Autonomous perception algorithms for a team of underwater robots : coordination strategy based on onboard cameras

Xiaomin Wang

THESE DE DOCTORAT DE

L'ECOLE NATIONALE SUPERIEURE
DE TECHNIQUES AVANCEES BRETAGNE

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies
de l'Information et de la Communication
Spécialité : *Automatique, Productique, Robotique*

Par
Xiaomin WANG

Autonomous Perception Algorithms for a Team of Underwater Robots : coordination strategy based on onboard cameras

Thèse présentée et soutenue à L'ENSTA Bretagne, Brest, le 10 novembre 2021
Unité de recherche : Lab-STICC CNRS UMR 6285

Rapporteurs avant soutenance :

Kouider-Nacer	M'SIRDI	Professeur des Universités, LiS, Aix-Marseille Université
Lionel	LAPIERRE	Maître de conférences, HDR, LIRMM

Composition du Jury :

Président :	Luc JAULIN	Professeur des Universités, Lab-STICC-ENSTA Bretagne
Examinateurs :	Franck RUFFIER	Directeur de recherche CNRS, ISM, Aix-Marseille Université
	Hélène THOMAS	Enseignant-chercheur, ENSTA Bretagne
Dir. de thèse :	Benoit ZERR	Enseignant-chercheur HDR, Lab-STICC-ENSTA Bretagne

Invité(s)

Benoit CLEMENT	Enseignant-chercheur, HDR, Lab-STICC-ENSTA Bretagne
Zhendong SUN	Professeur, Université des Sciences de Shandong (Chine)

Acknowledgement

I would like to express my sincere gratitude to all teachers, classmates, friends and family who have guided, cared about and helped me.

First and foremost I am extremely grateful to my supervisors, Prof. Benoit Zerr, Dr. Helene Thomas, Prof. Benoit Clement for their invaluable advice, continuous support, and patience during my PhD study. Prof. Benoit Zerr is so learned on the robotics, he always gives me intelligent advices, I am so lucky Prof. Benoit Zerr as my supervisor. Dr. Helene Thomas is so nice and responsible, she always seriously reads my reports and other manuscripts, and gives many advices. Prof. Benoit Clement gave me the chance to ENSTA Bretagne, so that I could have the PhD experience in ENSTA Bretagne.

I would also like to thank Ms. Annick Billon Coat, Dr. Irène Mopin, Dr. Dominique Monnet, Dr. Fabrice Le Bars, Kahina Bensafia, and other members in the Department of Ocean Sensing and Mapping. It is their kind help and support that have made my study and life in the ENSTA Bretagne a wonderful time. Besides, I would also like to thank the full support and patient instrument of Prof. Zexiao Xie and Prof. Ming Li, and thank other members in the Department of Automation and Measurement in the Ocean University of China for their help and accompanying.

Finally, I would like to express my gratitude to my parents, my elder brother and my husband. Without their tremendous understanding and encouragement in the past few years, it would be impossible for me to complete my study.

Table of contents

Introduction 1
1.1 Research background 2
1.1.1 Development of AUV 2
1.1.2 Development of multi-AUV system 7
1.2 Research review of multi-AUV coordination control method 11
1.2.1 Classical formation control methods 12
1.2.2 The formation control methods based on the sensors' perception and topology structure 14
1.2.3 The existing problems 16
1.3 Main work of thesis 16
1.3.1 Research ideas 17
1.3.2 Structure of thesis 21
1.4 Chapter summary 22
2 Formation definition of multi-AUV system 24
2.1 Relevant definitions and notes of graph theory 24
2.2 Analysis of formation structure 27
2.3 Definition of planar pyramid pattern 30
2.4 Chapter summary 33
3 Neighboring information acquisition 34
3.1 Image pre-processing of underwater images 35
3.1.1 Some state of the art 35
3.1.2 Defogging algorithm 38
3.1.3 Application of defogging algorithm in the water 41
3.2 Identification of AUVs and acquisition of neighbors' local information 42
3.2.1 Detection of AUVs 42
3.2.2 Estimation of neighboring AUVs' local information 44
3.2.3 Identification of ID and information exchange 49
3.3 Chapter summary 52
4. Local position-based control method 53
4.1 Establishment of the common frame 53
4.2 Collision avoidance strategy 54
4.2.1 Determination and optimization of pyramid distribution 55
4.2.2 Non-intercrossing straight trajectories planning 59
4.3 Simulation of the formation building 65
4.3.1 Pyramid building in case of explicit IDs 66
4.3.2 The pyramid building in case of implicit IDs 76
4.4 Chapter summary 78
5 Asynchronous discrete consensus-based formation control with local information 80
5.1 Markov chain and transition matrix 80
5.2 A brief description of consensus algorithms 82
5.3 The trajectory update based on the consensus algorithm with the displacement-based method 83

5.3.1 Calculation of the errors of relative positions $\mathbf{e}_j[k_i]$ 84

5.3.2 Calculation of the weights of neighbors $\mathbf{w}_i[k_i]$ 85

5.3.3 Formation control model based on displacement-based control method 88

5.3.4 Simulation results with the displacement-based control method 89

5.4 The trajectory update based on the consensus algorithm with the distance-based method 95

5.4.1 Calculation of the distance vector errors 96

5.4.2 Calculation of the weights $\mathbf{w}_i[k_i]$ 97

5.4.3 Formation control model based on distance-based control method 99

5.4.4 Simulation results with the distance-based control method 100

5.5 Chapter summary 104

6 Experiments and analysis of the formation building 106

6.1 Presentation of the experimental platform of humanoid robot 106

6.2 Acquirement of the neighbor information for NAOs 107

6.2.1 Recognition of NAOs 108

6.2.2 Extraction of neighboring information 112

6.3 The performance test of NAO 117

6.3.1 Recognition of NAOs at "side" postures 117

6.3.2 Evaluation of the visual compass 118

6.3.3 The movement control of NAO 120

6.4 Pyramid building with NAOs 123

6.4.1 Pyramid building with the local position-based method 123

6.4.2 Pyramid building with the asynchronous limited communication distributed control method 125

6.5 Chapter summary 130

7 Conclusion and prospects 132

7.1 The main conclusions 132

7.2 The prospects 134

Reference 136

Introduction

Oceans account for more than 70% of the surface area of earth, cover more than 90% human activity space of world, and contain extremely rich biological and mineral resources. Not only that, the oceans also play an important role in the global environment, such as pollution and carbon cycle.

With the increase of population and development of economic, humans consume more and more natural resources, and the resources on land will not meet the needs of humans. For survival and development, marine development is imperative. For marine development, it is first necessary to explore and know the oceans, and then to exploit marine resources, to complete resource transportation and to avoid or reduce the adverse effects of marine natural disasters on humans.

To this end, we need to develop the ocean remote sensing and monitoring technologies to obtain large-scale and accurate marine environment data and marine biological information, to develop the seabed exploration and analysis technologies to obtain the distribution of resources, to develop underwater platform construction and mining technologies to complete the resource exploitation and transportation, to develop the seabed supervision technology to avoid the damage of transport pipelines and marine pollution, to develop marine navigation and submarine search and rescue technologies to achieve the recovery of wrecked airplane and vessels, to develop port security monitoring technologies to reduce the losses from marine disasters.

Nowadays, with the development of marine technologies, many fields have made great progresses, such as marine transportation, ocean energy, exploration of seabed resources, ocean mapping, seabed mining, marine environment monitoring, biological sampling, port security monitoring, and submarine rescue etc [1]. While letting us continue to know deeply the oceans, marine technology poses certain threats to marine ecology and safety, for instance oil spills [2], marine garbage [3], and nuclear leaks [4].

In fact, our knowledge about the oceans is just a little, more than 80% of the sea has not been explored [5]. For example, there are some lakes containing many unknown creatures, as well as many mountains and valleys on the seabed at a depth of more than 100m. Moreover, compared with the surface of earth with a 100m resolution, only 10%-15% of the detected submarine areas have a resolution of 100m, and most areas have a resolution of 5km. This means that we lost many

submarine features.

In addition, deep sea exploration is very challenging. Human cannot face the pressure in the deep sea, the threat of unknown and dangerous creatures in the deep sea, the drastic frustration of temperature, and the limited view field. Therefore, sophisticated equipment and a large number of financial support are necessary to explore the oceans.

At present, the deep sea exploration mainly depends on the underwater vehicles. AUV has become one of the core equipment for deep sea exploration because it can carry power, achieve self-control and accomplish scheduled tasks automatically. This chapter will focus on the following aspects: the development of AUV and related technologies, the development of multi-AUV systems and related technologies, the research status of formation control, and the research ideas and article layout of this thesis.

1.1 Research background

1.1.1 Development of AUV

AUV system belongs to a multi-disciplinary subject of fluid mechanics, underwater acoustics, optics, communications, navigation, automatic control, computer science, sensor technology, bionics and other fields. The main shape of AUV is Torpedo (see Fig. 1-1). AUV has attracted the attentions of researchers in the whole world because of its important potential application value in civil and military fields. In-depth research work on AUV has been carried out, and it has become the crucial equipment to complete various underwater tasks.

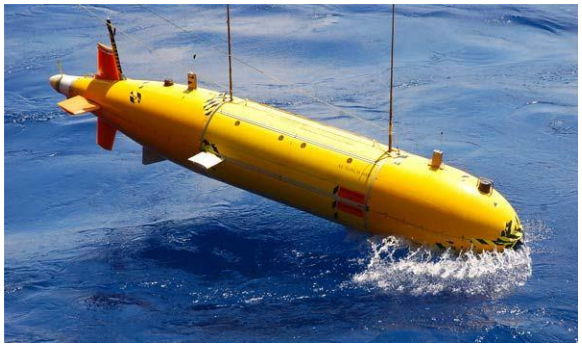


Fig. 1-1 Torpedo AUV (AUV Abyss)

The development of AUV can date from 1960s for specific and explicit applications, such as data collection[6]. During the decade years from 1970 to 1980, it had reached a high tide for the

potential exploration of AUVs, and a lot of AUV test-beds had been developed. For example, the SPURV (shown in Fig. 1-2) developed by Applied Physics Laboratory (APL) of University of Washington, it had cooperated with UARS in the air to collect experimental data in the arctic region. Meanwhile, institute of marine technology, Russian Academy of Sciences (IMTP, RAS) had started its AUV projects to develop the first generation deep-diving AUV (L1&L2) and aerial SKAT.

Fig.1-2 SPURV

In 1980s, the Proof of Concept (POC) of AUV was proposed. The development of low-power micro-computer, memory and software system made the execution of complex navigation and control algorithms be possible, and also identified the technical problems to be solved.

In 1990s, the fundamental researches about relevant technologies were expanded, and the development of AUVs became more international. During this period, AUV had evolved from the POC to the first generation of operable systems that can accomplish tasks. For a wide variety of tasks, many institutions around the world developed AUVs. In this decade years, the new paradigm of AUV was confirmed, and it also promoted the technology becoming more commercial, such as the Autonomous Ocean Sampling Network (AOSN) [7] shown in Fig. 1-3.

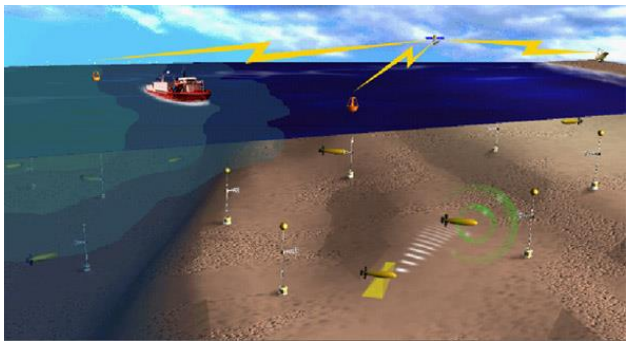


Fig. 1-3 Autonomous Ocean Sampling Network（AOSN）

From 2000 to now, AUVs have been applied to commercial business, and the AUV market supply is all on the rise. For example, the AUVs have been applied to the deep-sea oil and natural

gas exploitation in south Africa and south America, to the search and rescue of underwater wreckage, to the marine scientific research etc. Bluefin 21 shown in Fig. 1-4 was used to the signal exploration of Malaysia Airline MH370 on the south India ocean seabed. The scientific research data collected by ABE (Fig. 1-5) [8] and other marine vehicles also had positive influences to the marine science.

Fig. 1-4 Bluefin 21

The development of AUV includes six fields: autonomy, energy supply, navigation, sensors, underwater communication and movement control.

(1) Autonomy of AUVs means that an intelligent system is provided to the AUVs to sense and evaluate the surroundings, and to make appropriate adjustments for the complete of tasks. But actually, the intelligent level is not satisfying until now, the AUVs accomplish missions usually by a serial of programmed instructions.

(2) Energy supply of AUVs represents the endurance capability. At present, AUVs have the capability to sail in the water for 10 hours. Some systems can survive for several days or even years. Nevertheless, the enhanced endurance is at the expense of the sensors' performance and transmission speed. In the early stage, most of the AUVs used high-cost lead-acid and silver-zinc batteries. Some AUVs adopt lithium batteries, such as ABE submarine vehicles (see Fig. 1-5). The recently developed Ni-MH batteries provide a new kind of energy storage mode for AUVs. In addition, the solar energy can also be used to power AUVs, such as AUSI of USA shown in Fig. 1-6.

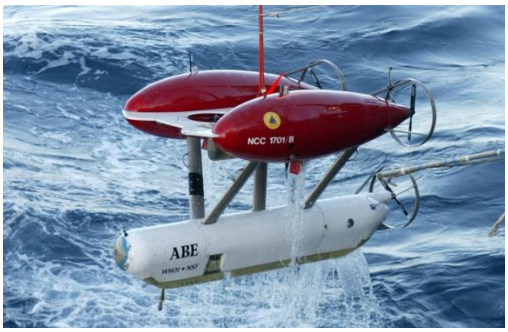


Fig. 1-5 ABE

Fig. 1-6 AUSI

(3) Navigation of AUV mainly refers to the self-localization in the water. Different from the navigation on land or in the air, the application of precise GPS navigation is largely limited because that the electromagnetic signals decay rapidly in the underwater environment. Now there are three kinds of underwater navigation systems: inertial navigation, acoustic navigation and geophysical navigation [9]. The three kinds of navigation systems have their own merits and shortcomings. They can work alone or jointly form a more robust navigation system. In addition, some relatively new communication and navigation systems are developed, such as underwater sensor networks [10] (see Fig. 1-7) etc.

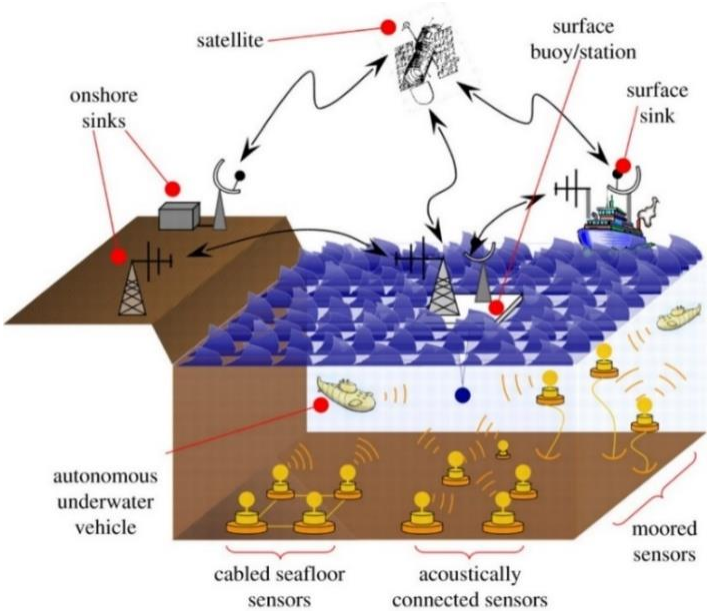


Fig.1-7 Underwater sensor networks

(4) Sensors of AUVs: AUV is mainly a platform equipped with sensors and sensor systems. After AUV can navigate steadily underwater, sensors are equipped to AUVs to acquire effective information from the ocean. However, some existing sensors are not designed for AUVs, and there are some limitations in their uses. Consequently, the more intelligent lower-power high-reliable and small-size sensors for AUVs are developed.

In addition, a lot of work has promoted the development of high-resolution image systems (optical and underwater acoustic images), and it is possible to obtain high-resolution images over long distances with the development of new processors [11]. Sonar is the main sensor to acquire information in the water. It can be used to navigation, communication and detection of objects, as shown in Fig. 1-8. Compared with sonar, the underwater camera has higher resolution and can acquire color images. As a result, although the working range of camera is small, it still can be used

to collect information, search objects and monitor submarine pipe lines in a small range. The 3D topographic map, shown in Fig. 1-9, is rebuilt by one underwater binocular camera system.

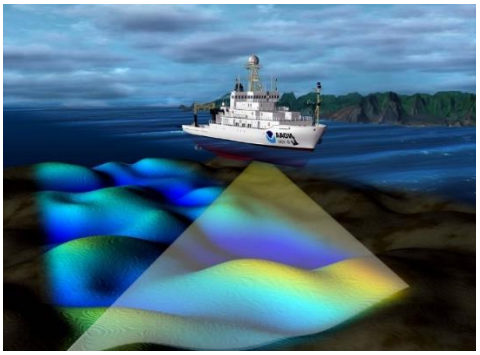


Fig.1-8 The application of Sonar underwater

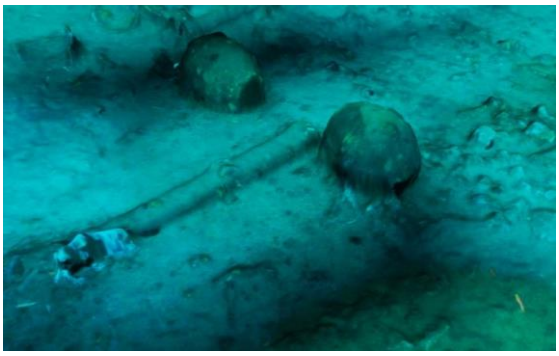


Fig.1-9 3D topographic map rebuilt by underwater binocular camera system

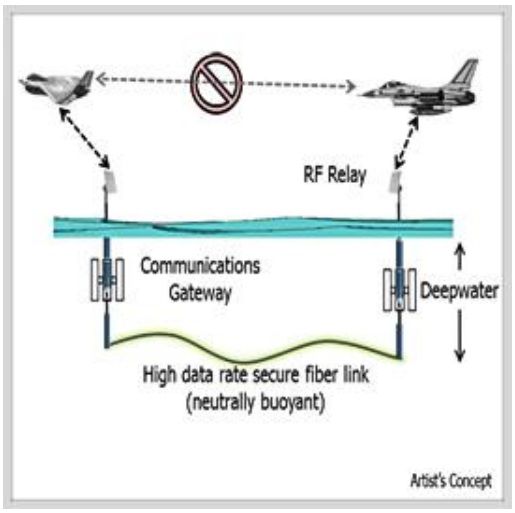
(5) The communication in the water mainly depends on the acoustic communication with the longest communication distance about 20km [13]. But its bandwidth is narrow, about 0b/s-20kb/s, and the acoustic communication is easily influenced by noises, turbidity, salinity and pressure gradient and so forth [14,15]. Besides, the acoustic communication has a certain impact on the marine organisms [16]. In some researches, other communication technologies are estimated. For instance, the light communication [17,21] is low cost and short working range but with ultra high bandwidth, gigabit/s, but it is easily influenced by the turbidity, particles and marine fouling; RF-EM communication (Radio frequency-electromagnetic communication) is suitable in the shallow water, it is not easily influenced by the noise, turbidity, salinity and pressure gradient [14,15], but its working range is small and it is easily disturbed by EMI (electromagnetic interference).

Another communication way is by the network communication facilities, such as the underwater sensor network shown in Fig. 1-7, and the developing underwater satellite system and so on (such as the TUNA project of USA, see Fig. 1-10). The underwater satellite system consists of a transmitter network connected by an optical fiber cable, which can float at the depth of several hundred meters.

(6) The research on the kinetical control of AUVs started early, many control methods have been developed, which includes classical PID control, robust control, adaptive control, fuzzy control,

neural control and hybrid control etc. Now the robust control represented by H_∞ control, neural control and genetic evolutionary algorithm are popular.

(a)



(b)

Fig. 1-10 The TUNA project of USA (a) Temporary data network consisting of a series of buoys and cables. (b) Schematic diagram of working principle of TUNA

Though the breakthroughs have been made in related science and technology, it still has great development space. These limits of technologies also limit the performance and applications of AUV systems. For example, to get a high coverage in the water or on the seabed, a single AUV should be equipped with high navigation sensors, large volume battery and high-performance payloads and so on. For all that, the reliability of a single AUV system is still not satisfying. While the fault occurs, the AUV system has to cancel the tasks. The proposed concept of multi-AUV provides new scheme for further exploring the ocean. The multi-AUV system has higher tolerance to the low cost and low accurate sensors. And even one or more AUV units go out of action, the other AUVs can still finish the tasks.

1.1.2 Development of multi-AUV system

Multi-AUV systems had been considered in the 1980s, but the missions were failed at that time. Because little relevant fundamental technical work had been studied. However, the researchers realized gradually the potential capacity of multi-AUV system in the past thirty years.

The real study to *N* submarine vehicles can be dated back to the Autonomous Ocean Sampling Network (AOSN) supported by American Navy in 1993 [24]. The main reason is that compared with a single submarine vehicle, *N* submarine vehicles have better space resolution and higher work efficiency with low time cost. The tasks with large coverage are more suitable for the *N* submarine vehicles, such as: ocean environment surveillance, ocean biological collection, oil pipe surveillance and mine countermeasures.

Until now, the projects of multi-AUV systems (or multi-submarine vehicles) mainly include: AOSN (Fig. 1-4)[24], AOSN-II (Fig. 1-11)[25], ocean observatories initiative (OOI) (Fig. 1-12) [26], generic oceanographic array technology system (GOATS) (Fig. 1-13) [27], coordination and control of cooperating heterogeneous unmanned systems in uncertain environments (GREX) (Fig. 1-14) [28], cooperative cognitive control for autonomous underwater vehicles/CO3-AUV (Fig. 1-15) [29], and advanced real-time earth monitoring network in the area (ARENA) (Fig. 1-16) [30] etc.

