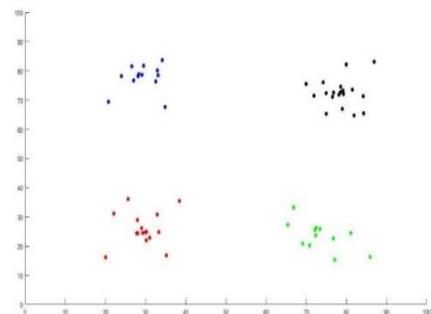
In the process of aggregation control, the robots of each group moved from the initial position to the center of each group. During the aggregation motion, the centers were updated in each sampling period (the sampling period  $T$  was set 1 second). The Fig.6(a) to Fig.6(e) correspond to a sampling period in the aggregation process, respectively. Fig.6(d) shows that the robots have gathered within a small range. Fig.6 (e) shows that the motion state of the robots was slightly adjusted relative to Fig.6(d), but it was still within a small range, so that the aggregation task was completed.



(a)



(b)



(c)

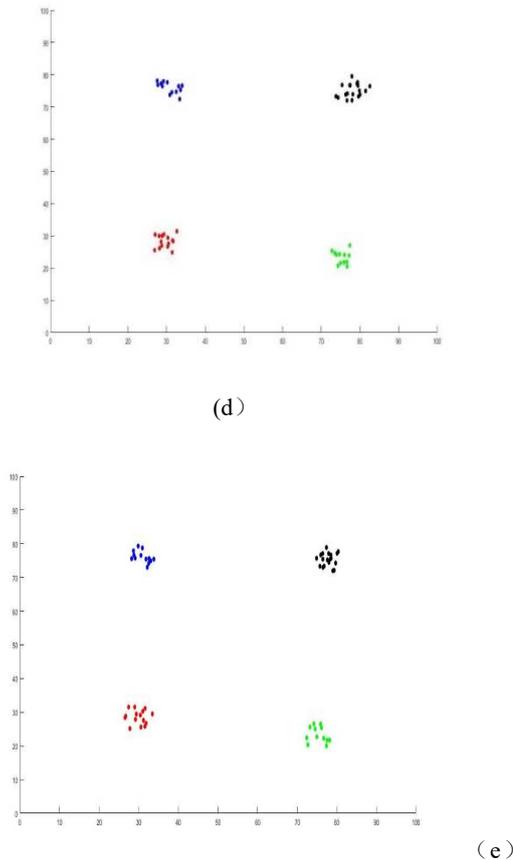


Fig.6. The Simulation of aggregation process

## 6. Conclusion

This topic takes multi-robot system as the research object, and takes grouping aggregation as the main task, and realizes the grouping and aggregation control of multi-robot in simple environment. The research group designed the center grouping algorithm and the center aggregation algorithm based on K-means algorithm. The MATLAB simulation analysis shows the detailed process of grouping operation and aggregation operation, which verifies the feasibility of the grouping aggregation algorithm. The algorithm embodies the swarm intelligence in a certain degree.

## References

1. Huailin Zhao, Yangguang Guo, Masanori Sugisaka. Design of the Multi-Car Collaboration System[C], *the 2017 International Conference on Artificial Life and Robotics (ICAROB 2017)*, 2017.01, Japan. 186-190.
2. Wei Tingting. Research on the surround control problem of multi-agent system [D]. *University of Electronic Science and Technology of China*, 2014.
3. Shi Xiaoxing. *Research on Formation and Coordination of Group Robots* [D]. Lanzhou University of

Technology, 2011.

4. Dimos V. Dimarogonas, Kostas J. Kyriakopoulos. Connectedness. Preserving Distributed Swarm Aggregation for Multiple Kinematic Robots[C]. *IEEE Transactions on Robotics*, VOL. 24, NO. 5, OCTOBER 2008, 1213-1223.
5. Huailin Zhao, Wei Ren. Consensus Problem of Distributed Multi-agent Systems[C], *Proceedings of the 2015 International Conference on Artificial Life and Robotics (ICAROB 2015)*, 2015.01, Japan. 201-206.
6. Huailin Zhao, Zheng Wu, Xiaoxing Wang. A Multi-robot Rescuing System[C]. *The 2018 International Conference on Artificial Life and Robotics (ICAROB 2018)*, 2018.02, Japan. 582-586.
7. Zhu Guoqing. *Study on Control on strategy for Self-Organized Formation of Swarm Robotics Systems*[D]. Shandong University, 2017.
8. Wang Qian, Wang Cheng, Feng Zhenyuan, Ye Jinfeng. Review of K-means clustering algorithm [J]. *Electronic Design Engineering*: 2012, 4:21-24
9. Yang Ping, Guo Chunyang, Li Cuiming. Swarm Robots Aggregation Formation Control Based on K-means Clustering Algorithm[J]. *Modular Machine Tool & Automatic Manufacturing Technology*, 2012, 8:89-92.
10. Wang Zhongqiang, Li Gui, Han Ziyang, Zheng Xinlu. Research and Implementation Based on K-medoids Clustering Algorithm[J]. *Silicon Valley*:2011,1:80-81
11. Xia Ningxia, Su Yidan, Yu Xi. Efficient K-medoids clustering algorithm[J]. *Application Research of Computers*:2010,12:4517-4519
12. Wang Wenming, Zhou Shuai. Research on Multi-Robot Cooperative Control and Formation Method[J]. *Netinfo Security*: 2014, 6:59-66.
13. Luo Jing, Liu Chenglin, Liu Fei. Pilot-following formation and obstacle avoidance control of multi-mobile robots[J]. *CAAI Transactions on Intelligent Systems*, 2017, 12(2): 202-212.