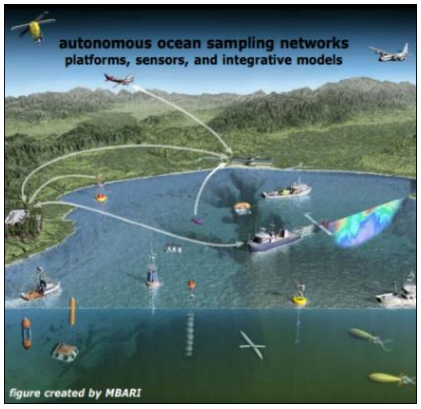


Fig. 1-11 AOSN-II

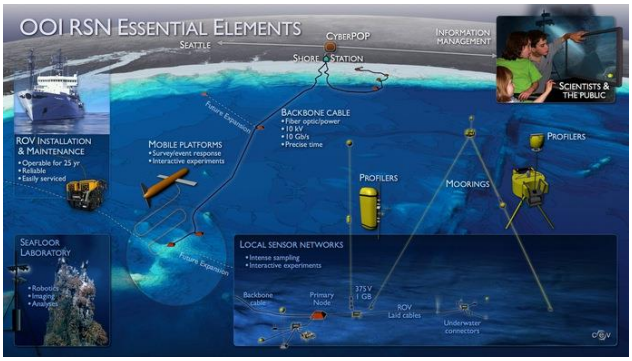


Fig. 1-12 OOI

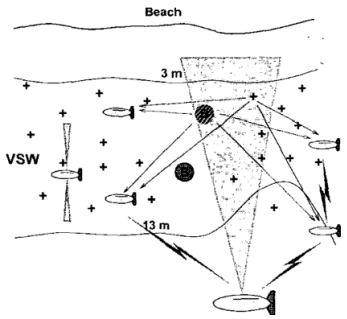


Fig. 1-13 GOAT

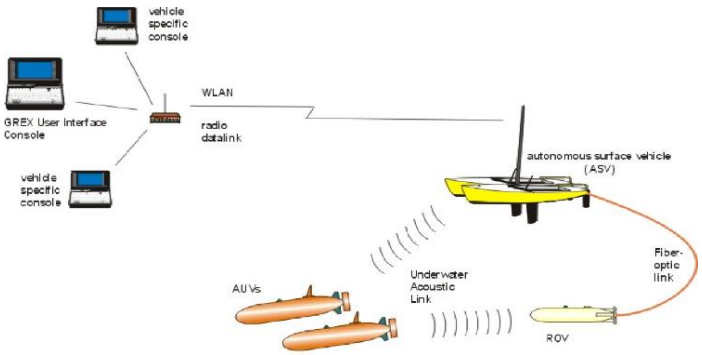


Fig. 1-14 GREX

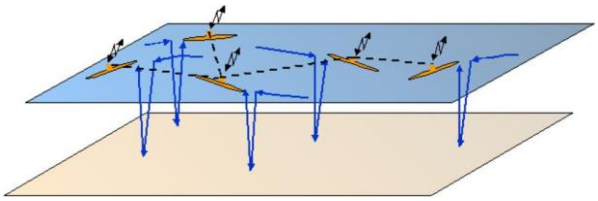


Fig. 1-15 CO3-AUV

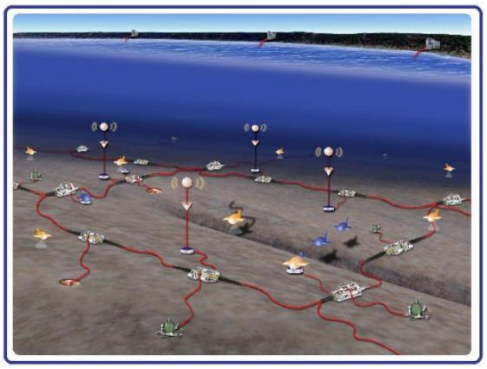


Fig. 1-16 ARENA

The technologies of multi-AUV systems include navigation, trajectory planning, coordination control and acoustic communication etc.

(1) Navigation of multi-AUV system: now the communication among AUV units mainly depends on Sonar, and the navigation systems of different structured multi-AUV systems are different [31]. For instance, each AUV unit of the homogeneous structure systems usually is equipped with high accurate navigation devices. For some of the leader-follower structured systems, the leader is generally equipped with high-qualified navigation devices, while the devices of the followers are low accurate. However, the acoustic communication has time-delay, composed by the underwater acoustic detection delay and communication delay. The detection delay is usually caused by the detection devices, but the communication delay is due to the complex communication process, bandwidth and lower conversion rate.

Though the working range of optical sensors is short, combining with the multi-AUV system, the optical sensors can make up to a visual net with extended working range. And the optical sensors can sense the surroundings, get the relative positions between neighbors. In addition, optical communication has been studied (see section 1.2.2) and been tested the application in the water. As a result, with the multi-submarine vehicles, optical sensors can be used to get the local information as well as to assist the acoustic communication to reduce the communication difficulty.

(2) Trajectory planning of multi-AUV system: trajectory planning plays an important role in the multi-AUV system, which will choose and identify an appropriate path for each AUV to arrive at one position in a working space, even in the unknown environment. The planned path should make the AUVs avoid collision with the obstacles or other vehicles. Two problems should be solved in this part: 1) sensing the environment and building a model; 2) planning trajectory for each vehicles considering the energy cost and collision avoidance. In the known environment, heuristic

algorithms are used widely, such as A*, Dijkstra, D*[34]. In the unknown environment, environment should be perceived and corresponding model should be established, the widely used methods are Karlman filter [35], distributed H_∞ filter [36], and distributed consensus filter [37].

(3) Coordinated control of multi-AUV system: coordinated control of multi-AUV system is to effectively connect all units together to a shape (pattern formation), which provides a good platform for AUV units' interaction and coordination with each other to share information in time. Besides, with the distribution of the multi-AUV system, the coordinated control method can optimize the task decomposition and assignment, solve the resource conflict, and enhance the working efficiency.

(4) Communication of multi-AUV system: similar to the single AUV systems, the multi-AUV systems mainly depend on the acoustic communication. Meanwhile, the optical communication (such as: laser-based communication, LiFi), RF communication are also being studied.

Though multi-AUV systems have many advantages, its development is still limited by two factors: cost and reliability. To be competitive, the cost of each AUV unit should be much lower than that of traditional sophisticated single AUV. The main ways to cut the cost are to reduce the size of AUV unit, use low-accurate navigation sensors and reduce the performance of uploads. With the development of micro-AUV technology (such as the μAUV[2] developed by university Bremen [38] in Fig. 1-17 with the size $393\times188\times200\ \mathrm{mm}^3$ and the weight 2010g), the cost of multi-AUV system is very competitive. In multi-AUV system, optical sensor or low-resolution sonar can replace the high-resolution synthetic aperture sonar. In addition, the reliability of multi-AUV system has been improved with the increasing underwater tasks.

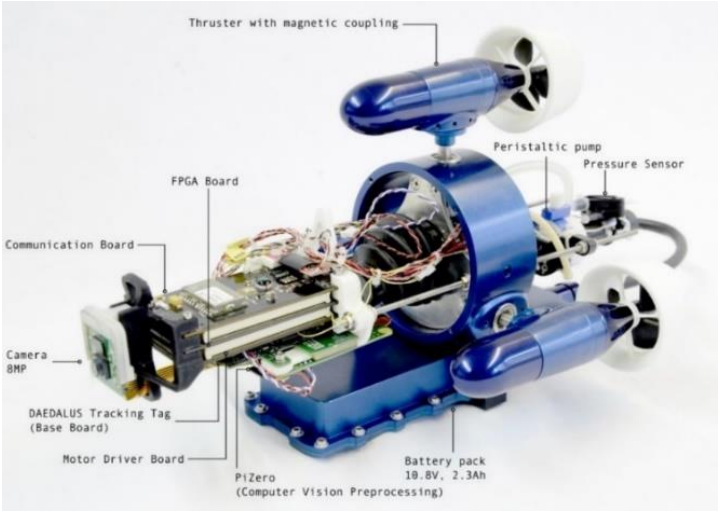


Fig. 1-17 An example of small AUV, μAUV2

1.2 Research review of multi-AUV coordination control method

Multi-AUV system needs the coordination control among AUV units to interact with each other to accomplish the tasks. In the early studies, multi-robot systems on land and in the air get more attentions. In recent 10 years, more and more researchers put their attentions on the coordination control of multi-AUV systems. Different from the environment on land or in the air, though it exists communication limits and complex location, many classical patterns and pattern control strategies can be improved and used in the multi-AUV systems. Then the current status of multi-robot systems' patterns and pattern control methods are given below.

To achieve the tasks, the AUV units should constitute a pattern and cooperate with each other by corresponding control methods until they finish the tasks.

The formation patterns of multi-AUV systems mainly have two kinds of structures: homogeneous structure and hetero-structure. In the systems with hetero-structure, the types of AUV units are different, shown in Fig. 1-18, which are mainly used in cooperating navigation, scanning the seabed and mapping. The AUV units in the homogenous structured systems are all the same, and share the same algorithms (see Fig. 1-19). This kind of system is mainly used to the ocean surveillance and ocean biological sample collection etc.

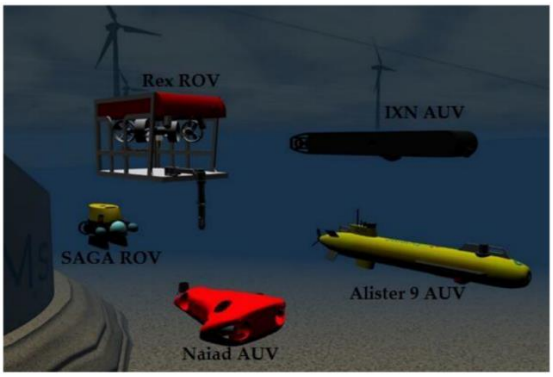


Fig. 1-18 Multi-AUV system with hetero-structure

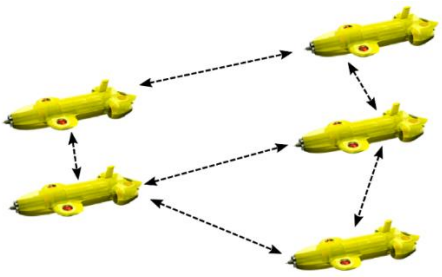
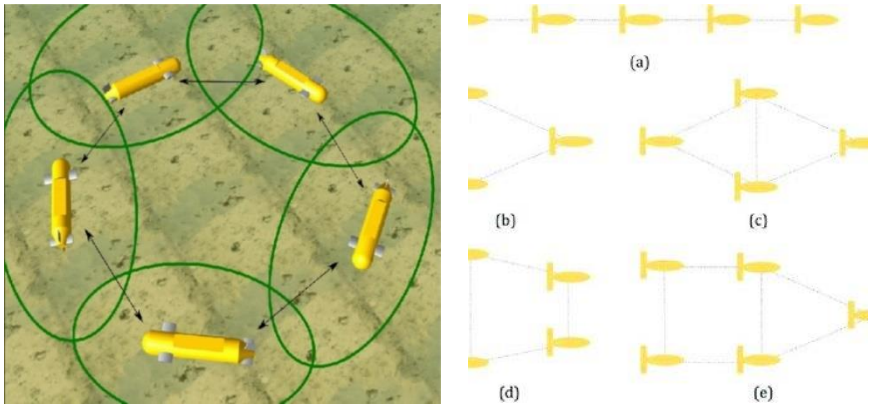


Fig. 1-19 homogeneous structured Multi-AUV system

To accomplish tasks in specific environments (with obstacles, in small space, with currents or limited communication), several geometry patterns are given, which mainly includes two kinds: circular pattern and linked pattern, two examples are shown in Fig. 1-20. The circular pattern includes circle, eclipse and other quasi-circular patterns: for example, literature [39] adopts the circle or eclipse patterns to collect the ocean samples. In [40], a kind of adaptive quasi-circular

pattern based on gene regulatory network can be generated automatically to arrest the target objects. The linked pattern mainly include line, triangle, diamond, polygon etc. Literature [41], [42] present several predefined patterns, especially, the multi-robot system in [42] can evaluate and select appropriate alternative formations automatically according to the situation and environment. The aerial multi-robot systems in [43] and [44] use triangle, multi-triangle or "V" patterns. According to [45], a virtual triangle structure is used to coordinate the system to move forward and sample data. Similarly, in [46], a triangle pattern is built based on the relative positions between AUVs, which is used in the deep sea exploration.



(a) circular pattern (b) linked pattern

Fig. 1-20 Geometry pattern of multi-AUV system

1.2.1 Classical formation control methods

1. According to the communication modes, the formation control can be separated into two classes: Sensing-based control mode and communication-based control mode.

(1) Sensing-based control methods depend on the local information of neighbors got from the sensors, without explicit communication, which has obvious advantages in the limited communication underwater environment [47,48]. However, in the environment where obstacles are covered largely, the sensing range of robots will be declined significantly, which leads to the interruption of network and temporary control failure.

(2) Communication-based control methods can achieve the coordination control among robots in the area with obstacles covered largely, which can compensate the default of sensing-based control methods. The communication-based control methods usually adopt broadcast communication [49].

These two kinds of control methods can combine with each other to enhance the adaptability in the complex environment [50,51].

2. According to the basic ideas of control strategies, the formation control can be separated into four classes:

(1) Leader-follower methods [52,53,54]: In this method, at least one robot is regarded as leader, the others are seen as followers. Leader moves according to the given trajectory, while the followers move and correct their postures by sensing the leader's position online. This method achieves the formation control usually based on the relative orientations, relative positions, or relative distances. The merit of this method is easily to be understood and used, nevertheless the robustness and anti-interference of the system is not good.

(2) Behavior-based methods [55,56]: In this method, the formation task is divided into a serial of robot behaviors. The basic behaviors include: gathering, obstacle avoidance, collision avoidance. Each robot determines its next behavior by sensing the surroundings and its neighbors' information, such as changing its speed, its directions etc. The merit of this method is the possession of effective feedback and collision/obstacle avoidance, and its shortcoming is the lack of explicit definition of each kind of behavior, which results in the difficulty of stability analysis by using mathematical ways.

(3) Virtual-structure methods [57,58]: The idea of this method is to design an ideal structure as a reference, called as virtual structure (VS). This method will control this VS to move according to given trajectory, while the robots will adjust their speeds to tend to the positions in the VS. The merit of this method is that it can maintain the formation well, but the defect is the strict formation will limit its application.

(4) Artificial potential field methods [59,60]: This method is proposed by Khatib [61]. The basic idea is that the robots movements are similar to the electric field generated by positive and negative charges, the destinations and robots are viewed as the inter-attractive charges, while the obstacles and robots are regarded as the inter-repelled charges. This method is mainly used in 2D plane scene. The merit of this method is that it is suitable to the scene with obstacles, but its weaknesses are that the potential field function is difficult to be constructed sometimes, and it may lead to local-minimum.

3. According to whether the formation is clear or not, the methods are divided into two

categories: morphous and amorphous formations [62]. The fixed form formation is identified by the given relative positions, distributions or distances; while non-fixed formation is defined by behaviors, for example, gathering and collision avoidance is used to achieve the formation control. Usually, the non-fixed formation is relevant to the behavior-based method.

4. According to whether the formation given is time-variant or not, the formation control methods are separated into two categories: formation producing problems and formation tracking problems [63]. The purpose of the robots in the former method is to construct a given pattern, which has been realized by the matrix theory, Lyapunov methods, graph theory and receding horizon control method. The robots in the latter method should track the given trajectories, which also has been solved by matrix theory, potential functions, and Lyapunov methods.

1.2.2 The formation control methods based on the sensors' perception and topology structure

According to the sensing capacity of sensors and topology structures of formations, the formation control methods categorize into three classes: position-, displacement- and distance-based methods [62].

(1) Position-based method

Position-based method is very popular [39-44], and a classical position-based method needs each robot meeting two conditions:

① Sensors capacity: All the robots should be distributed in a global frame, the sensors should be sense their absolute coordinates in the global frame.

② Topology structure: Formation is expressed by the absolute positions. As a result, the topology structure is not necessary, because each robot can arrive at its destination by controlling its global position. But the internal contacts among robots can be regarded as the position constraints to enhance the control performance or to reach to other control target, such as the formation maintainance.

(2) Displacement-based methods

Displacement-based methods achieve the formation control according to the relative positions and global orientations [64-67], similar to the fish schooling [68-70]. The orientation-based method

[71] can be seen as one kind of displacement-based method.

Traditional displacement-based methods need each robot meeting two conditions:

① Sensors capacity: Each robot needs a local frame, whose orientation is parallel to the global frame, but without knowing the origin point of global frame. According to the local frame, each robot can perceive the relative positions between neighbors in the global frame, which means that the methods needs topology structure.

② Topology structure: The expected formation is explicit to each robot. To achieve the formation building, each robot will adjust the distribution with its neighbors continuously. Thus, its topology structure should be clear, and the formation control should be achieved only according to neighbors distribution information. This implies that the topology should include at least one spanning tree. When the topology is time-variant, the structure should be strongly connected.

(3) Distance-based methods

Compared with the displacement-based methods, the distance-based methods accomplish the formation building only depending on the local distance information got by each robot in its own local frame. Though it is a big challenge to achieve formation control by distance-based method, relevant researches are increasing gradually [82-75].

The conditions to control formation by using distance-based methods are:

① Sensors capacity: Each robot should build a local frame, and different from the displacement-based method, local frames are mutually unrelated. Without sensing the global orientation, the distance-based methods realize the formation control according to the distance information in each local frame. It means that this method has low demand to the sensors properties and it needs topology structure.

② Topology structure: The expected formation is built depending on the expected distances among robots. Consequently, this structure is defined beforehand. To achieve the formation control, the controlled targets are the relative distances among robots. The distance-based methods are nonlinear methods, even if the model of each robot is linear.

Therefore, the position-based methods are suitable to the general outdoor environment, such as with the assistance of GPS to get the global coordinates. But in other environment, such as indoor or underwater environment, displacement-based and distance-based methods are more appropriate.

1.2.3 The existing problems

(1) Identification of formation structure. Until now, most of the linked or circular structures are designed for the specific tasks with specific number of AUVs. In the dynamic environment, it is always the pursuit of researchers to define a pattern formation for multi-robot system with independence on the number of robots, high robustness of disturbance, large coverage.

(2) Information acquisition and transformation in the water. The main way of communication in the water is acoustic communication with long working range, but also with inevitable time-delay and failure. Influenced by the noises, the accuracy of acoustic communication sometimes is not satisfying. As a result, on one hand, we need to work hard to improve the performance of acoustic communication; on the other hand, we can adopt optical sensors to assistantly sense the surroundings and get information to reduce the volume and frame of acoustic communication. Especially, though the working range of a single optical sensor is short because of light attenuation, the visual network composed by several optical sensors associated with the multi-AUV system has an extended working range. All of these need to be studied further.

(3) The formation building in the water. The formations can be built based on the consensus algorithms which have been developed well on land and in the air. However, most of the studies on land or in the air focus on the continuous consensus algorithms with time-variant (or time-unvaried) topologies; and some of them pay attention to the time-delay consensus algorithms situation considering the communication time-delay [76-81]. Now there are increasing studies on the discrete consensus algorithm [82,83]. Nevertheless, because of the limited communication in the water, the existing consensus algorithms are not effective. In particular, to reduce the frame and distance of acoustic communication, the formation building based on the local information got by optical sensors needs further research.

1.3 Main work of thesis

With the introduction of the first two subsections, underwater vehicles play crucial roles in the ocean exploration and development. But one single AUV faces to a lot of problems during executing tasks, multi-AUV system can conquer some defects of single AUV and get more attention. Then the coordination control becomes one research hotspot. The relevant technologies and algorithms

cannot be used in the water directly because of the failure of GPS positioning and RF communication. Now, the main communication in the water is realized by acoustic sensors. Sonars undertake great roles in the sensing and mapping seabed and the navigation in long distances. However, for the tasks with high accuracy or color requirements, optical sensors have more advantages than sonars. Further, the underwater optical communication as a new kind of communication starts to get attentions in recent years. Underwater camera, as a kind of optical sensor, can not only sense the ambient environment, but also assist to communication for information exchange. Though a single camera has short working range, the visual network constituted by several cameras combined with multi-AUV systems, has an extended working range.

This thesis derives from the application of multi-AUV system, in which, multi-small-AUV system is used to search small objects, improves the seabed mapping precision, and samples the colorful biological information. Considering the cost, the endurance, the resolution, and the collection of colorful information, this thesis mainly depends on the low-cost, high-resolution underwater color camera to sense information.

Ordered formation structure is the basis of multi-AUV systems' coordination control. For multi-small-AUV system, the relative distances among them are not large, which results in the possibility to use camera to get information. Thus, the purpose of this thesis is to achieve the formation building for low-cost small-size multi-AUV systems in unknown underwater environment based on the local information got by underwater color camera. Until now, the studies on the multi-AUV system coordination control based on vision system are few, consequently, this thesis is meaningful for the development of small-size multi-AUV system and underwater optical perception. In this section, we will state the research ideas, key technologies and innovative points of the formation building based on the local information got by cameras.

1.3.1 Research ideas

1. Formation definition

The working range of small-size multi-AUV system is not large, but the number of AUV units can be many considering the low cost of an AUV unit. In order to enhance the success ratio of missions, the reliability of systems should be further improved. Consequently, the formation

structure of the system should meet several demands:

(1) The formation should be independent on the number of AUV units. In other words, the formation should adapt to all the system with N ($N \geq 3$) units.

(2) The formation should be not limited by the missions. For most of the tasks, the formation can be valid.

(3) The formation should be robust, not be destroyed easily by the disturbances, such as the current, obstacles etc.

2. Image processing of underwater digital images

The main sensor of information acquirement is underwater color camera. But due to the fast attenuation, the images got are not clear. To improve the quality of images, image pre-processing is necessary. Then the recognition of AUV and identification of IDs are also significant. In addition, each AUV is only equipped with one single camera, it also should be solved to extract the neighbors' local positions from single image. As a result, the research ideas are given below:

(1) Image deblurring

Image deblurring is done by introducing the defogging algorithm. The reason is that the image deblurring generally are done by adding polarization filter, or compensating the light scattering or reconstructing the color, but the color distortion caused by wave-length attenuation is still existing. In this thesis, the introduced defogging algorithm can correct the wave-length attenuated, even can identify the relative distances according to the atomization degree, which can be studied further in the future.

(2) Recognition of AUV and identification of ID

In this thesis, the AUV (CISCREA) with ROV structure and 4 freedom degrees (3 translation freedom degree and one rotation freedom degree around Z axis) is used (see Fig. 1-21). CISCREA is one kind of robot of ENSTA Bretagne, Lab CNRS 6285, whose main parameters are shown in table 1-1 [84].

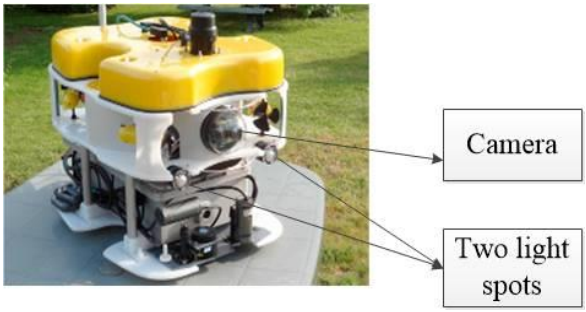


Fig. 1-21 CISCREA

Table 1-1 The main parameters of CISCRE

Physical dimension	0.525m(L), 0.406m(W), 0.395m(H)
Weight(on land)	15.56kg(without load and floating body)
Movement freedom degree	Surge, Sway, Heave, Yaw
Propeller	2 vertical propellers and 4 parallel vector combination thruster
Depth	50m
Battery	2-4h voyage
Maximum speed	2knots (Surge)，1knot (Sway, Heave)

The yellow region of CISCREA is very clear with regular shape, consequently, the CISCREA is recognized according to the yellow region.

The ID identification of AUV is done by two ways. In the process of formation building, the ID can be explicit or implicit. The explicit ID is a kind of logo on the top of CISCREA, while the implicit ID is written in the program. As a result, the explicit ID can be identified by image processing, while the implicit ID will be distinguished by algorithms.

(3) The calculation of AUV's local information

To get the relative positions between AUVs from one single image, a relative distance model and a relative angle model should be established according to the camera model. The calibration in the real environment is a difficult problem. The qualities of images got with different water qualities are different, which results in different calibration results. To simplify the process, a 1:1 CISREA model is modeled in a 3D simulation environment which is close to the real environment. And the models calibrations are done in the simulation environment.

With the calibration results and the positions and sizes of AUVs extracted from images, the local information of AUVs' neighbors can be calculated by the relative models.

3. Coordination strategy

The coordination strategy is composed by a predefined pattern and corresponding formation control. To improve the efficiency and successful rate of formation building, the formation control includes two parts: (1) identifying and optimizing the distribution of predefined pattern to make it tend to the initial distribution and shorten the sum of movement distances of multi-AUV system; (2) planning trajectory for each AUV unit considering the obstacle and collision avoidances.

When planning trajectories, to save time and improve the efficiency of formation building, the environments are separated into two categories: obstacle-free environment and general environment (perhaps with obstacles). Considering the communication limitation in the water, limited communication coordination strategy is proposed.

(1) Obstacle-free environment

In the obstacle-free environment, a local position-based formation control method inspired by the global position-based method and relevant graph theories is proposed. The key idea is that each AUV can construct the common frame independently to identify the initial and predefined formation distributions with the exchanged information; then a collision avoidance strategy is given to plan the non-intercrossing straight trajectories. Because of the same algorithm, the planned trajectories by each AUV will be as same as each other. With the non-intercrossing trajectories, each AUV can build the formation without updating trajectories and can save time. The details are shown below:

① The local positions between AUVs can be got by cameras: with the shared neighbors' information shared by the only once communication, the common frame can be constructed in each AUV according to the same algorithm and the initial distribution of multi-AUV system can be obtained.

② The distribution of predefined formation is determined according to the initial distribution and the collision avoidance strategy, and it will be made to tend to the initial distribution.

③ The initial and predefined pattern distributions are divided into combined sub-groups by collision avoidance. Then the non-intercrossing trajectories will be planned in each combined sub-group.

(2) General environment

In the general environment, the movement behaviors of AUV units along their trajectories may be interrupted in any time because of the emergence of unexpected obstacles in the unknown

environment. Consequently, the local position-based control method without the update of trajectories is not applicable any more. As the formation building can be seen as a consensus problem, the consensus algorithm is introduced to plan and update the trajectories only depending on the neighbors' local information at current moment. Each AUV is dynamic (being in the state of trajectory updating or moving along its updated trajectory), consequently the corresponding topology is time-variant digraph (directed graph). And the communication is not necessary with the identified expected position relationship among AUVs, because each AUV can update its trajectory independently according to the local information got by its on-board camera. The AUVs build the pattern by repeating the loop: "search neighbors, compute local information and update trajectory, move along trajectory", which implies that topology is an asynchronous discrete time-variant digraph.

After studying a great number of consensus algorithms with continuous (or discrete) time-invariant (or time-variant) weighted (or average) undirected (or directed) graphs, the asynchronous formation control method with discrete time-variant digraph is proposed in the limited communication condition based on the asynchronous consensus algorithm with continuous time-variant digraph in [127],[128].

In addition, the confidence of each neighbor (the weight of edges in the topology) can be computed by the limited value of transition matrix of first-order non-homogeneous Markov chain of this topology.

Therefore, the local position-based control method is proposed to build pattern by planning non-updated trajectories due to the obstacle free environment. The asynchronous distributed formation control based on local information is proposed to carry out the formation building due to the general environment. And all the local information used in both methods is obtained by the underwater camera or compass.

1.3.2 Structure of thesis

The whole thesis is separated into seven sections:

Section 1 introduces the development of AUVs and the relevant technologies, the development of multi-AUV systems and the relevant technologies. In this section, the conventional formations

and formation control methods are presented. Besides, the research ideas, key problems and thesis organization are also given.

In section 2, the relevant graph theories about multi-AUV formations are listed. The planar pyramid pattern (called pyramid pattern for short in the latter parts of the thesis) is defined by analysis.

Section 3 describes some image processing methods, which include image deblurring methods, the recognition algorithm of AUV, identification methods of ID, and the constructions of relative distance model and relative angle model.

The local position-based control method is illustrated in section 4. The collision avoidance strategy is proposed to optimize the distribution of planar pyramid pattern according to its initial distribution, and to match up the positions in the pyramid and in initial distributions for getting the expected position relationship among AUVs. Then the non-intercrossing straight trajectories are planned well.

The asynchronous distributed formation control method is explained in section 5. Markov chain and transition matrix are described. Consensus algorithm is given in detail, the trajectory updating models of displacement-based and distance-based asynchronous consensus algorithms are established respectively. The stability of this algorithm is proofed too.

In section 6, the indirect experiments done on land are described. To further verify the coordination strategy, we construct a serial of indoor experiments with limited communication condition, in which, NAO humanoid robots are used to build the pyramid pattern to test the performance of the coordination strategy. The results show the feasibility and stability of this strategy, but also expose some shortcomings in the image processing and movement control of NAOs.

In section 7, the conclusions and prospects are stated. This section concludes the whole thesis, summarizes the main achievements and innovations, analyzes some defects of image processing and formation control methods, and states the future work.

1.4 Chapter summary

In this chapter, we introduce the development of AUV, multi-AUV system, summarize the

formation control methods, and describe our research work. The main work is to propose a coordination strategy to achieve the building of given formation, which includes two parts: (1) designing a robust formation independent of the number of AUVs; (2) proposing two kinds of control methods, local position-based method and asynchronous distributed method based on the local information got from cameras due to the obstacle-free environment and general environment respectively. In the local position-based method, a collision avoidance strategy with complexity $O(n\log n)$ is given based on relevant graph theories to optimize the distribution of given formation, optimize the destinations of AUVs in the given pattern, identify the expected relationship among AUVs, and generate the non-updated non-intercrossing straight trajectories. In the asynchronous distributed control method, the trajectories are updated continuously by the discrete consensus algorithm with time-variant weighted digraph based on the local information neighbors. The neighbors are determined by the expected position relationship generated by collision avoidance strategy. The coordination strategy is tested by simulations and real experiments indoors with limited communication as similar as the underwater environment. The results imply the feasibility and stability of this coordination strategy.

2 Formation definition of multi-AUV system

As described in section 1.3.1, the formation structure should be robust to the disturbances and should be independent to the number of AUVs in the multi-AUV system. Since the AUVs can be seen as the points, the information exchange among them can be expressed by the edges connecting them, then the multi-AUV system can be regarded as a digraph or an undirected graph. To define a formation meeting all the demands given in section 1.3.1, relevant graph theories needs to be described beforehand.

2.1 Relevant definitions and notes of graph theory

Generally, a multi-robot system can be seen as a graph $G(V, E)$. Robot-*i* is one vertex of graph G, noted as v_i. V is the set including all the vertexes, $v_i \in V$. e_{ij} is the communication channel between robot-*i* and robot-*j*, which is the edge connecting v_i and v_j. E is the set including all edges, $e_{ij} \in E$. According to the way of information exchange among AUV units, $G(V, E)$ can be modeled as a digraph or undirected graph. If G is a digraph, e_{ij} expressed one edge from v_i to v_j (communication from robot-*i* to robot-*j*). For an undirected graph, e_{ij} represents that robot-*i* and robot-*j* can share their information synchronously in an ideal environment.

Definition 2-1 Connected graph: in an undirected graph, if it exists at least one path from v_i to v_j, it is said that v_i and v_j are connected. If any two vertexes v and v' are connected, it is said that G is a connected graph.

Assuming that G is a digraph, v_i and v_j are said to be strongly connected if there is a path in each direction between v_i and v_j. And G is said to be strongly connected, if any two vertexes v and v' are strongly connected [85,86].

Definition 2-2 Root point: a connected graph can be seen as a rooted tree, since there is always one vertex connecting all the other vertices by the edges. This vertex is noted as the root point, such as the point v_0 in Fig. 2-1. The root point is commonly known as a leader in a leader-follower

formation.

Definition 2-3 Parent/child/brother: in a rooted tree, the parent of a vertex is the vertex connected to it on the path to the root; every vertex except the root has a unique parent. A child of a vertex v is a vertex of which v is the parent. A sibling (brother) to a vertex v is any other vertex on the tree which has the same parent as v. As shown in Fig. 2-1, v_0 is the parent of v_i and v_h, v_i and v_h are brothers, v_j is a child of v_i, but v_j and v_k are not brothers.

Definition 2-4: Depth/height: in a rooted tree, the height of a vertex in a rooted tree is the length of the longest downward path to a leaf (the vertex without children) from that vertex. The height of the tree is the height of the root point. The depth of a vertex is the length of the path to its root point (root path).

The depth of the root point is zero, the height of leaves is zero, and a tree with only a single vertex (hence both a root and leaf) has depth and height equal to zero. Conventionally, an empty tree (tree with no vertices, if such are allowed) has depth and height equal to −1. For example, the height of the tree shown in Fig. 2-1 is 2.

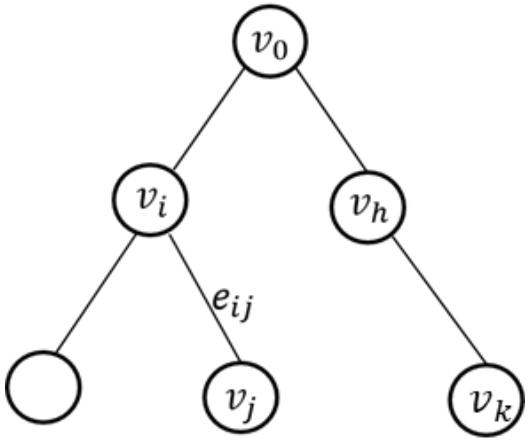


Fig. 2-1 One representation of a graph $G(V, E)$

Definition 2-5 A spanning tree: $G(V, E)$ contains a spanning tree if there exists a root, such that all other vertexes can be linked to this root via a directed path [87]. Usually, a $G(V, E)$ contains several spanning trees.

Definition 2-6 Balanced graph/weighted digraph: a digraph is called a balanced graph if the total values of edges entering a vertex and leaving itself are equal for all vertexes, whose information exchange way is similar to the undirected graph [88]. Otherwise, the digraph is called a weighted digraph.

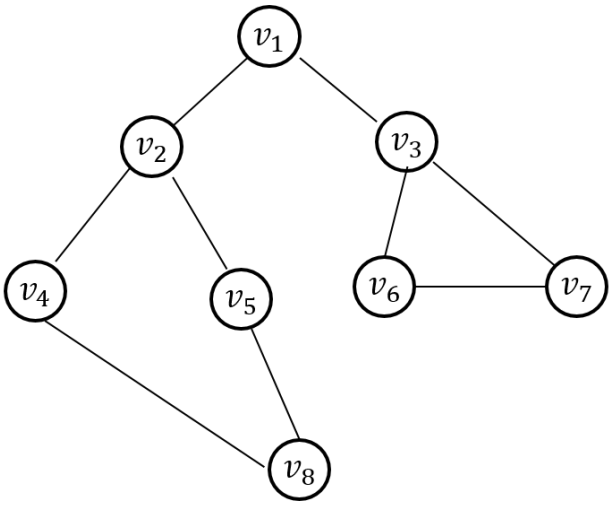
Definition 2-7 Adjacency matrix: an adjacency matrix is a square matrix used to represent a graph, noted as $\mathbf{A}=[a_{ij}]\in \mathrm{R}^{N\times N}$, a_{ij} is the value of edge e_{ij}, N is the number of vertices. $a_{ij}>0$, if $e_{ij}\in E$; $a_{ij}=0$, if $e_{ij}\notin E$ [81]. $\mathbf{A}$ is a symmetric matrix if $G(V,E)$ is an undirected graph or balanced graph.

Definition 2-8 Laplacian matrix: The Laplacian matrix $\mathbf{L}$ of a graph $G(V,E)$ is defined as $\mathbf{L}=\Delta-\mathbf{A}$, where $\Delta=diag(d_i)$ is the degree matrix of $G(V,E)$ with the diagonal elements $d_i=\sum_{j=1}^{N}a_{ij}$.

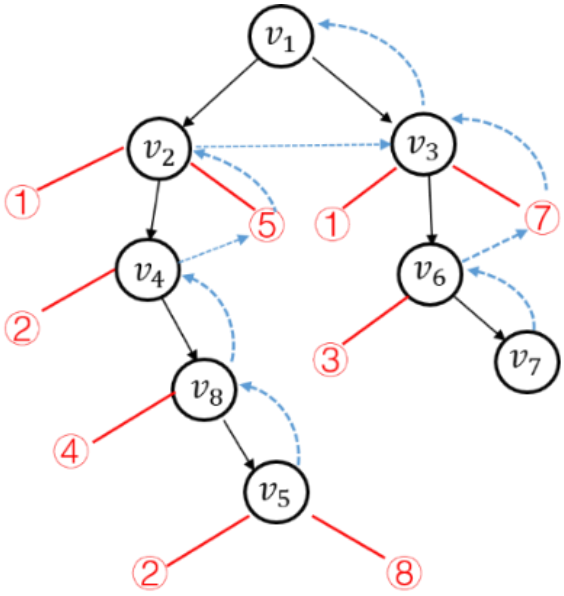
Lemma 2-1 The adjacency matrix $\mathbf{A}$ and the Laplacian matrix $\mathbf{L}$ both are irreducible if their associated digraph is strongly connected [90].

Definition 2-9 Graph traversal: graph search (also known as graph traversal) refers to the process of visiting (checking and/or updating) each vertex in a graph. Such traversals are classified by the order in which the vertices are visited.

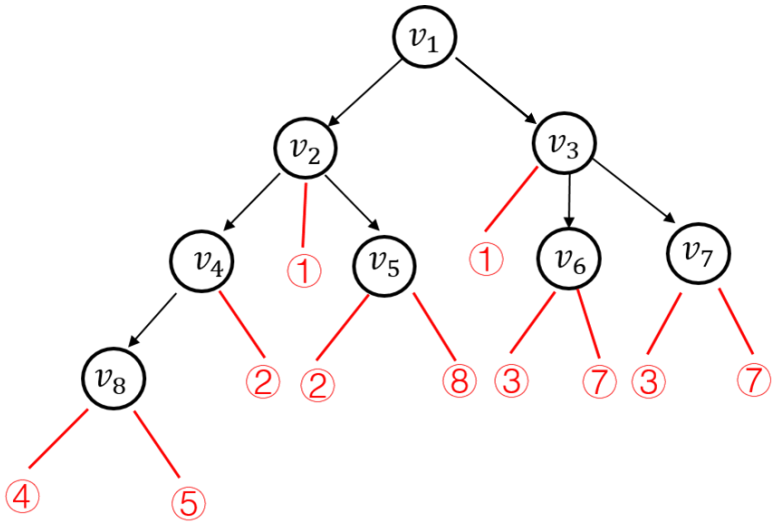
Graph search includes two kinds of ways: Depth First Search (DFS) and Breadth First Search (BFS) [91]. Shown in Fig. 2-2, Fig. 2-2(a) shows an undirected graph G_i to be traversed, Fig. 2-2(b) shows the search process of DFS, with the result $v_1\to v_2\to v_4\to v_8\to v_5\to v_3\to v_6\to v_7$; while Fig. 2-2(c) shows the search process of BFS, with the result $v_1\to v_2\to v_3\to v_4\to v_5\to v_6\to v_7\to v_8$.



(a) Undirected graph G_i



(b) The process of Depth First Search (DFS)



(c) The process of Breadth First Search (BFS)

Fig. 2-2 Explanation of two kinds of graph search: DFS and BFS. The solid lines with arrows represent the access paths when traversing, while the dotted lines with arrows represent the return paths; the red nodes stand for the nodes that have been visited when the black nodes are being visited.

Remark: Considering that each vertex may be adjacent to the other vertices, it is inevitable that the same vertex will be visited more than once. To avoid duplication, it is necessary to mark the visited vertices in the traversal process.

2.2 Analysis of formation structure

To define a robust pattern formation to confront the complex disturbances in the water, the

analysis of formation structure is necessary. Since the structure can be regarded as one graph, the corresponding Laplacian matrix of this graph can be discussed to analyze the influences of the eigenvalues of Laplacian matrix on the robustness of structure in this part.

According to definition 2-8, the row sum of Laplacian matrix $\mathbf{L}$ is 0. When there are at least 2 vertices in the graph *G*, its eigenvalues are noted as $0=\lambda_1 \le \lambda_2 = a(G) \le \lambda_3 \le ... \le \lambda_N$, and the eigenvector corresponding to the eigenvalue 0 is $\mathbf{1} = [1,1,...,1]^T_{N\times 1}$. In this section, we only discuss the robustness of formation structure, so that the weights of all the edges of *G* are set to be equal to 1. In section 5, the weights of edges are set differently to calculate the confidences of neighbors.

Knowable by lemma 2-1, when graph G is strongly connected, the Laplacian matrix $\mathbf{L}$ is irreducible. It means that $\mathbf{L}$ is a semi-positive definite matrix with only one of its eigenvalues equal to 0, $0 = \lambda_1 < \lambda_2$.

Lemma 2-2[92] If the row sum of a positive definite matrix $\mathbf{M}$ is equal to 0, then its second smallest eigenvalue meets $a(G) = \min_{\mathbf{x}\in W} \mathbf{x}^T \mathbf{M} \mathbf{x}$. In which, $W = \{\mathbf{x} \mid \mathbf{x} \in \mathbf{x}_1, \mathbf{x}_2, ...\}$ is the set of all vectors.

Lemma 2-3[92] If G_1, G_2 are edge-disjoint graphs with the same set of vertices, then $a(G_1) + a(G_2) \le a(G_1 \bigcup G_2)$.

Proof: According to definition 2-8, the row sum of Laplacian matrix is 0. And we have $\mathbf{L}(G_1 + G_2) = \mathbf{L}(G_1) + \mathbf{L}(G_2)$, based on the lemma 2-2, $a(G_1 \bigcup G_2) = \min_{\mathbf{x}\in W}(\mathbf{x}^T \mathbf{L}(G_1)\mathbf{x} + \mathbf{x}^T \mathbf{L}(G_2)\mathbf{x}) \ge \min_{\mathbf{x}\in W} \mathbf{x}^T \mathbf{L}(G_1)\mathbf{x} + \min_{\mathbf{x}\in W} \mathbf{x}^T \mathbf{L}(G_2)\mathbf{x} = a(G_1) + a(G_2)$.

Corollary 2-1[92] The function $a(G)$ is non-decreasing for graphs with the same set of vertices, i.e. $a(G_1) \le a(G_2)$ if $G_1 \subseteq G_2$ (and G_1, G_2 have the same set of vertices).

Proof: Assume that $G_2 = G_1 \cup G_3$, then $G_1 \subseteq G_2$. According to lemma 2-3, $a(G_1) + a(G_3) \le a(G_2)$, then $a(G_1) \le a(G_2)$.

Corollary 2-2 For a connected graph G, the more the number of edges connecting vertices, the greater the second smallest eigenvalue $a(G)$ will be and the more constraints among vertices will be.

In the graph with the maximum value of $a(G)$, $\max a(G)$, any two vertices are connected by edges.

Lemma 2-4[92] Let $\mathbf{M}=(m_{ik})_{N\times N}$ be a symmetric positive semi-definite matrix such that $\mathbf{M1}=\mathbf{0}$. Then the second smallest eigenvalue λ_2 of $\mathbf{M}$ satisfies

$$\lambda_2 \le [N/(N-1)]\min_i m_{ii} \tag{2-1}$$

Proof: According to lemma 2-2, $\lambda_2=\min_{\mathbf{x}\in W}\mathbf{x}^T\mathbf{M}\mathbf{x}$. Then let us define a new matrix $\tilde{\mathbf{M}}=\mathbf{M}-\lambda_2(\mathbf{I}-N^{-1}\mathbf{1}\mathbf{1}^T)$. Let $\mathbf{y}=c_1\mathbf{1}+c_2\mathbf{x}$ (c_1 and c_2 are two constants).

Since $\tilde{\mathbf{M}}\mathbf{1}=\mathbf{M}\mathbf{1}-\lambda_2(\mathbf{I}-N^{-1}\mathbf{1}\mathbf{1}^T)\mathbf{1}=\mathbf{0}-\lambda_2(\mathbf{1}-N^{-1}N\mathbf{1})=\mathbf{0}$, and similarly we get $\mathbf{1}\tilde{\mathbf{M}}=\mathbf{0}$, then $\mathbf{y}^T\tilde{\mathbf{M}}\mathbf{y}=c_2^2\mathbf{x}^T\tilde{\mathbf{M}}\mathbf{x}=c_2^2(\mathbf{x}^T\mathbf{M}\mathbf{x}-\lambda_2)\ge 0$.

As a result, $\tilde{\mathbf{M}}$ is also nonnegative, which means that the minimum value of its diagonal elements is nonnegative, $\min_i m_{ii}-\lambda_2\ge 0$. As $\lambda_2\ge 0$, then $\min_i m_{ii}-\lambda_2(1-N^{-1})\ge 0$. Then we get $\lambda_2\le[N/(N-1)]\min_i m_{ii}$.

Lemma 2-5[92] Let $G(V,E)$ be a graph, n_{v_i} be the valency (the number of participating in constituting edges) of vertex v_i, n_E be the number of all the edges, then $a(G)\le[N/(N-1)]\min_i n_{v_i}\le 2|n_E|/(N-1)$.

Proof: Because of $N\min_i n_{v_i}\le\sum_i n_{v_i}=2|n_E|$, then the second inequality holds. And the diagonal elements of Laplacian matrix meet $l_{ii}=n_{v_i}$. According to lemma 2-4, the first inequality holds.

Corollary 2-3 The larger second smallest eigenvalue, the more stable the structure will be.

Proof: According to corollary 2-1, if $G_1\subseteq G_2$ (and G_1, G_2 have the same set of vertices), then $a(G_1)\le a(G_2)$, and $E_1\subseteq E_2$. Meanwhile, the second smallest eigenvalue $a(G)$ is related to $\min_i n_{v_i}$ according to lemma 2-5. The more edges among vertices, the higher value of $a(G)$ will be. As the more edges among vertices, the more stable the structure will be, therefore, based on lemma 2-3, corollary 2-1 and corollary 2-2, corollary 2-3 is proofed well.

2.3 Definition of planar pyramid pattern

To define a robust pattern, firstly the properties of the multi-AUV systems considered in this thesis are given below:

(1) To get enough local information from local sensors, there are at least 3 homogeneous AUV units equipped with the same sensors and sharing the same algorithms. Each AUV unit has a unique ID to distinguish with each other.

(2) The system has limited communication structure. Most of surrounding information is acquired by each AUV's on-board camera.

(3) The camera sensing is one-way, which means the graph composed by all the AUV units can be considered as a digraph. If AUV-*i* finds a neighbor AUV-*j*, then $e_{ij} > 0$.

To simplify the formation building process, we assume moreover that:

(1) Because of the short working range of on-board cameras in the water, all the AUV units are located on the same depth in a bounded area (2D plane) at the beginning, noted as the initial distribution, $V_I = \{p_{c1}, p_{c2}, \ldots p_{cn}\}$, where p_{ci} expresses the initial position of AUV-*i* .

(2) The neighbors (AUV-*j*) found by AUV-*i* make up to one sub-set $V_{Ii} = \{p_{cj}, \ldots\}$, where $V_{Ii} \subseteq V_I$, and $V_{Ii} \neq \varnothing$ (if $V_{Ii} = \varnothing$, it means that AUV-*i* does not find any neighbors, then AUV-i will fine tune its position and search its surroundings again until it finds at least one neighbors), $V_I = \bigcup_{i=1}^{N} V_{Ii}$.

(3) Ignoring the errors of compass, the accuracy of measured global orientations of AUVs meets our demands.

(4) Required communications among AUVs have been achieved by acoustic communication without loss of information.

The single camera is installed on the front of each AUV unit (see Fig. 1-22). By this way, each AUV can detect the neighbors viewed by its own camera and estimate their relative positions in its own frame. Since when moving forward, the AUVs can only find and follow the neighbors ahead of itself without the assistance of global sensors, a leader-follower model is chosen to be as the formation [41]. In this model, the leader is identified according to the initial distribution, the other

AUVs follow this leader or other AUVs ahead of themselves. To get a coverage area as large as possible, reduce the searching time cost and to increase the workload within the same time period, the formation should trend to be a horizon line with large coverage (perpendicular to the moving direction).

However, except for the coverage, robustness ability (stability) is another significant element of the formation to keep the formation even it encounters disturbances. Based on the structure analysis in section 2.2, the robustness of a formation can be analyzed according to the second smallest eigenvalue $a(G)$ of the corresponding Laplacian matrix. The larger the value of $a(G)$ is, the formation robustness will be better according to the corollary 2-3[92,93]. And the value of $a(G)$ is decided by the constraints among AUV units. The more the constraints is, the larger of the value of $a(G)$ will be. Hence, though a horizon line formation has the largest coverage area, the constraints among AUVs exist only between the adjacent AUV units (such as the linear formation in [64]). The horizon line formation is poorly stable because of the weak constraints among vertices, and this horizon line formation require a huge view field of underwater cameras, at least 180°. In this thesis, an inverted "V" formation with a relative large coverage area is introduced. At the same time, to reinforce the constraints among AUVs and reduce the communication distances, several AUV units are filled in the inter part of the inverted "V" formation (such as the AUV-4, AUV-7, and AUV-8 in Fig. 2-3), which is also suitable to compensate the short working range of camera. Therefore, inspired by the triangular formation [42,44], swarm intelligent control [94] and stability analysis of the network dynamic system [92,93], we propose a planar pyramid pattern (see Fig. 2-3) [95].

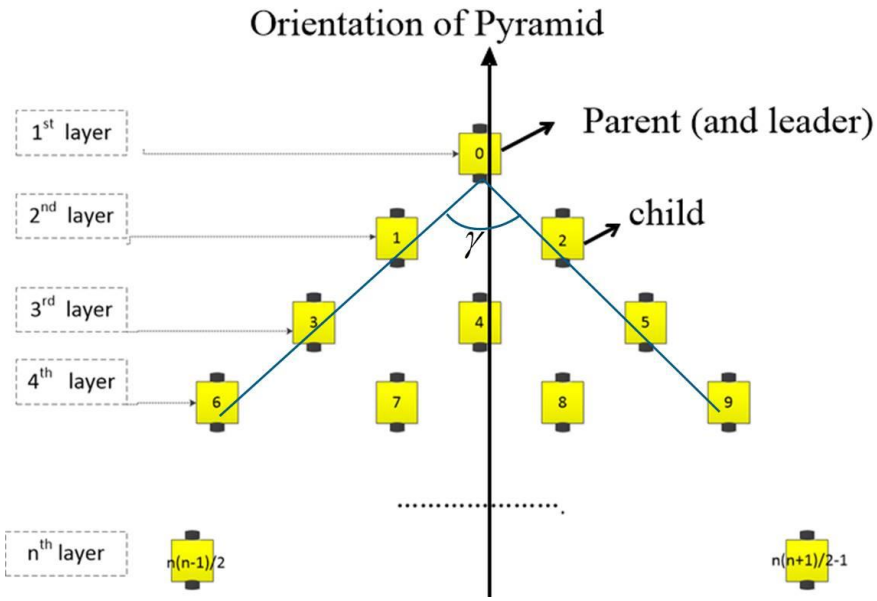


Fig. 2-3 Layout of a general planar pyramid pattern

The definition of the pyramid is presented below:

(1) The angle γ of the entire pattern is fixed by the view angle of the camera.

(2) There are i robots in the i_{th} layer, $i \geq 1$. And each robot has an order, the serial number of robot position (the number written on robots shown in Fig. 2-3). The order is equal to the sum of robots before itself layer by layer.

(3) The robot in the 1_{st} layer is called the leader. Each of the other robots follows the leader or one robot in front of itself. For ease of description, the followers and the robots to be followed are noted as children and parents, respectively. Each robot (located at the i_{th} layer, $i > 1$, except the leader), is the parent of one robot behind itself (at the $(i+1)_{th}$ layer), at the same time, it is also one child of one robot in front of itself (at the $(i- 1)_{th}$ layer) except the robots located at the last layer.

(4) Each robot has only one parent. Except for the last robot of each layer whose parent is located at its left-front, the parents of other robots are located at their right-front. The adjacent robots at the same layer are brothers, each robot (except the leader) has at least one brother, but no more than two brothers. This kind of parent-children-brother relationship is designated as position relationship.

(5) The distance and the angle between a parent and its child are fixed: d_s and $\pm\varphi_s = \frac{1}{2}\gamma$ (for the last robot of each layer, the angle is '+', while the angle is '-' for the other robots of the same layer).

To plan trajectories for all AUVs to build this pattern from their initial positions, formation control methods will be proposed. Two crucial problems need to be solved: one is to find the associated destination in the pyramid pattern for each robot in the initial distribution; the other one is to avoid collision between AUVs when moving. To match up with the destinations, the posture of the pyramidal pattern (position and orientation) needs to be first identified depending on the initial distribution. Secondly, collision avoidance trajectories should be planned in obstacle free and general situations, respectively.

The details of the approaches studied and proposed are given in chapter 4 and 5.

2.4 Chapter summary

In this chapter, we define a pyramid pattern formation for our multi-AUV system. Firstly, the robustness of the structure is analyzed by using the corresponding Laplacian matrix of graph, since the topology structure of multi-AUV system can be regarded as a graph. According to the analysis, we get that the more constraints among AUVs have, the more robust of the system will be. And to get a relative large coverage, combing with the camera property of our multi-AUV system, a leader-follower model is introduced. Then, to reduce the communication distance and difficulty, and adjust to the short working range of underwater cameras, the redundant AUV units are filled in the inter part of the leader-follower model. Finally, a planar pyramid pattern is defined and its properties of this pyramid pattern are also described in detail.

3 Neighboring information acquisition

In the ocean engineering, to obtain abundant information by video cameras in the water, image de-blurring is essential [96,97]. The qualities of optical images are significant to accomplish scientific tasks, such as marine life monitoring, statistics of marine species, underwater archaeology etc. However, the underwater images got directly from the cameras are blurred because of the reflection and scattering of light by particles in the water, and the colors are distorted because of the different attenuations of different wavelengths. For example, the red light will vanish at a distance of 3m, the orange and yellow lights will disappear at a distance of 10m, and the green light will dissolve at a distance of 20m, while the blue light will traverse the longest distance, about 30m, shown in Fig. 3-1. Consequently, obtaining one satisfying underwater image is a challenge [97].

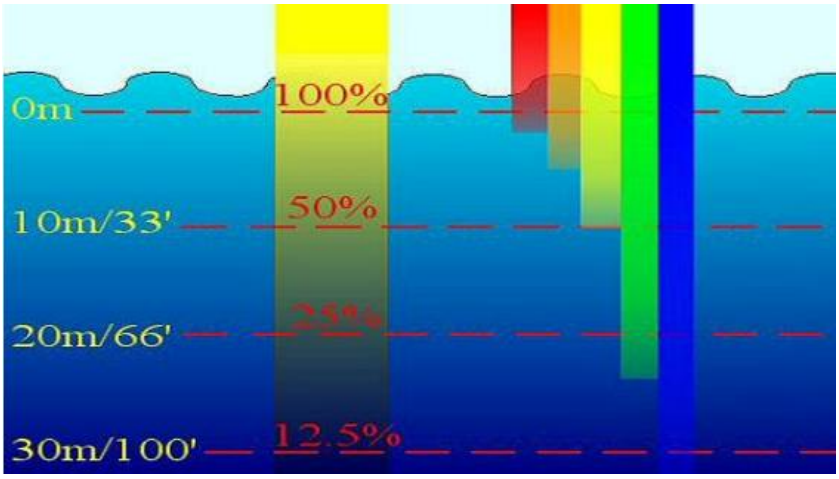


Fig. 3-1 Underwater spectrum of light. The lights from red to blue will disappear gradually at different depth in the water because of their different wavelengths.

Fig. 3-2 shows some yellow pipelines put down on the seafloor used to locate the positions of AUVs through being detected by AUVs in the "Student Autonomous Underwater Vehicles Challenge-Europe, SAUC-E" competition [98]. Fig. 3-2(a) shows a view of the pipeline image on land, while Fig. 3-2(b) presents an underwater image of pipelines acquired by the on-board camera of one CISCREA in the bay of the CMRE (Center for Maritime Research and Experimentation) at Las Pezia, Italy. Obviously, we find that the underwater image is blurred and distorted.

Therefore, before to extract relevant information from on-board cameras to identify neighboring AUVs for each AUV, we need to improve the qualities of the acquired underwater

images. The image process mainly includes 5 parts [99,100]: (1) image acquisition; (2) image pre-processing; (3) image segmentation; (4) feature extraction; (5) image explanation.

(a) One image of pipeline on land (b) One image of pipeline in the water

Fig. 3-2 Comparison of the images of yellow pipeline on land and in the water

3.1 Image pre-processing of underwater images

3.1.1 Some state of the art

As a key step of image processing, image pre-processing is generally used to correct the scattering, reflection and absorption of light waves by water. It includes homomorphic filter [101-103], bilateral filter [104-106], anisotropic filter [107-109], constrast limited adaptive histogram equalization [110-112] and wavelet de-noising [113] etc.

(1) Homomorphic filter

Homomorphic filter is mainly used to correct non-uniform illumination. It is based on an illumination reflection model. One digital image $f(x,y)$ includes two sections: an illumination section $i(x,y)$ and a reflection section $r(x,y)$, modeled in equation (3-1).

$$f(x,y) = i(x,y) \cdot r(x,y) \tag{3-1}$$

Illumination section $i(x,y)$ stands for the total amount of incident light into the scene. In this model, the illumination section is assumed to vary uniformly, and it corresponds to the low frequency part of the Fourier transform of $f(x,y)$. Reflection section $r(x,y)$ represents the total amount of light reflected by the objects in the scene, and it is corresponding to the high frequency part of the Fourier transform of $f(x,y)$.

To improve the dynamic range and the contrast of $f(x,y)$, $i(x,y)$ should be suppressed, while the $r(x,y)$ should be enhanced. To separate these two sections, the equation (3-1) is

rewritten to equation (3-2).

$$g(x,y) = \ln(i(x,y)) + \ln(r(x,y)) \tag{3-2}$$

After the Fourier transform, equation (3-3) is obtained.

$$F\{g(x,y)\} = G(u,v) = I_l(u,v) + R_l(u,v) \tag{3-3}$$

At last, an homomorphic filter $h(x,y)$ (with FFT, $H(u,v)$) is designed to reduce the influence of the low-frequency part ($i(x,y)$) and to enhance the mid- and high-frequency part ($r(x,y)$), shown in equation (3-4).

$$S(u,v) = H(u,v) \cdot G(u,v) \tag{3-4}$$

Where $H(u,v) = \dfrac{1}{1+e^{-s(\sqrt{u^2+v^2}-w_0)}} + \rho$, w_0 is the cutoff frequency, s is a coefficient to be used to control slope steepness, and ρ is an offset.

(2) Bilateral filter

Bilateral filter is a non-linear weighted mean filter, proposed by Tomasi and Manduchi in 1988 [104]. It extends the concept of Gaussian smoothing and carries out spatial averaging while preserving edges. Bilateral filter includes two functions: one function $d(x,y)$ denotes the geometric closeness between the pixel of interest x and a neighborhood pixel y, whose value decreases with the increasing of the distance between x and y. The other function $r(f(x), f(y))$ is used to find the similarity between the intensity of $f(x)$ and $f(y)$, whose value decreases with the increase of the pixel difference. By this way, it can not only keep the edges, but also smooth the noises [105,106]. The bilateral filter can be expressed by

$$\%(x) = \eta^{-1} \int d(x,y) r(f(x), f(y)) f(y) dy \tag{3-5}$$

Where, η is a standardization factor, $\eta = \int d(x,y) r(f(x), f(y)) dy$ [106].

(3) Anisotropic filter

Anisotropy refers to the difference of mechanical and physical properties of materials in all directions. For an image, anisotropy expresses that the gradient variations of each pixel in four directions around it are different. This filter overcomes the shortcoming of isotropy of Gaussian filter. Anisotropic filter processes an image as a heat field, and each pixel is seen as a heat flow, which always flows to a place with similar value. For instance, if there is a big difference between a neighborhood pixel and the current pixel, the neighborhood pixel may be on the edge, then the

current pixel will not diffuse along that direction, so that the boundary is preserved.

Anisotropic filter realizes the image denoising by partial differential equation (PDE) [107]. A series of smoothed images are obtained by diffusion operators (see equation (3-6) [108]) in all directions.

$$\begin{cases} \dfrac{\partial u}{\partial t} = div[c(|\nabla u|)\nabla u] \\ u(x, y, t=0) = u_0(x, y) \end{cases} \tag{3-6}$$

Where, ∇ is the gradient operator, div is the direction operator, which expresses four directions, $u_0(x, y)$ is the initial pixel value, $c(\cdot)$ is the diffusion factor with positive value, which can be calculated by the formula of thermal conductivity:

$$c(|\nabla u|) = \exp\left(-\left(\frac{|\nabla u|}{k}\right)^2\right) \tag{3-7}$$

Where, k is the reference coefficient. The smaller the value of k, the more edge information can be retained, while the larger the value of k, the stronger the denoising ability will be. And, the weak edges and small details will be smoothed [108,109].

At different times, the diffusion process can be expressed by

$$I_{t+1} = I_t + \lambda(c_{N_{x,y}}\nabla_N(I_t) + c_{S_{x,y}}\nabla_S(I_t) + c_{E_{x,y}}\nabla_E(I_t) + c_{W_{x,y}}\nabla_W(I_t)) \tag{3-8}$$

Where, I stands for images, t means the iteration times. $\nabla_N(I_t)$, $\nabla_S(I_t)$, $\nabla_E(I_t)$, $\nabla_W(I_t)$ represent the gradient operators in four directions, $c_{N_{x,y}}$, $c_{S_{x,y}}$, $c_{E_{x,y}}$, $c_{W_{x,y}}$ express the conduction coefficients at every iteration in four directions. λ is a coefficient for the numerical scheme to be stable, usually, $0 \le \lambda \le 0.25$.

(4) Contrast Limited Adaptive Histogram Equalization (CLAHE)

CLAHE is first proposed by Pizer et al. in 1987 for low-contrast medical images [110]. CLAHE is an extension of adaptive histogram equalization, in which a contrast-constrained parameter, Clip Limit (CL), is introduced to avoid noise amplification and uniformly improve the contrast. With this method, the origin image is divided into blocks first, and the cumulative distribution histogram of each sub-image block is calculated. Then the processed adjacent sub-image blocks are further integrated by bilinear interpolation. In the generated image, the pixels are evenly distributed, which makes the hidden features more visible.

The equation using CLAHE to calculate new pixel gray values is [111]:

$$p = [p_{max} - p_{min}] * P(f) + p_{min} \quad (3\text{-}9)$$

In which, p_{max} and p_{min} are the maximum and minimum pixel values of the processed image respectively, p_{max} is limited by the value of CL, $P(f)$ is the cumulative probability distribution, p is the new generated pixel [112].

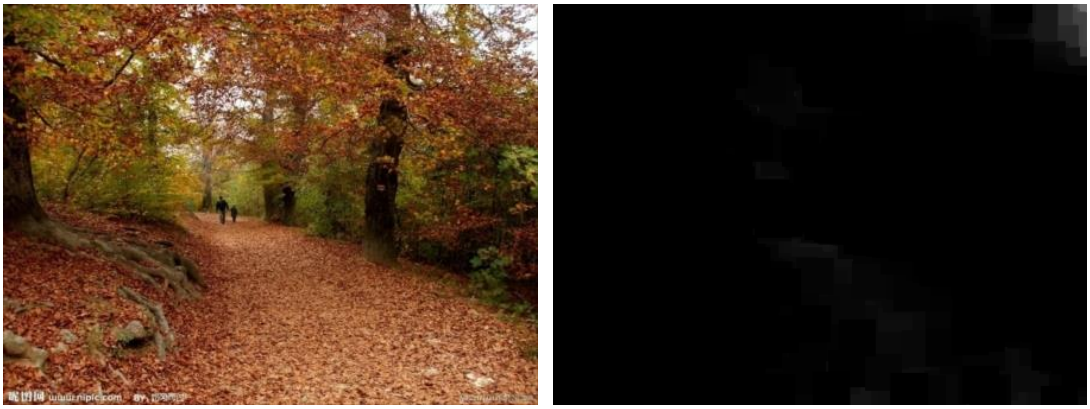
(5) Wavelet de-noising

Wavelet transform is widely used in image processing, because it can represent the local characteristics of signals in both spatial and temporal domains. To remove noises, wavelet transform transforms information from real space to wavelet space. Usually, we assume the signal noise collected is white Gaussian noise. In the wavelet space, the modulus of the wavelet coefficients corresponding to the effective signal is large, while the white Gaussian noise still shows strong randomness. Then the white Gaussian noise is still considered as the white Gaussian noise in the wavelet space, and the corresponding wavelet coefficients are small. Therefore, the noise can be filtered out by setting some threshold, and the soft threshold function with better effect is more suitable than hard threshold function [113].

These image processing methods mentioned above are able to enhance the qualities of underwater images to some extent, but they cannot remove the noises and correct the color distortion correctly. Many features still cannot be extracted effectively, which limits the application of cameras in the water. In this thesis, we find that a defogging algorithm is good to improve the qualities of the underwater images, which is described below.

3.1.2 Defogging algorithm

The basic idea of defogging algorithm is that the minimum pixel value of three color channels is very small, trending to 0. This channel is called dark channel [114], shown in Fig. 3-3(b). The values of dark channel pixels become larger in images (underwater or foggy images) due to the scattered light, shown in Fig. 3-4. As a result, the scattering information of light can be evaluated by these dark pixels, and then the quality of the image can be improved.



(a) Outdoor fog-free image (b) The corresponding dark channel of (a)

Fig. 3-3 One outdoor fog-free image and the corresponding dark channel

(a) An underwater image (b) Its corresponding dark channel

Fig. 3-4 An underwater image and its corresponding dark channel

1. Estimation of the dark channel

The mathematical expression of the dark channel is given in equation (3-10).

$$\mathbf{J}^{\text{dark}}(\mathbf{x}) = \min_{\mathbf{y}\in\Omega(\mathbf{x})}\left(\min_{c\in(r,g,b)}\mathbf{J}^{c}(\mathbf{y})\right) \tag{3-10}$$

In which, $\mathbf{J}^{c}$ is one color channel of the image $\mathbf{J}$, $\Omega(\mathbf{x})$ is one patch whose center is $\mathbf{x}$, and the size of this patch can be changed. The dark channel is the result of two minimal operators: $\min_{c\in(r,g,b)}$ represents the minimum value of RGB per pixel, $\min_{\mathbf{y}\in\Omega(\mathbf{x})}$ stands for the minimum filter. In a fog-free image outdoor, it can be observed that $\mathbf{J}^{\text{dark}}(\mathbf{x}) \to \mathbf{0}$ (see Fig. 3-3).

2. Model of fogging image

According to the literature [114], one fogging image can be expressed by

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t(\mathbf{x}) + \mathbf{A}(1-t(\mathbf{x})) \tag{3-11}$$

Where, $\mathbf{x}$ is one point of the scene, $\mathbf{I}(\mathbf{x})$ is the image captured by one camera, $\mathbf{J}(\mathbf{x})$ is the fog-free image corresponding to position $\mathbf{x}$, $t(\mathbf{x})$ is light transmittance, which is relevant to the

distance $t(\mathbf{x})=e^{-\beta d(\mathbf{x})}$, $\mathbf{A}$ is the global atmospheric light, which can be seen as a constant. As a result, to get $\mathbf{J}(\mathbf{x})$, $t(\mathbf{x})$ should be calculated.

3. Calculation of light transmittance $t(\mathbf{x})$

Assuming that $\mathbf{A}$ is a known value, and all the patches $\Omega(\mathbf{x})$ share the same light transmittance, noted as $\tilde{t}(\mathbf{x})$, then, for each channel c of the color images $\mathbf{I}$ and $\mathbf{J}$:

$$\frac{\mathbf{I}^c(\mathbf{x})}{\mathbf{A}^c}=t(\mathbf{x})\frac{\mathbf{J}^c(\mathbf{x})}{\mathbf{A}^c}+1-t(\mathbf{x}) \tag{3-12}$$

$$\min_{\mathbf{y}\in\Omega(\mathbf{x})}\left(\min_{c\in(r,g,b)}\left(\frac{\mathbf{I}^c(\mathbf{y})}{\mathbf{A}^c}\right)\right)=\tilde{t}(\mathbf{x})\min_{\mathbf{y}\in\Omega(\mathbf{x})}\left(\min_{c\in(r,g,b)}\left(\frac{\mathbf{J}^c(\mathbf{y})}{\mathbf{A}^c}\right)\right)+1-\tilde{t}(\mathbf{x}) \tag{3-13}$$

As $\mathbf{A}$ is positive, we can demonstrate that

$$\min_{\mathbf{y}\in\Omega(\mathbf{x})}(\min_{c\in(r,g,b)}\mathbf{J}^c(\mathbf{y}))=0,\quad \min_{\mathbf{y}\in\Omega(\mathbf{x})}\left(\min_{c\in(r,g,b)}\left(\frac{\mathbf{J}^c(\mathbf{y})}{\mathbf{A}^c}\right)\right)=0 \tag{3-14}$$

However, the existance of fog is the basic clue for human to percept the depth information [114][116], called as the aerial perspective. If the fogging factors are totally removed, the images will become unnatural, even lose the sense of depth. Thus, a constant parameter ω（$0<\omega<1$，usually set as $\omega=0.95$）is introduced into equation (3-14) and $\tilde{t}(\mathbf{x})$ is derived as:

$$\tilde{t}(\mathbf{x})=1-\omega\min_{\mathbf{y}\in\Omega(\mathbf{x})}\left(\min_{c\in(r,g,b)}\left(\frac{\mathbf{I}^c(\mathbf{y})}{\mathbf{A}^c}\right)\right) \tag{3-15}$$

4. Estimation of the atmospheric light $\mathbf{A}$

In one foggy image $\mathbf{I}(\mathbf{x})$, it can be shown that the sky is extremely similar to the atmospheric light $\mathbf{A}$, then in the sky area $\min_{\mathbf{y}\in\Omega(\mathbf{x})}\left(\min_{c\in(r,g,b)}\left(\frac{\mathbf{I}^c(\mathbf{x})}{\mathbf{A}^c}\right)\right)\to 1$, $\tilde{t}(\mathbf{x})\to 0$. Consequently, we can estimate the atmospheric light value $\mathbf{A}$ from the foggy image by means of dark channel map.

Firstly, we extract the most serious fogged pixels with the maximum values of the first 0.1% from the dark channel map, and record their positions $\mathbf{P}$. Secondly, we regard the maximum pixel value of these positions $\mathbf{P}$ in the origin image $\mathbf{I}(\mathbf{x})$ as the value of atmospheric light $\mathbf{A}$.

In the end, with the estimated light transmittance $t(\mathbf{x})$ and atmospheric light $\mathbf{A}$, the fog-free image $\mathbf{J}(\mathbf{x})$ can be estimated by equation (3-11).

3.1.3 Application of defogging algorithm in the water

Due to the light scattering, reflection and absorption, underwater images can be compared to foggy images in some sense, thus we decide to introduce the defogging algorithm to pre-process the images [96].

$$\mathbf{I}_\lambda(\mathbf{x}) = \mathbf{J}_\lambda(\mathbf{x})t_\lambda(\mathbf{x}) + \mathbf{B}_\lambda(1 - t_\lambda(\mathbf{x})), \lambda \in \{\text{red}, \text{green}, \text{blue}\} \quad (3\text{-}16)$$

$\mathbf{x}_\lambda$, $\mathbf{I}_\lambda(\mathbf{x})$, $\mathbf{J}_\lambda(\mathbf{x})$ have the same meaning with those in equation (3-11), λ is the wavelength. $\mathbf{B}_\lambda$ is the underwater isomorphic background light, and can be stated as $\mathbf{B}_\lambda = \max_{\mathbf{x}\in\mathbf{I}} \min_{\mathbf{y}\in\Omega(\mathbf{x})} \mathbf{I}_\lambda(\mathbf{y})$ [117,118]. $t_\lambda(\mathbf{x})$ summarizes the overall effects for both light scattering and color change suffered by light with wavelength λ traveling the underwater distance $d(\mathbf{x})$, which is a ratio between the retained energy $E_\lambda^{\text{residual}}(\mathbf{x})$ and the initial energy $E_\lambda^{\text{initial}}(\mathbf{x})$ of light with λ wavelength:

$$t_\lambda(\mathbf{x}) = \frac{E_\lambda^{\text{residual}}(\mathbf{x})}{E_\lambda^{\text{initial}}(\mathbf{x})} = 10^{-\beta(\lambda)d(\mathbf{x})} = \text{Nrer}(\lambda)^{d(\mathbf{x})} \quad (3\text{-}17)$$

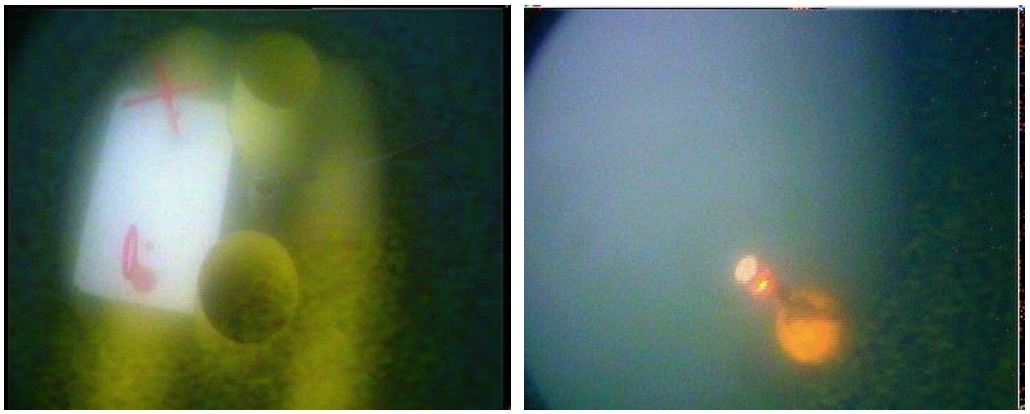
Where, $\beta(\lambda)$ is the extinction coefficient, $\text{Nrer}(\lambda)$ is the ratio of energy residuals to initial energy propagated per unit distance[118], which is relevant to the wavelength [119]. For example, shown in Fig. 3-1, the wavelength of red light is longer with lower frequency compared with blue light with shorter wavelength, consequently, it has faster attenuation speed than blue light, which results in the underwater images being mainly blue tones [120]. In addition to the wavelength, $\text{Nrer}(\lambda)$ is also affected by water salinity and plankton aggregation [118], as a result, sea water is further divided into three categories: the first category is very clear sea water; clear sea water with high attenuation level belongs to the second category; turbid and surging sea water belongs to the third category. These three types of water correspond to poor, medium, and eutrophic water bodies respectively [121]. To the first type of water body, for every unit of distance propagated by light with wavelength λ, $\text{Nrer}(\lambda)$ is 82%, 95%, 97.5% corresponding to red, green and blue lights. Based on $\text{Nrer}(\lambda)$ of the first type of water body, the normalized $\text{Nrer}(\lambda)$ can be adjusted to [96]:

$$\text{Nrer}(\lambda) = \begin{cases} 0.8 \sim 0.85 & if\ \lambda = 650 \sim 750\mu m(\text{red}) \\ 0.93 \sim 0.97 & if\ \lambda = 490 \sim 550\mu m(\text{green}) \\ 0.95 \sim 0.99 & if\ \lambda = 400 \sim 490\mu m(\text{blue}) \end{cases}.$$

However, because $d(\mathbf{x})$ is usually unknown, to simplify the process, similar to equation (3-15), a new equation (3-18) is given to compute $t_\lambda(\mathbf{x})$.

$$\tilde{t}_\lambda(\mathbf{x}) = 1 - \omega \min_{\mathbf{y}\in\Omega(\mathbf{x})}\left(\min_{c\in(r,g,b)}\left(\frac{\mathbf{I}_\lambda(\mathbf{x})}{\mathbf{B}_\lambda}\right)\right) \tag{3-18}$$

With the estimated light transmittance $\tilde{t}_\lambda(\mathbf{x})$ and underwater isomorphic background light $\mathbf{B}_\lambda$, the underwater image $\mathbf{J}(\mathbf{x})$ can be recovered by equation (3-16). As an illustration, Fig. 3-5 shows the application results of defogging algorithm on Fig. 3-2(b) and 3-4(a)



(a) (b)

Fig. 3-5 The recovered underwater images

Therefore, the qualities of underwater images can be improved by estimating the effect of light scattering and by compensating wavelength, the haze can be removed and the image color can be balanced, which means that the underwater camera can be regarded as the main sensors to get surrounding information in a certain range. The colors of images' background in Fig. 3-5 are still blue-green, which is caused by the simplified compensation of wavelengths when calculating the light transmittance $\tilde{t}_\lambda(\mathbf{x})$. If the wavelengths of different color lights are compensated accurately, the defogged underwater images will have higher qualities [96].

3.2 Identification of AUVs and acquisition of neighbors' local information

Since video cameras are the main sensors to sense the surroundings in this thesis, the neighboring AUVs should be detected from the background of images and the relative place information among AUVs should be estimated depending on the geometries of the detected AUVs.

3.2.1 Detection of AUVs

Based on the structure and hydrodynamic model of the real robot available in laboratory, CISCREA, (see Fig. 1-21), the 1:1 simulation models were constructed in 3D simulation software "Blender", shown in Fig. 3-6 [46]. To make the recognition of each CISCREA easier, two ways are explored: one way is by the introduction of explicit IDs in limited communication environment,

which are composed by 4 cylinders with white and/or black colors and are fixed on the top of CISCREA (see Fig. 3-7(a)); the other way is by implicit IDs, which are unique numbers written in the program of each robot (see Fig. 3-7(b)).

Due to the dominant and visible yellow part of CISCREA, the CISCREA are detected easily according to the yellow part with regular shapes from a single image, and the shapes from different view angles are similar. Then the CISCREA can be segmented from the surroundings in HSV space [46]. Setting the yellow threshold range as: $(25,100,100)$ ((Hmin, Smin, Vmin)) and $(100,255,255)$ ((Hmax, Smax, Vmax)), the yellow regions in the images will turn to white, while the other regions will turn to black. Then the open-close and close-open operations of morphological operations are used to do the image enhancement processing for getting the AUVs positions $(u_{ij}[k_i], v_{ij}[k_i])$ and the heights of yellow parts $h_{ij}[k_i]$ [47], shown in Fig. 3-8.

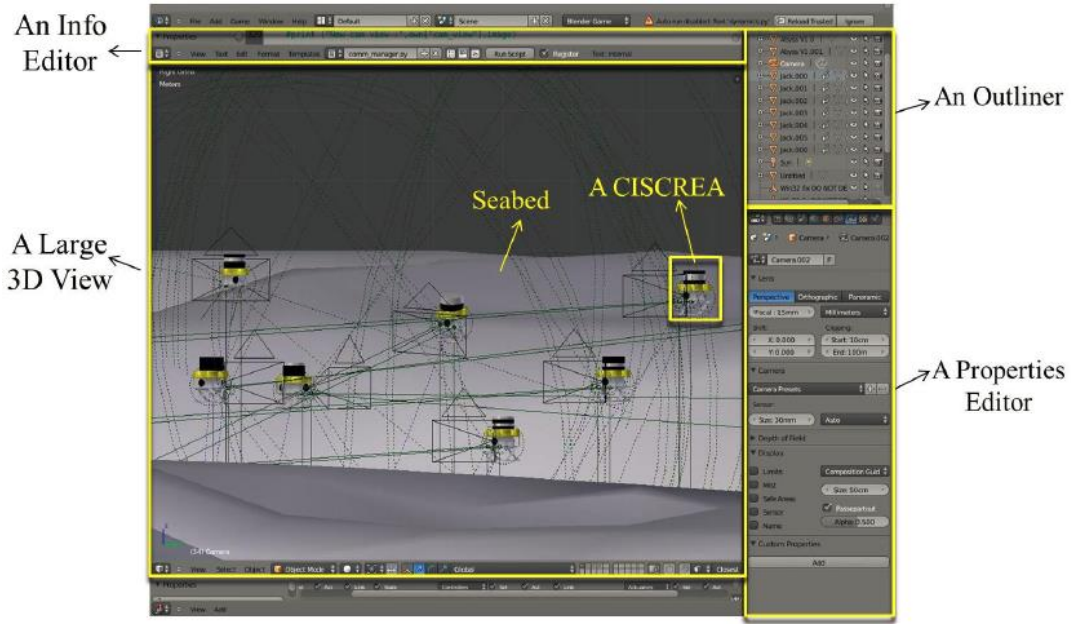
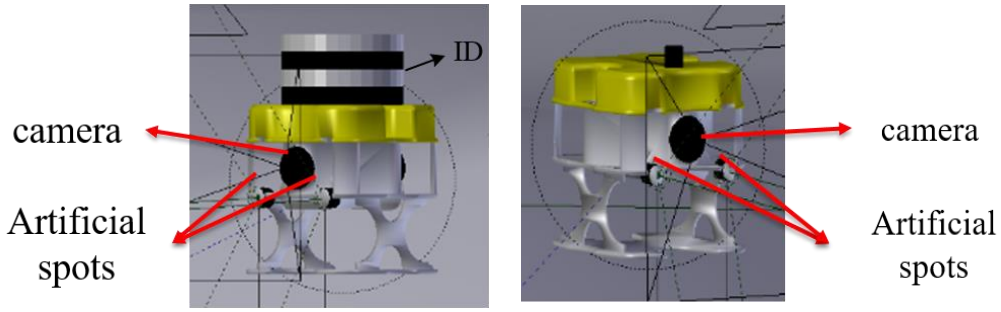


Fig. 3-6 3D simulation environment constructed in Blender



(a) Model with the explicit ID (b) Model with the implicit ID

Fig. 3-7 CISCREA simulation model

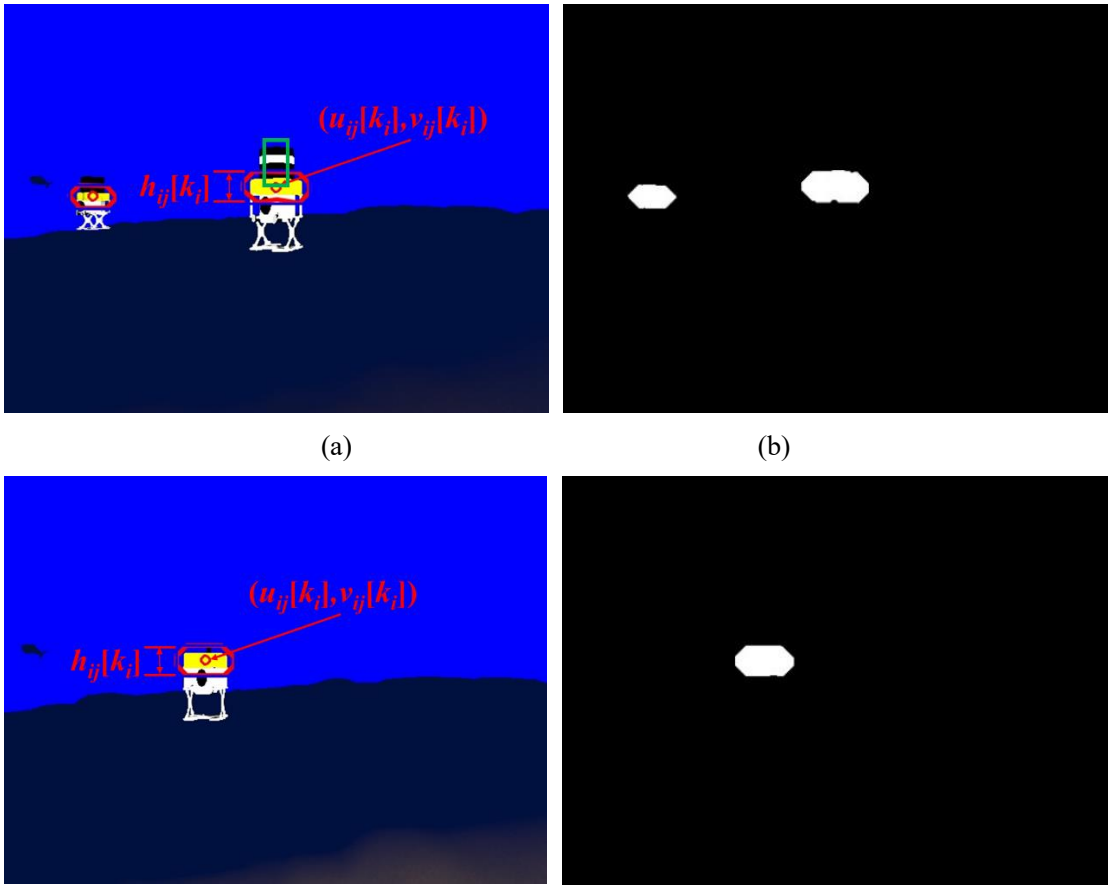


Fig. 3-8 Detection of AUVs: (a), (c) the positions and heights of AUVs shown in the original RGB images; (b), (d) the segmentation of the yellow parts.

3.2.2 Estimation of neighboring AUVs' local information

1. Proposal of local information model

To obtain the local information of neighbors, a camera frame camera-*i* (noted as $O_{ci}X_{ci}Y_{ci}$) and a local frame local-*i* (noted as $O_{li}X_{li}Y_{li}$) are introduced first. Since the hypothetical condition stated in section 2.3 is that all the AUVs are located at the same depth, $O_{ci}X_{ci}Y_{ci}$ and $O_{li}X_{li}Y_{li}$ both are planar frames (see Fig. 3-9 and Fig. 3-10). In which, the optical center of camera is regarded as the origin point O_{ci} of camera frame $O_{ci}X_{ci}Y_{ci}$, y axis is along the optical axis, while x axis is perpendicular to y axis in the horizontal plane. The origin point O_{li} of local frame $O_{li}X_{li}Y_{li}$ coincides with O_{ci}. We choose the north direction of compass as $O_{li}Y_{li}$, and $O_{li}X_{li}$ is perpendicular to $O_{li}Y_{li}$ in the horizontal plane. Ignoring the errors of compass of each AUV, then all the local frames $O_{li}X_{li}Y_{li}$ are parallel with each other.

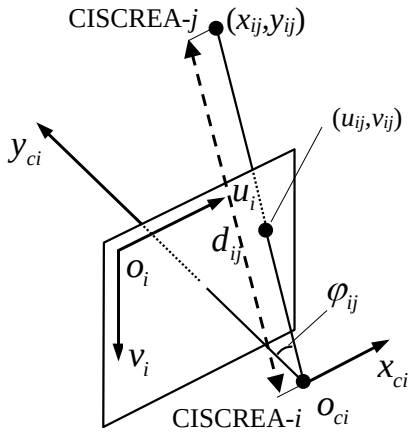


Fig. 3-9 Camera frame $o_{ci}x_{ci}y_{ci}$. o_{ci} is the optical center of camera, $o_{ci}y_{ci}$ is along the optical axis, while $o_{ci}x_{ci}$ is perpendicular to $o_{ci}y_{ci}$ in the horizontal plane. $o_iu_iv_i$ is the pixel frame of camera-*i*, o_iu_i and o_iv_i are along the width and height directions of images respectively. d_{ij} and φ_{ij} are the relative distance and relative angle between CISCREA-*j* and CISCREA-*i*. (x_{ij}, y_{ij}) and (u_{ij}, v_{ij}) is the coordinate of CISCREA-*j* in the $o_{ci}x_{ci}y_{ci}$ and in the $o_iu_iv_i$ respectively.

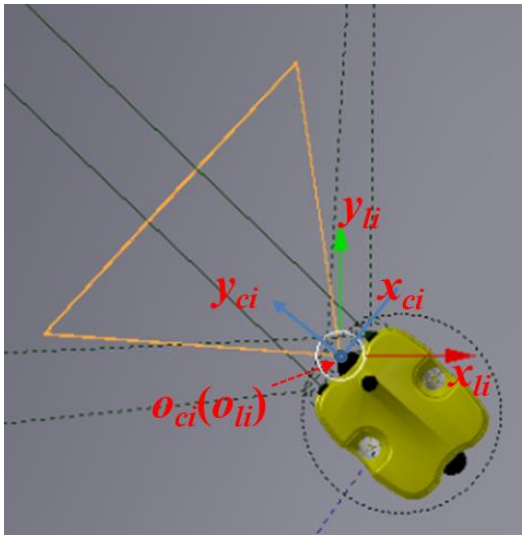


Fig. 3-10 Local frame of AUV $o_{li}x_{li}y_{li}$. o_{li} coincides with o_{ci}, $o_{li}y_{li}$ is always along to the north direction of compass, while $o_{li}x_{li}$ is perpendicular to $o_{li}y_{li}$ in the horizontal plane.

As it is generally not possible to get the depth information from a single 2D image, we construct a relative distance and relative angle models based on the camera model. With the assumption that the relative distance model and relative angle model are inter-independent and uncoupled, the relative distance model describes the relative relationship between relative distances and the sizes of AUVs in the images, while the relative angle model expresses the relative

relationship between relative angles and the positions of AUVs in the images.

To construct these two models, experimental data were obtained in Blender to be analyzed, and the details are given in the next part.

Then we find that the relative distance is inversely proportional to the height of the yellow region of CISCREA. To minimize the standard deviation, we adopt the sum of two exponential functions to express the distance model, shown in equation 3-19:

$$d_{ij} = a_1 e^{-b_1 h} + c_1 e^{d_1 h} \qquad (3\text{-}19)$$

Where, a_1, b_1, c_1, d_1 are the coefficients to be calibrated, h is the height of AUV-*j* in the image of AUV-*i*, d_{ij} is the relative distance between AUV-*j* and AUV-*i*.

And we find that the angle model expresses the linear relation between the u position of AUV in the images and the relative angle, shown in the equation 3-20:

$$\varphi_{ij} = a_2 u + b_2 \qquad (3\text{-}20)$$

In which, a_2, b_2 are the coefficients to be calibrated, u is the pixel value along the width direction of the image of AUV-*i*, φ_{ij} is the relative angle between the AUV-*j* and AUV-*i* in $o_{ci}x_{ci}y_{ci}$ (see Fig. 3-9).

2. Calibration of the models

To achieve the calibration of the models described above, the AUVs should be detected and identified from images, and their positions $(u_{ij}[k_i], v_{ij}[k_i])$, and their heights $h_{ij}[k_i]$ in images should be estimated beforehand.

1) Calibration of the relative distance model

To save time and to get more experimental data, three CISCREAs are structured in Blender to sample the data (relative distances d_i and heights h_i of the detected yellow regions of AUVs in images).

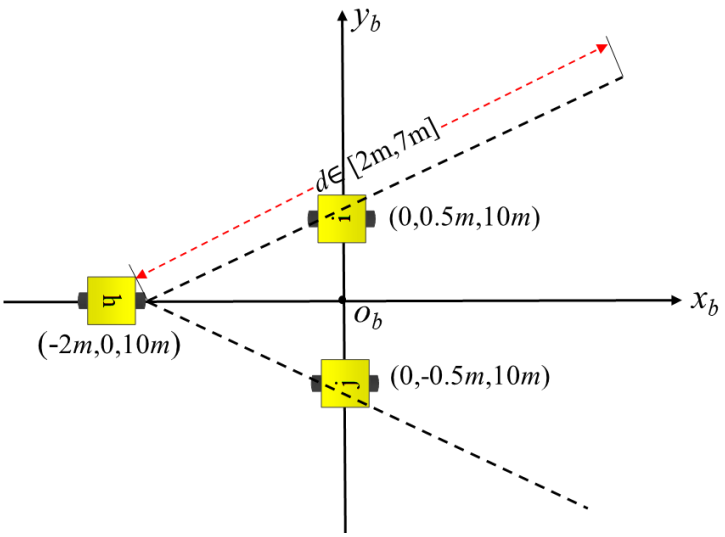


Fig. 3-11 The scheme of the positions of three CISCREAs in calibration. $o_b x_b y_b$ is the Blender frame (the frame given by Blender software).

The sampling steps are described below:

(1) Shown in Fig. 3-11, three 1:1 animulated robots (CISCREA-*i*, CISCREA-*j*, and CISCREA-*h*) are constructed in Blender according to the parameters of real CISCREA [84]. Their IDs are composed by 2 cylinders with white and/or black colors, shown in Fig. 3-12(a).

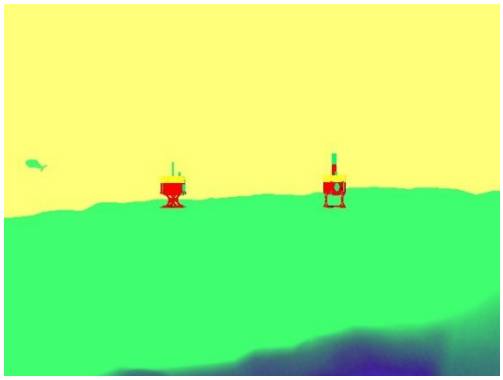
(2) CISCREA-*h* is located at the x axis of Blender frame (the frame defined by Blender), and its orientation is along the positive direction, origin point is set as $(-2m, 0, 10m)$, which remains unchanged.

(3) The orientation of CISCREA-*i* is along the positive direction of y axis of Blender frame and remains unchanged. The origin points of CISCREA-*i* and CISCREA-*j* are set as $(0, 0.5m, 10m)$, $(0, -0.5m, 10m)$ respectively.

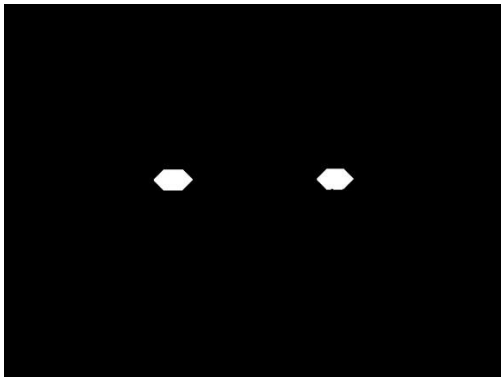
(4) In the sample range $d \in [2m, 7m]$, CISCREA-*i* and CISCREA-*j* synchronously move with the sample interval $(0.25m, 0.125m, 0m)$, $(-0.25m, -0.125m, 0m)$ respectively. CISCREA-*h* keeps static and samples data at every position of CISCREA-*i* and CISCREA-*j*. Fig. 3-12 presents one sampling process.

(5) To make the sampled data accurate, CISCREA-*h* captures 100 times repeatedly at every position of CISCREA-*i* and CISCREA-*j* and estimate the heights of yellow part in each image, the average value of the heights, $h_{hi} = \frac{1}{100}\sum_{i=1}^{100} h_{hi}$ and $h_{hj} = \frac{1}{100}\sum_{j=1}^{100} h_{hj}$, is regarded as the height at this relative distance.

(a) One sampled RGB image



(b) The HSV image turned from (a)



(c) The yellow region segmented in HSV space

(d) Identification of local information

Fig. 3-12 One sampling presentation of the relative distance model calibration process

All the sampling data are imported into Matlab, and the parameters of the distance model can be obtained by fitting curves using the "cftool" (Curve Fitting Toolbox). The calibration results are shown in equation (3-21) and Fig. 3-13.

$$d_{ij} = 95.73e^{-0.1154h} + 3.091e^{-0.007636h} \quad (3\text{-}21)$$

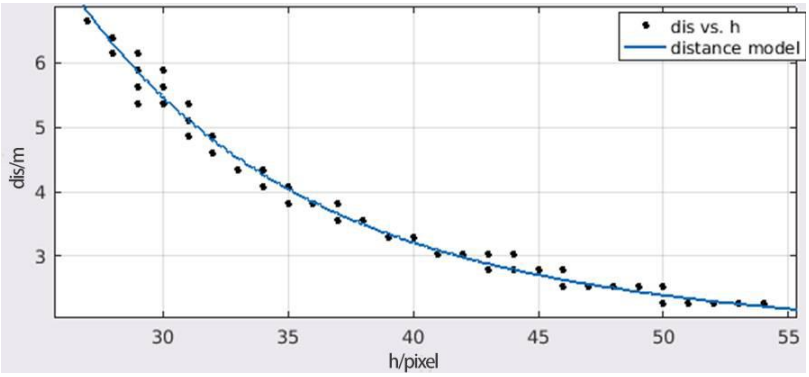


Fig. 3-13 The fitted curve of relative distance model

To evaluate the accuracy of the calibrated model, R-square goodness of fit tests are performed on simulated data, whose values are located in the interval [0,1]. The larger value of R-square is, the more accurate of the fitted model will be. For the fitted distance model, the R-square value is 0.9903, which means the fitted results are accurate and acceptable.

2) Calibration of the relative angle model

Similar to the calibration of the relative distance model, first three CISCREAs are used to sample data: relative angles φ_{ij} and u positions of detected AUVs in images. The sample interval is $\varphi \in [-30^{\circ}, 30^{\circ}]$, the sample step is $\varphi_{si} = \pm 1.5^{\circ}$. Similar to the calibration of relative distance model, 100 images are captured at each position of the angle φ_i, and the $\hat{u}_{ij}$ corresponding to each image is estimated, and then the average value of all the $\hat{u}_{ij}$ of this position is regarded as the real position of AUV, $u_{ij} = \frac{1}{100}\sum_{j=1}^{100}\hat{u}_{ij}$. Then the angle model is also calibrated by using the cftool of Matlab, the calibrated results are shown in equation (3-22) and Fig. 3-14.

$$\varphi_{ij} = 0.1087 u_{ij} - 36.77 \tag{3-22}$$

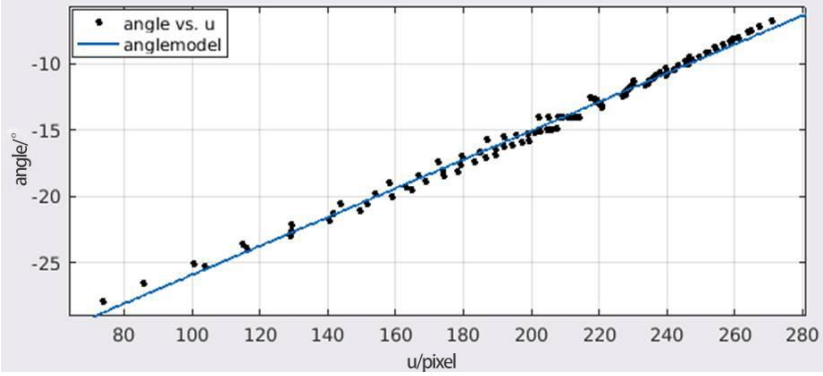


Fig. 3-14 The fitted curve of angle model

The R-square value of relative angle model is 0.9923, it implies that the accuracy of the relative angle model is also acceptable.

Remark: The relative distance and angle models are calibrated in an ideal simulation environment. In the real underwater environment, these models should be improved by taking the light scattering factors into account according to the categories of water body.

Subsequently, the relative position of AUV-j $p_{ij}(x_{ij}, y_{ij})$ in local frame $o_{li}x_{li}y_{li}$ can be computed by using the relative distance d_{ij}, relative angle φ_{ij} and the global orientation of AUV-i θ_{ij} estimated by compass when AUV-i detects AUV-j, shown in equation (3-23).

$$\begin{pmatrix} x_{ij} \\ y_{ij} \end{pmatrix} = \begin{pmatrix} \sin\theta_{ij} & \cos\theta_{ij} \\ -\cos\theta_{ij} & \sin\theta_{ij} \end{pmatrix} \begin{pmatrix} d_{ij}\sin\varphi_{ij} \\ d_{ij}\cos\varphi_{ij} \end{pmatrix} \tag{3-23}$$

3.2.3 Identification of ID and information exchange

As mentioned in section 3.2.1, AUV's ID can be an explicit one or an implicit one. The explicit ID can be identified by image processing, while the implicit ID is identified by the proposed

neighbor-check mechanism after sharing the information among AUVs.

1. Identification of the explicit ID

The identification of an explicit ID is not complex. After identifying the AUV and computing its position successfully in the image, we decode the ID around the position of AUV $(u_{ij}[k_i], v_{ij}[k_i])$ in the origin RGB image and decode it. Firstly, searching a certain area from $(u_{ij}[k_i], v_{ij}[k_i])$ to the top of the green area (see the green area in Fig. 3-8(a)), then the code of the ID can be got according to the detected order of black and white colors. For example, the detected orders of black and white colors of Fig. 3-8(a) are "black, black, white and black", so that the code is "0010", which means the ID is 2.

The information structure of the information exchange among AUVs based on explicit IDs at the beginning of the building process is shown below:

{ID of AUV-i itself: i;

Number of the detected neighbors: N_i;

IDs of the neighbors: $j_1, j_2, \ldots, j_{Ni}$;

The relative positions of neighbors got by image processing: $\mathbf{p}_{ij1}$, $\mathbf{p}_{ij2}$, …, $\mathbf{p}_{ijN_i}$, where $\mathbf{p}_{ijk} = (x_{ijk}, y_{ijk})$

}

2. Identification of implicit ID

The identification of an implicit ID is relatively complex. Each ID is written in program, then the ID cannot be identified by external means. To identify the IDs of neighbors, one neighbor-check mechanism is proposed. Firstly, the AUVs should share their neighbors' information by acoustic communication, in this case, the IDs of neighbors are unknown at this stage:

{ID of AUV-i itself: i;

Number of the found neighbors: N_i;

The relative positions of neighbors got by image processing: $\mathbf{p}_{ij1}$, $\mathbf{p}_{ij2}$, …, $\mathbf{p}_{ijN_i}$

}

Because all the local frames local-i, $o_{li}x_{li}y_{li}$, are parallel (see section 3.2.2), if AUV-i and AUV-j can be detected by each other, the vector sum of the coordinate of origin point $o_{li}(x_{ji}, y_{ji})$ in the frame $o_{lj}x_{lj}y_{lj}$ and the coordinate of origin point $o_{lj}(x_{ij}, y_{ij})$ in the frame $o_{li}x_{li}y_{li}$ is vector $\mathbf{0}$, shown in Fig. 3-15. But due to the image processing errors and movement control errors, the vector sum should meet the condition:

$$\begin{bmatrix} -\varepsilon \\ -\varepsilon \\ -\varepsilon \end{bmatrix} < \begin{bmatrix} \varepsilon_{xij} \\ \varepsilon_{yij} \\ \varepsilon_{dij} \end{bmatrix} = \begin{bmatrix} x_{ij} \\ y_{ij} \\ d_{ij} \end{bmatrix} + \begin{bmatrix} x_{ji} \\ y_{ji} \\ -d_{ji} \end{bmatrix} < \begin{bmatrix} \varepsilon \\ \varepsilon \\ \varepsilon \end{bmatrix} \quad (3\text{-}24)$$

Where $d_{ij} = \left\| O_{ij} \right\| = \sqrt{x_{ij}^2 + y_{ij}^2}$.

According to the given threshold ε , the real neighbors (the AUVs detected correctly) and their IDs can be determined according to **Algorithm 1**. If there are multiple relative positions (*m* is the number of positions) meeting the equation (3-24), such as in the case when more than one neighbors within a small area, the neighbor with minimum distance error ($\varepsilon_{d_{\min}} = \min\{\varepsilon_{d_{ij}}\}$, $j \in 0,1,...,m$) is the real neighbor. As a result, if one neighbor of AUV-*i* and one neighbor of AUV-*j* can find each other, their local information meets equation (3-24), and the distance error ε_{dij} is the minimum value $\varepsilon_{d_{\min}}$, then this neighbor's ID of AUV-*i* is *j*, while the corresponding neighbor's ID of AUV-*j* is *i*. And AUV-*i* and AUV-*j* are called a neighbor pair, the detail of the neighbor check mechanism is presented in **Algorithm 1**.

Algorithm 1 Identification of AUVs with implicit ID (neighbor check mechanism)

for *i* in all AUVs: **do**
  for *j* in all the neighbors of AUV-*i*: **do**
    for *k* in all the AUVs: **do**
      if *k*!=*i*: t**hen**
        f**or** *t* in all the neighbors of AUV-*k*: **do**
          if the elements (ε_{xij} , ε_{yij} , ε_{dij}) of vector sum meet the equation (3-24), and ε_{dij} is the minimum value: **then**
            the ID of AUV-*i*'s neighbor-*k* is *j*
            the ID of AUV-*k*'s neighbor-*t* is *i*
          end if
        end for
      end if
    end for
  end for
end for

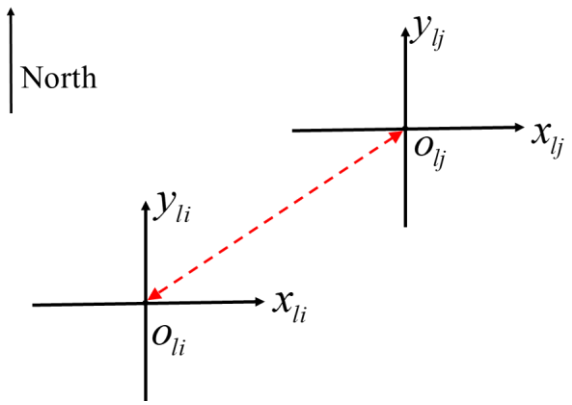


Fig. 3-15 The relative positions between the origin points of local frames $o_{li}x_{li}y_{li}$ and $o_{lj}x_{lj}y_{lj}$.

Ideally, The vector sum of the coordinate of origin point $o_{li}(x_{ji}, y_{ji})$ in the frame $o_{lj}x_{lj}y_{lj}$ and the coordinate of origin point $o_{lj}(x_{ij}, y_{ij})$ in the frame $o_{li}x_{li}y_{li}$ is vector $\mathbf{0}$.

3.3 Chapter summary

This chapter introduces a method of recognizing AUV units from digital images and extracting their local information by an underwater image processing method. Firstly, the defogging algorithm considering wavelength compensation is introduced to enhance the underwater digital image after presenting several common underwater image pre-processing methods. Then the quality of digital image is improved, which verifies the feasibility of using underwater camera to obtain information in a certain range. Subsequently, the CISCREA is recognized, and its size and position in the image are also estimated from the enhanced image. According to the camera model, we construct a relative distance model and a relative angle model. Depending on these two models, the relative distance and angle can be computed from a single image based on the extracted size and position of CISCREA in the image and then the relative positions can be computed. Further, the identification methods of the explicit and implicit IDs are given to determine the IDs. In a word, the neighbors' IDs and their relative positions can be obtained by the image processing.

4. Local position-based control method

To achieve the coordinated control of multi-AUV systems, the location determination and update of AUVs in the limited communication conditions is one key problem. In this thesis, we only consider the situation, in which on-board cameras and compasses are the main sensors to sense the surroundings and get the global relative positions among AUVs. And the local information cannot update on-line because of the limited view field of on-board cameras. Then the most methods mentioned in section 1.2 cannot be considered, as they need global information or need communication to update information in real-time.

Considering that the initial distribution of AUVs can be accessed by sharing neighbors' information obtained by cameras (see chapter 3), the AUVs can achieve the formation building if they can avoid collisions in the process. To avoid collisions without trajectory update in real-time, we can plan non-intercrossing trajectories for all the AUVs according to the initial distributions in the obstacle-free environment.

Based on this idea, we propose this local position-based control method, whose core is the proposed collision avoidance strategy with time complexity$O\ (n\log n)$. In this method, a common frame is established with the shared local information, and it is regarded as the global frame. Then the initial distribution of AUVs in this common frame can be obtained. Subsequently, the distribution of the pyramid pattern is optimized to make it tend to the initial distribution according to the proposed collision avoidance strategy, which achieves the planning of the non-intercrossing straight trajectories. Finally, the AUVs move to their destinations with a simple kinematic control and achieve the formation building.

Remark: to reduce the communication difficulty, the AUVs do not communicate with each other any more after the complement of sharing local information.

4.1 Establishment of the common frame

Based on section 3.2, each AUV can get all the neighbors information by information exchange. Because of the parallel local frames $o_{lj}x_{lj}y_{lj}$, there is only translation relationship between these local frames. Since the digraph composed by all the AUVs can be considered strongly connected with our hypothesises (see chapter 2), all the positions can be translated into one frame by Breadth-First Search (BFS). Consequently, when one local frame is chosen as the common frame, all the positions of AUVs can be translated into the common frame, and the initial distribution can be obtained. To keep the digraph strongly connected, each AUV will determine the common frame depending on the **Condition 4-1**.

Condition 4-1

1. If each AUV finds at least one neighbors, the local frame $o_{ln}x_{ln}y_{ln}$ of the AUV with maximum neighbors will be regarded as the common frame $o_cx_cy_c$ (see Fig. 4-3), otherwise, all the AUVs should execute the 2nd step.

2. If at least one AUV does not find any neighbors, these AUVs should slightly move their positions and re-search neighbors again. Then all the AUVs share again the neighbors' information and repeat 1st step.

After comparison, the local frame $o_{ln}x_{ln}y_{ln}$ of AUV-*n* with maximum neighbors is chosen as the common frame. Then, as said above, all the positions of AUVs will be transformed into this common frame according to BFS. For example, p_{ni} is the relative position of AUV-*i* in $o_{ln}x_{ln}y_{ln}$. After $o_{ln}x_{ln}y_{ln}$ is seen as the common frame, p_{ni} is re-noted as (x_{ci}, y_{ci}). After all the positions of neighbors AUV-*i* of AUV-*n* ($i = 1, 2, ...N_n$, N_n is the number of neighbors of AUV-*n*) are translated into the common frame, the positions of neighbors AUV-*j* ($j = 1, 2, ...N_i$) of AUV-*i* are translated into $o_cx_cy_c$ gradually. In other words, for each neighbor of AUV-*i*, its position in the common frame (x_{cj}, y_{cj}) can be computed by equation (4-1).

According to the hypothetical condition (2) $V_I = \bigcup_{i=1}^{N} V_{Ii}$ (see section 2.3) and condition 4-1, the digraph G composed by the initial distribution is strongly connected. As a result, all the positions of AUVs can be determined in the common frame and the initial distribution can be obtained.

$$\begin{pmatrix} x_{cj} \\ y_{cj} \end{pmatrix} = \begin{pmatrix} x_{ij} \\ y_{ij} \end{pmatrix} + \begin{pmatrix} x_{ci} \\ y_{ci} \end{pmatrix} \tag{4-1}$$

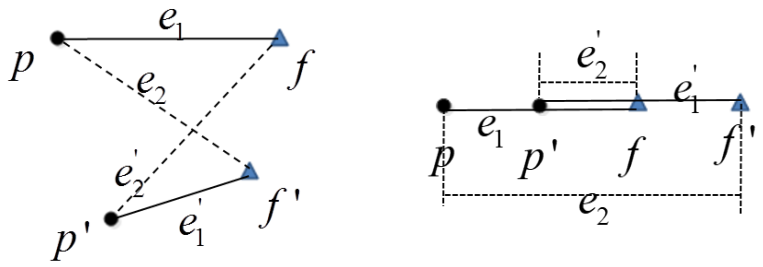
4.2 Collision avoidance strategy

Based on the initial positions of AUVs in the common frame, a collision avoidance strategy should be put forward to identify the pyramid distribution and to make it trend to the initial distribution, and then to plan the trajectories without collision. Before giving its designation, we present a lemma 4-1 of graph theory.

Since the net constituted by the initial distribution is regarded as a digraph G, the points p, p' are used to represent the initial position of AUV-*i* and AUV-*j*, while the points f, f' stand for the pyramid positions of AUV-*i* and AUV-*j*.

Lemma 4-1[121]: Let $e_1 = (p, f)$ and $e_1' = (p', f')$ be two edges in the graph G, let $e_2 = (p, f')$ and $e_2' = (p', f)$ be the other two edges in the graph G, and assume that p,

p', f and f' are distinct, shown in Fig 4-1. Then e_1 and e_1' never cross, although they can overlap ($e_1 \text{ I } e_1' \neq \Phi$, see Fig. 4-1(b)), if $d_{e_1} + d_{e_1'} < d_{e_2} + d_{e_2'}$ and $\max(d_{e_1}, d_{e_1'}) < \max(d_{e_2}, d_{e_2'})$, where d_{e_1}, $d_{e_1'}$, d_{e_2} and $d_{e_2'}$ are the lengthes of corresponding edges respectively.



(a) (b)

Fig. 4-1 Illustration of lemma 4-1. If the points p and p' choose the short distances as the edges, the edges will be never inter-crossing. (a) The general situation. (b) The situation with all the points are aligned. In this case, $d_{e_1} + d_{e_1'} = d_{e_2} + d_{e_2'}$, but if $\max(d_{e_1}, d_{e_1'}) < \max(d_{e_2}, d_{e_2'})$, we regard the e_1 and e_1' are not inter-crossing.

After the pyramid distribution is determined and optimized, the AUVs will never collide with each other, if the positions in both distributions match up according to lemma 4-1, which means that the sum of all the distances between matched-up positions will be the smallest, expressed by equation (4-2). And the matched-up positions make up one sub-group, called combined sub-group, for example, the positions p and f make up one combined sub-group.

$$w(G) = \arg\min\left(\sum_{p,f \in G} dis(p, f)\right) \quad (4\text{-}2)$$

Where, $dis(p, f)$ is the relative distance between the initial position p and its final position f, and both p and f participate to constitute one and only one edge.

Considering that getting $w(G)$ by the mathematical method is complex$O(n!)$, a collision avoidance strategy with complexity $O(n \log n)$ is proposed to determine the optimized pyramid distribution and plan trajectories according to this lemma 4-1.

4.2.1 Determination and optimization of pyramid distribution

1. Determination of the optimized pyramid distribution

Similar to literature [123], to avoid collision, the whole area including all the AUVs is divided into several sub-areas according to the initial distribution of AUVs. Considering that the pyramid pattern is symmetrical about the principal axis, this principal axis should also separate the initial

distribution into two sets L_I and R_I as evenly as possible to make the pyramid distribution tend to the initial distribution, which means the numbers of elements in both sets N_{L_I} and N_{R_I} should meet the condition: $\left|N_{L_I} - N_{R_I}\right| \le 1$, shown in Fig. 4-4. To do so, the principal axis of pyramid pattern should be determined correctly. Since the eigenvectors of the covariance matrix of positions have the ability to evenly divide the positions, and the circle pattern is also a symmetrical pattern, two ways are given to identify the pyramid orientation: (1) based on the principal component analysis (PCA) [124]; (2) based on an intermediate formation (circular pattern, CP). Subsequently, a pyramid frame $o_p x_p y_p$ is created: the principal axis is regarded as y axis, the average position of all the positions of AUVs is seen as the origin point, and the x axis is identified according to the right-hand rule. During this step, the position of leader will be identified by these two methods presented below.

(1) Identification of the principal axis of pyramid pattern and the position of leader based on PCA

In this method, the principal axis of pyramid pattern is determined first, the position of leader is computed later. The principal axis is the eigenvector associated to the larger eigenvalue of the covariance matrix of all the positions of robots in the common frame (shown in equation (4-3)).

$$Cov = \begin{pmatrix} \frac{\sum_{i=1}^{N} x_{c_i}^2}{N} - x_{op}^2 & \frac{\sum_{i=1}^{N} x_{c_i} y_{c_i}}{N} - x_{op} y_{op} \\ \frac{\sum_{i=1}^{N} x_{c_i} y_{c_i}}{N} - x_{op} y_{op} & \frac{\sum_{i=1}^{N} y_{c_i}^2}{N} - y_{op}^2 \end{pmatrix} \quad (4\text{-}3)$$

In which, (x_{op}, y_{op}) is the coordinate of the origin point of pyramid frame in the common frame, N is the number of AUVs. And $x_{op} = \frac{\sum_{i=1}^{N} x_{c_i}}{N}$, $y_{op} = \frac{\sum_{i=1}^{N} y_{c_i}}{N}$.

The position of leader of the pyramid pattern is $(0, y_L)$. y_L is determined by the AUV with the minimum *y* value (such as the AUV-2 in Fig. 4-2).

$$y_L = y_{\min} + ly_{\max} d_s \cos\varphi_s \quad (4\text{-}4)$$

Where, $ly_{\max}$ is computed by equation (4-5) with $od_{auv} = N$, d_s and φ_s are the expected relative distance and angle between each AUV and its parent.

The AUV with the shortest distance to the leader position of pyramid distribution is selected as the real leader. As an example, the leader in Fig. 4-2 would be the AUV-6.

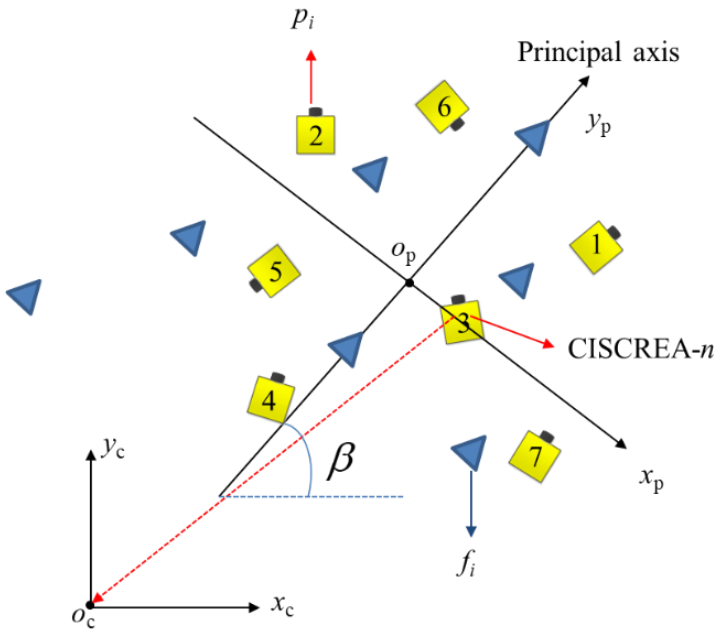


Fig. 4-2 Pyramid pattern built based on the PCA. $o_c x_c y_c$ and $o_p x_p y_p$ are the common frame and pyramid frame respectively. CISCREA-*n* is assumed to be the robot with the maximum number of neighbors, $o_c x_c y_c$ is the local frame of CISCREA-*n*. Here, $o_c x_c y_c$ only presents the orientation and its origin point is the position of CISCREA-*n*, while $o_p x_p y_p$ is located at the correct position, its y axis $o_p y_p$ is the principal axis, origin point o_p is the average value of all the positions in the $o_c x_c y_c$. β is the angle between the axis $o_p y_p$ the and the axis $o_c x_c$.

(2) Identification of the principal axis of pyramid pattern and position of leader based on CP

In this approach, the leader position of pyramid distribution is determined first. All the robots will constitute a circle pattern with one robot as the center and the others located on the circumference. To achieve the circle pattern, the average position of all the positions in the initial distribution is regarded as the center position. And the AUV-*n* with the maximum neighbors, whose local frame is chosen as the common frame, moves to the average position to be the circle center.

Then to calculate the positions on the circumference, the radius of the circle is determined according to d_s and the number of robots: $r_c = \max\left\{d_s, d_s/(2\sin(\frac{\pi}{N-1}))\right\}$, and all the positions on the circumference should be distributed evenly. First, we fix one position $\mathbf{p}_0$ located on the axis $o_c y_c$ on the circumference with position $(0, r_c)$, then the other positions on the circumference can be computed.

Subsequently, except for AUV-*n*, the other AUVs determine their positions on the circumference according to the PCA method to avoid collision during the building of the circular

pattern. Once the AUVs achieve the circular pattern, each position of the AUV on the circumference can be regarded as the leader, and the direction from the center position to the leader position is the orientation of principal axis, shown in Fig. 4-3. Generally, the AUV with position $\mathbf{p}_0(0, r_c)$ is the leader by default, such as the AUV-1 in Fig. 4-3. If we choose the defaulted leader, the common frame $o_c x_c y_c$ and the pyramid frame $o_p x_p y_p$ are coincided.

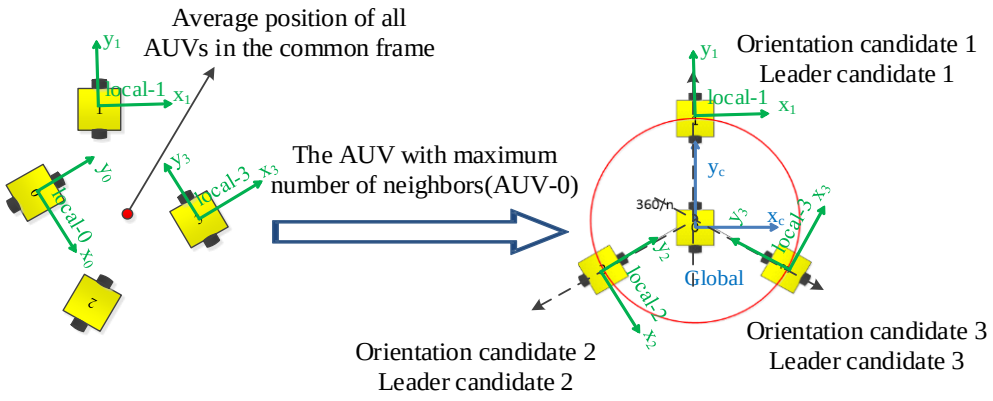


Fig. 4-3 Pyramid pattern built based on the CP. The local-0 with maximum neighbor is regarded as the common frame and also as the circle center. Each AUV on the circumference (AUV-1, AUV-2, and AUV-3) can be considered as the leader, the orientation from the circle center to the leader is the orientation of the pyramid pattern. In this case, AUV-1 is the default leader, and the common frame $o_c x_c y_c$ and the pyramid frame $o_p x_p y_p$ are coincided.

2. Identification of the pyramid distribution

After the identification of the principal axis and leader position of the pyramid pattern, the other positions in the pyramid distribution can be computed according to the definition of pyramid pattern, the expected relative distance d_s and relative angle φ_s between the parent and child AUVs by BFS (see section 2.3). Then the pyramid distribution is obtained. The key issue is to compute the order of parent according to the order of child, shown in equations (4-5) and (4-6). Then the child position (x_{ch}, y_{ch}) in the pyramid frame can be calculated by equation (4-7), where "$\pm$" is decided by the relative position relationship between parent and child AUVs (see the definition of pyramid pattern in section 2.3).

$$ly_{auv} = \left\lfloor \frac{\sqrt{1+8od_{auv}}+1}{2} \right\rfloor \quad (4\text{-}5)$$

Where, od_{auv} and ly_{auv} are the order and layer number of this AUV in the pyramid distribution.

$$od_{pa} = \begin{cases} od_{ch} - ly_{ch}, \text{if the child is located at the last position of each layer} \\ od_{ch} - (ly_{ch} - 1), \text{if the child is located at the other positions of each layer} \end{cases} \quad (4\text{-}6)$$

Where, od_{pa} and od_{ch} are the orders of parent and child respectively in the pyramid distribution, ly_{ch} is the layer number of the child.

$$\begin{cases} x_{ch} = x_{pa} + d_s \sin(\pm\varphi_s) \\ y_{ch} = y_{pa} - d_s \cos(\pm\varphi_s) \end{cases} \quad (4\text{-}7)$$

In this section, we obtain the pyramid distribution according to the initial distribution, in the next section, the non-intercrossing straight trajectories will be planned.

4.2.2 Non-intercrossing straight trajectories planning

The trajectory planning of AUVs is done in pyramid frame $o_p x_p y_p$. Thus, the initial distribution is transformed into the pyramid frame by equation (4-8).

$$\begin{pmatrix} x_f \\ y_f \end{pmatrix} = \begin{pmatrix} \sin\beta & -\cos\beta \\ \cos\beta & \sin\beta \end{pmatrix} \begin{pmatrix} x_c - x_{op} \\ y_c - y_{op} \end{pmatrix} \quad (4\text{-}8)$$

Where, β is the angle between the y axis of pyramid frame and the x axis of common frame, (x_f, y_f) and (x_c, y_c) are the original positions in the pyramid frame and common frame respectively. Fig. 4-2 shows an example.

The trajectory planning explains explicitly how robots in the positions $p_{ci}(x_{ci}, y_{ci})$ will move to positions $p_{fi}(x_{fi}, y_{fi})$ according to lemma 4-1. The key problem is to equally separate the regions including both kinds of distributions to generate the combined sub-groups meeting the conditions of lemma 4-1. Then in each combined sub-group, the positions ($p_{ci}(x_{ci}, y_{ci})$ and $p_{fi}(x_{fi}, y_{fi})$) will match up according to lemma 4-1.

To optimize the positions assignment in both distributions appropriately to make the generated combined sub-groups disjoint as far as possible, the positions should be sorted according to their "y" values descendingly and their "x" values ascendingly, and also should be divided into sub-groups. Each sub-group includes at most two positions with one marked order, such as L_{I1}, R_{I1}, L_{P1}, R_{P1}, shown in Fig. 4-4(a) and 4-4(b). The sub-groups with the same orders in both distributions (such as L_{I1}, L_{P1}) constitute one combined sub-group, shown in Fig. 4-4(c). In each combined sub-group, the destination of each AUV is determined according to lemma 4-1. Then the matched-up positions in the pyramid distribution are the destinations of AUVs, and the IDs of AUVs and the orders of their corresponding destinations make up the expected positions relationship. The edges of the matched-up positions are non-intercrossing and straight, which will be proofed below

following the presentation of the detailed collision avoidance strategy.

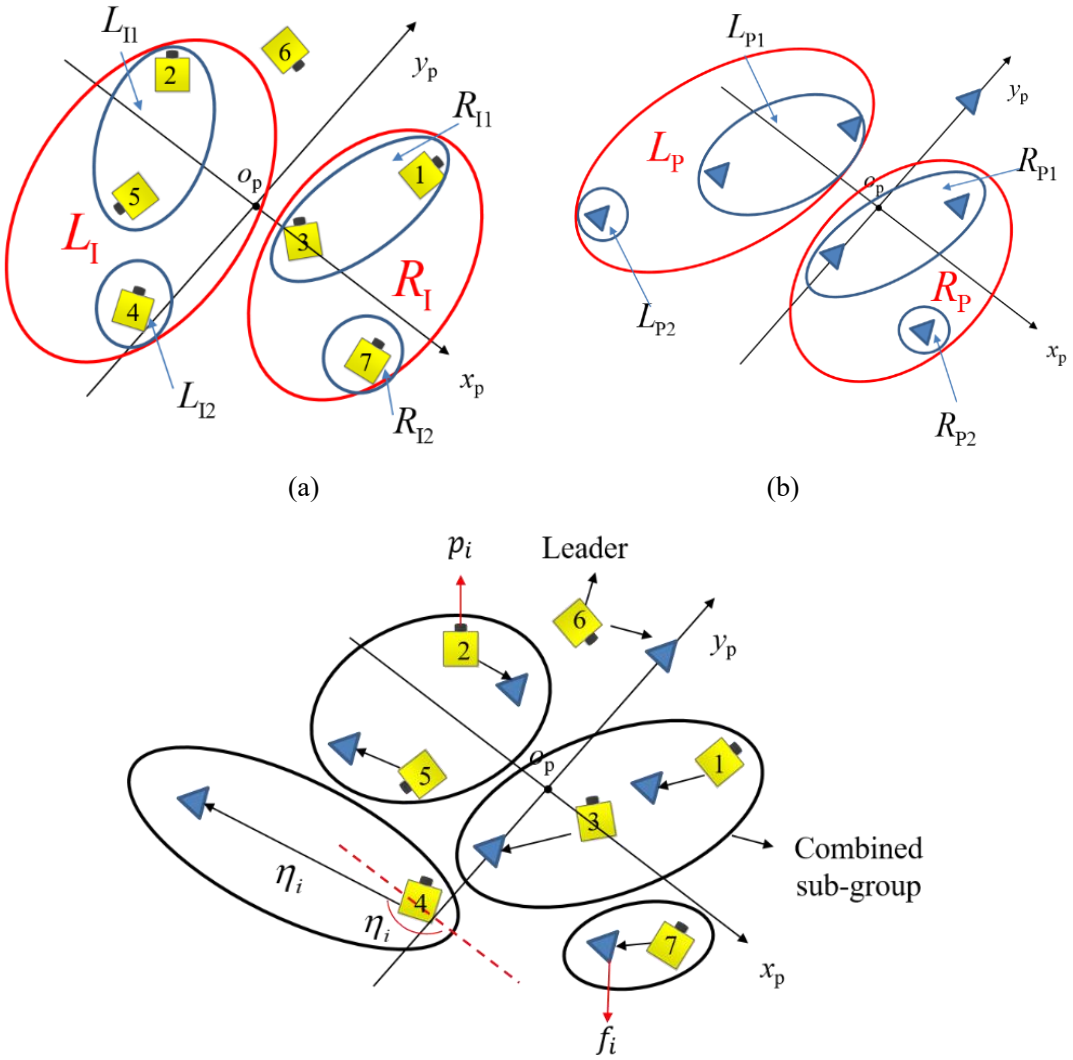


Fig. 4-4 One example of the generation of straight and non-intercrossing trajectories in the pyramid pattern built by PCA. Except the leader, the positions in the initial and pyramid distributions (Fig. 4-4(a) and (b)) are divided into sub-groups with one or two positions respectively according to their subscriptions ordered by their coordinates. The sub-groups with the same order in both distributions are combined together to be the combined sub-groups, where the positions match up depending on lemma 4-1 to generate the straight and non-intercrossing trajectories, noted as (η_i, d_i), Fig. 4-4(c).

The detail of the collision avoidance strategy is shown below:

(1) In the PCA method, the leader is the AUV with shortest distance to the first position of pyramid pattern, then its trajectory is planned well, which is the edge connecting its initial position and this first position in the pyramid pattern. In the CP method, the position of leader in both distribution is the same, so that the leader does not need movement. In the subsequent steps, the leader position is not included. And these steps are the same for PCA and CP method. Here we take

PCA as an example.

(2) The initial distribution $\{p_{ci}(x_{ci}, y_{ci})\}$ is divided into two sets located at both sides of the principal axis of pyramid pattern, such as the two solid ellipses drawn in Fig. 4-4(a): the left set L_I and right set R_I. The number of the positions in both two sets (N_{L_I} and N_{R_I}) should meet $|N_{L_I} - N_{R_I}| \le 1$, otherwise, the positions near the principal axis should be re-assigned until meeting $|N_{L_I} - N_{R_I}| \le 1$. Subsequently, all the positions are ranked in descending order of y_{ci} in each set.

(3) Similarly, the pyramid distribution $\{p_{fi}(x_{fi}, y_{fi})\}$ is also separated into two sets: left set L_P and right set R_P, presented in Fig. 4-4(b). The positions located on the y axis are added to L_P or R_P according to the number of elements in the sets L_I and R_I of initial distributions. As the similar manner with step (2), all the positions in each set L_P and R_P are sorted in descending order of y_{fi} values.

(4) In the set (L_P or R_P), the positions $p_{fi}, p_{fi+1}, ..., p_{fi+j}$ with the same y value ($y_{fi} = y_{fi+1} = ... = y_{fi+j}$) are re-ordered again according to the x_{fi} values in ascending order. At the same time, the corresponding positions in the set (L_I or R_I) with the same subscripts, $p_{ci}, p_{ci+1}, ..., p_{ci+j}$, are also re-ordered in ascending order of x_{ci} values.

(5) Similar to step (4), if the errors of some y_{ci} values in the set L_I or R_I are within a small range ε, $|y_{ci} - y_{ci+k}| < \varepsilon$, $k = 1, ..., m$, the positions need to be re-ordered in ascending order of x_{ci} values; correspondingly, the positions in the pyramid distribution $p_{fi}, p_{fi+1}, ..., p_{fi+m}$ also need to be re-ordered in the same manner. Besides, ε can be taken as: $(d_{\max} - d_s)(1 - \cos\varphi_s) < \varepsilon < d_s \cos\varphi_s$, shown in Fig. 4-6(c), where $d_{\max}$ is the largest distance between the positions in both initial and pyramid distributions.

(6) The eventually re-ordered sets $(L_I, R_I), (L_P, R_P)$ are further separated into sub-groups, which include one or two positions, noted as $\{\{L_{I1}, L_{I2}, ...\}, \{R_{I1}, R_{I2}, ...\}\}$, and $\{\{L_{P1}, L_{P2}, ...\}, \{R_{P1}, R_{P2}, ...\}\}$, such as the blue ellipses drawn in Fig. 4-4(a) and 4-4(b). The sub-groups with the same orders in both distributions are combined together to form a combined sub-group, shown in Fig. 4-4(c).

(7) In each combined sub-group, the non-intercrossing straight trajectories and the expected positions relationship (the IDs of robots and their corresponding orders in the pyramid distribution) are obtained according to lemma 4-1.

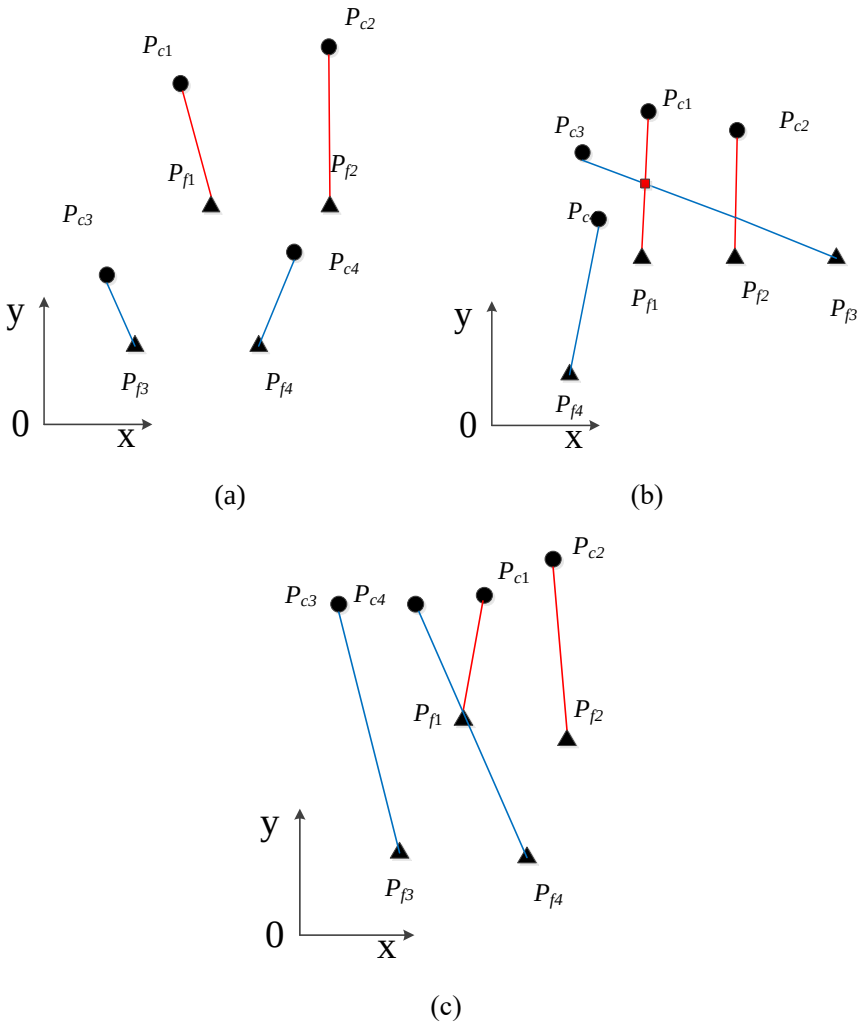


Fig. 4-5 Analysis of the collisions among AUVs (a), (b), (c) show three different potential collision situations respectively. (a) No inter-crossing area exists between two adjacent combined sub-groups, in this case, no collision will happen. (b), (c) Two adjacent combined sub-groups are inter-crossing: (b) the trajectories are inter-crossing; (c) the destination of one robot-1 is on the trajectory of another robot-4, and robot-1 arrives at its destination earlier than robot-4.

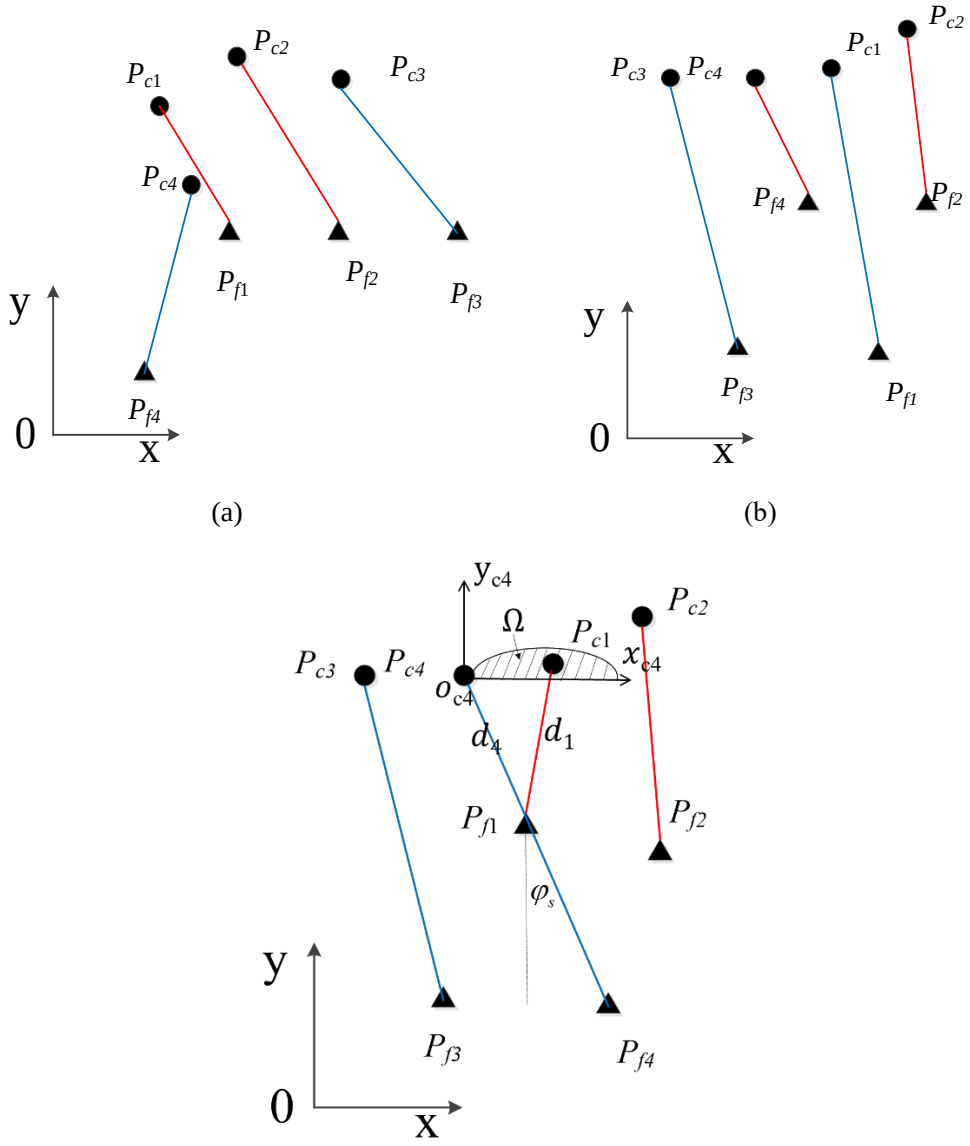


Fig. 4-6 The actual match-up results associated with Fig. 4-5(b) and (c). Actually, situations shown in Fig. 4-5(b) and (c) cannot occur because of the adjustment according to the x values in steps (4) and (5). (a), (b) present the actual match-up results, while (c) gives the areas of positions meeting

$$\left|y_{ci} - y_{ci+k}\right| < \varepsilon .$$

Now, we give the proof that with the collision avoidance strategy, the robots will not collide with each other when they move to build a pyramid pattern in the obstacle-free environment.

Proof of collision avoidance strategy:

(1) In each combined sub-group (Fig. 4-4(c)), the AUVs will never collide with each other according to lemma 4-1.

(2) For two adjacent sub-groups, $\left[p_{c1}, p_{c2}\right]$, $\left[p_{c3}, p_{c4}\right]$ $(\min(y_{c1}, y_{c2}) \geq \max(y_{c3}, y_{c4}))$ and $\left[p_{f1}, p_{f2}\right]$, $\left[p_{f3}, p_{f4}\right]$ $(\min(y_{f1}, y_{f2}) \geq \max(y_{f3}, y_{f4}))$, $\left[p_{c1}, p_{c2}, p_{f1}, p_{f2}\right]$ and

$\left[p_{c3}, p_{c4}, p_{f3}, p_{f4}\right]$ are composed to combined sub-groups respectively according to step (6). Two situations need to be discussed:

① $\min(y_{f1}, y_{f2}) > \max(y_{c3}, y_{c4})$: in this case, two combined sub-groups $\left[p_{c1}, p_{c2}, p_{f1}, p_{f2}\right]$ and $\left[p_{c3}, p_{c4}, p_{f3}, p_{f4}\right]$ have no inter-crossing area, the AUVs in positions $\left[p_{c1}, p_{c2}\right]$ (and $\left[p_{c3}, p_{c4}\right]$) can reach positions $\left[p_{f1}, p_{f2}\right]$ (and $\left[p_{f3}, p_{f4}\right]$) respectively without any collision (see Fig. 4-5(a)).

② $y_{c3} > \max(y_{f1}, y_{f2})$ and/or $y_{c4} > \max(y_{f1}, y_{f2})$: in this case, there would be collisions: (a) if two AUVs belonging to two combined subsets can arrive at the same position at the same time (Fig. 4-5(b)), or (b) if the final position of one AUV is on the path of another AUV (Fig. 4-5(c)).

For the situation (a), p_{f1}, p_{f2}, p_{f3} and p_{c1}, p_{c2}, p_{c3} will be re-ordered according to their x values like step (4), thus the combined sub-groups will be re-made up. It means that the situation in Fig. 4-5(b) are avoided by step (4) and it will never happen, the improved match-up result of Fig. 4-5(b) is shown in Fig. 4-6(a).

If the situation of Fig. 4-5(c) exists (like the type shown in Fig. 4-6(c)), robot-1 with position p_{c1} should arrives at its destination p_{f1} earlier than robot-4 with position p_{c4}, then the positions p_{c1} and p_{c4} must meet $y_{c1} - y_{c4} \geq 0$ and $d_1 \leq d_4$, then we get that $0 \leq (y_{c1} - y_{c4}) \leq d_4(1-\cos\varphi_s)$, and p_{c1} is located in the area Ω, shown in Fig. 4-6(c). Equation (4-9) describes the Ω in the frame $o_{c4}x_{c4}y_{c4}$. As a result, when we set $\varepsilon > d_4(1-\cos\varphi_s)$, the orders of p_{c1}, p_{c4}, and p_{f1}, p_{f4} will be re-sorted according to their "x" values. Because of $\left(d_{\max} - d_s\right) \geq d_4$ ($d_{\max} \geq \left\|o_{c4}P_{f4}\right\|, d_s = \left\|P_{f1}P_{f4}\right\|$), without the loss of generality, the AUVs will not collide with each other if the threshold value of step (5) is set as $\varepsilon > \left(d_{\max} - d_s\right)(1-\cos\varphi_s)$. The improved result is shown in Fig. 4-6(b).

$$\Omega = \int_0^t \int_0^q [(x - d_4\sin\varphi_s)^2 + (y + d_4\cos\varphi_s)^2 - d_4^2]dxdy \qquad (4\text{-}9)$$

Where, $t = d_4(1-\cos\varphi_s)$, $q = 2d_4\sin\varphi_s$.

Based on the proof above, the proposed collision avoidance strategy can prevent collision successfully in obstacle-free situations. Then we give the proof of the time complexity of collision avoidance strategy.

Proof of the time complexity $O(n\log n)$ of collision avoidance strategy:

(1) In step (2) of the algorithm, the orientation of the principal axis is determined by computing the eigenvector of covariance matrix (equation (4-3)), its time complexity $O(f_{11})$ is known as

$O(f_{11}) = O(1)$. The position fine-tuning needs to count the numbers of elements in both left and right sets, the time complexity of worst situation $O(f_{12})$ is estimated as $O(f_{12}) \le O(n)$.

(2) In steps (2)-(5), the positions in both sets are sorted by using the function (sort()) of Python respectively. The function is "Timesort", the time complexity of worst situation $O(f_2)$ is is $O(f_2) = O(n \log n)$.

(3) In step (6), the positions in both sets are separated into sub-groups with one or two positions according to their orders, with the time complexity $O(f_3) = O(n)$. To match up the positions according to lemma 4-1, it needs to calculate the distances 4 times and run the logical operations 2 times if there are four positions in each combined sub-group, whose time complexity $O(f_4)$ is $O(f_4) = O(n)$.

The collision avoidance strategy is executed sequentially, so that the time complexity of the collision avoidance strategy is $O(f) = O(n \log n)$ based on the proof above.

According to the collision avoidance strategy, each AUV not only identifies its own destination, but also the expected positions relationship among all the AUVs. The trajectory of each AUV is a straight line connecting the initial and final positions in the pyramid frame, which can be noted as (η_i, d_i). η_i stands for the orientation from the initial position to the final position, which can be adjusted by a compass; d_i is the distance between the initial and final positions, which can be estimated by IMU (Inertial Measurement Unit). Considering that the orientations of AUVs are estimated in common frame, the trajectories are transformed into common frame according to equation (4-8). Finally, the pyramid pattern is achieved by the local position-based control method.

4.3 Simulation of the formation building

Simulations are done in the 3D simulated deep sea environment established in Blender (see Fig. 3-7). Considering that the working range of cameras in the deep sea usually is $d \le 5m$, and the horizon view field is $\gamma \le 60^\circ$, the expected relative distance and angle between parents and children in the pyramid pattern are set to: $d_s = 3.5m$, $\varphi_s = 30^\circ$ respectively. Assuming that there are no impacts from wave and tide, the water current flows are assumed with constant speed, which can be compensated completely by adding a speed offset. In the simulations, the visual sensing is assumed to be similar to the real experiments, so that we do not need to add extra noises to the images; to simulate the movement errors, the random noises within $\pm 10\%$ of expected (d_s, φ_s) are added into the trajectories (η_i, d_i) of each CISCREA.

We recall below the steps of pyramid building based on the local position-based control method

described in section 4.2:

(1) searching surroundings to get neighbors' information: each CISCREA-*i* rotates around itself to search and detect neighbors by using its camera, then estimates their IDs, relative distances d_{ij} and relative angles φ_{ij}. Combining with the global orientation θ_{ij} of CISCREA-*i* at this moment estimated by its compass, the position of each neighbor CISCCREA-*j* in the local frame $o_{li}x_{li}y_{li}$ can be calculated by equation (3-23).

(2) Sharing the neighbors information and building the common frame: each CISCREA can get the neighboring information of all the CISCREAs by communication. To make sure the neighboring information correct, the neighbors with huge relative errors are checked out and deleted. Then the common frame $o_cx_cy_c$ is derived.

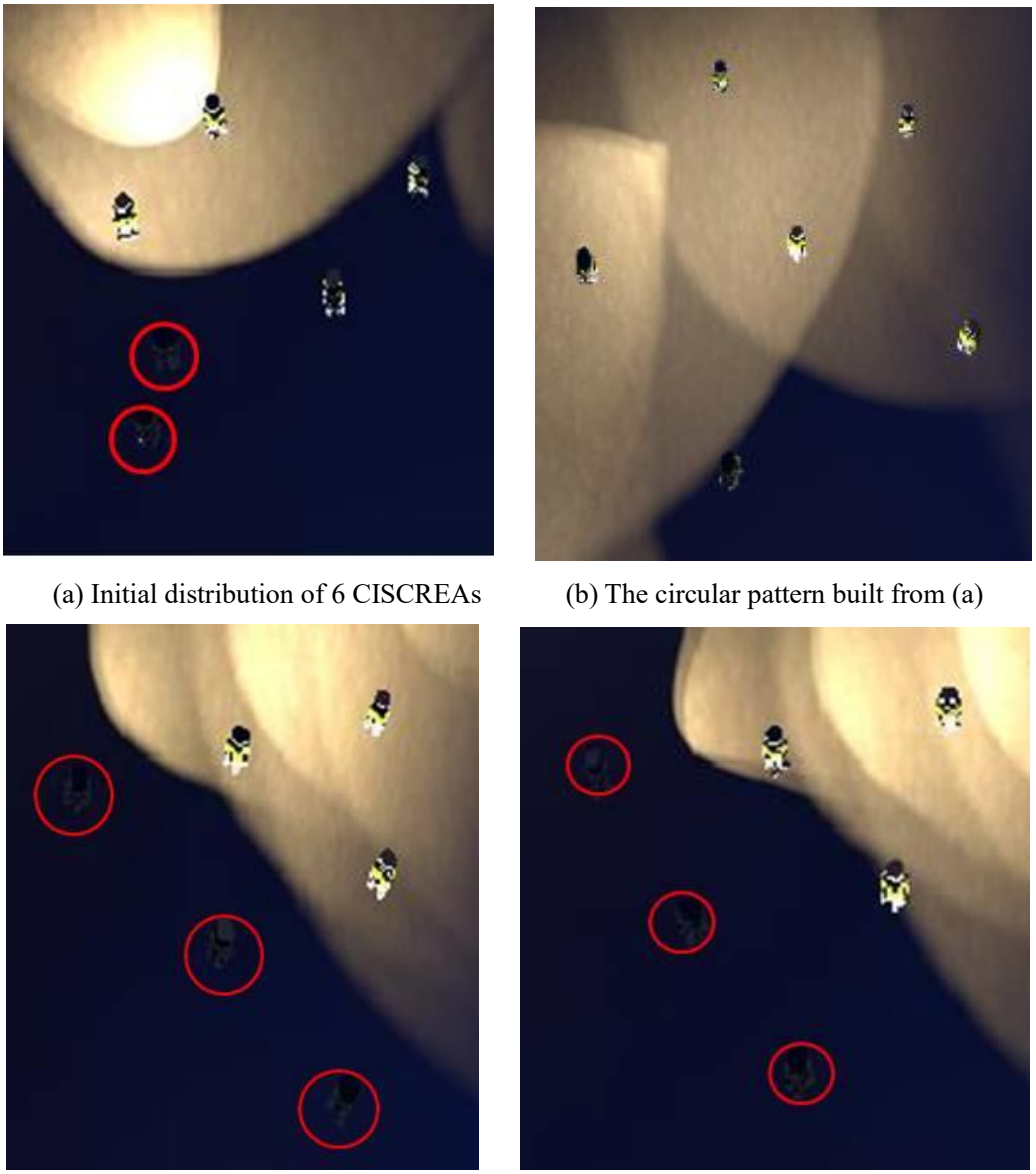
(3) Determining the distribution of pyramid pattern: the pyramid distribution is identified and optimized by collision avoidance strategy. The principal axis and the leader's position of an pyramid pattern are determined by one of two ways (CP or PCA). Subsequently, the pyramid distribution can be obtained according to the definition of pyramid pattern (see section 2.2) and the identification of pyramid distribution (see section 4.2.1).

(4) Planning the straight non-intercrossing trajectories: based on the collision avoidance strategy, the positions in both initial and pyramid distributions are separated to constitute the combined sub-groups. Then the straight non-intercrossing trajectories of AUVs are planned in each combined sub-group according to the lemma 4-1.

Then the pyramid is built successfully, when each CISCREA reaches its destination along its trajectory.

4.3.1 Pyramid building in case of explicit IDs

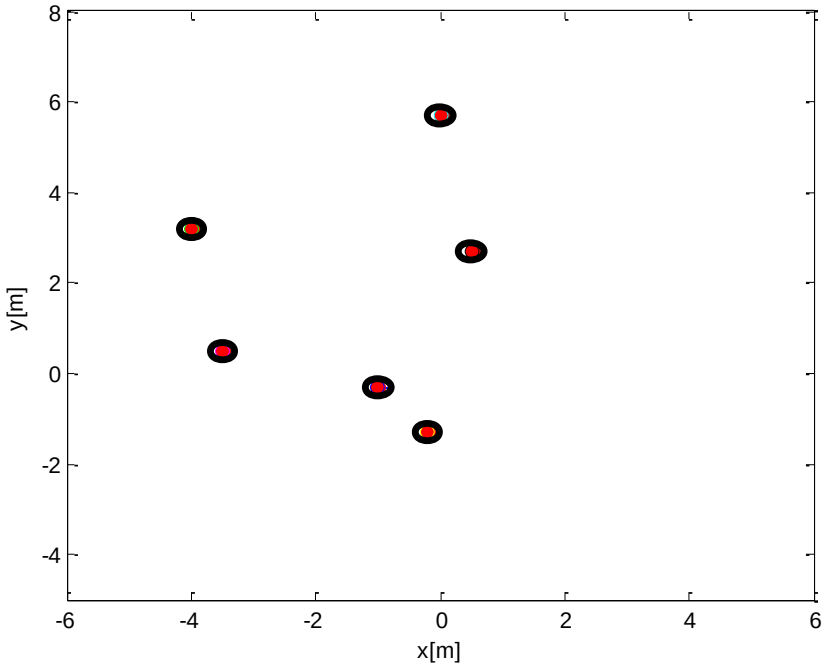
Fig 4-7 and Fig. 4-8 show respectively the built pyramid patterns and corresponding trajectories of CISCREAs from the same initial distribution with two ways (CP and PCA) to get the principal axis of pyramid. In Fig. 4-7(c) and 4-7(d), we remark the CISCREAs located at the last layer who are hardly to be seen. The main reason is that the simulation environment is similar to the real deep sea environment, which lacks of natural light, and the artificial light installed on each CISCREA can only illuminate the CISCREAs in front of them, so that the CISCREAs at the last layer cannot be exposed to the light.



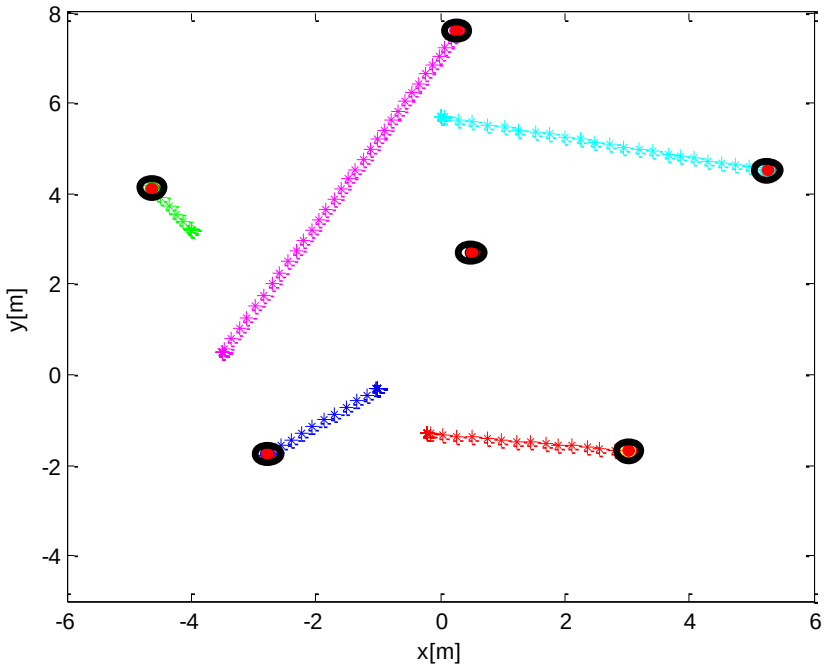
(a) Initial distribution of 6 CISCREAs (b) The circular pattern built from (a)

(c) Pyramid pattern built from (b) (d) Pyramid pattern built directly from (a) with the PCA method

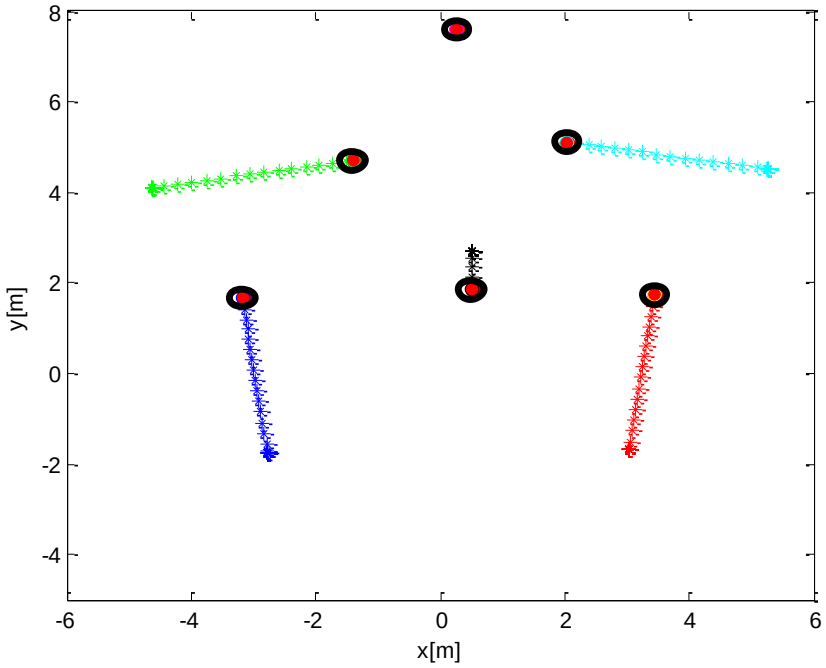
Fig. 4-7 Examples of built pyramid patterns by 6 CISCREAs with explicit IDs based on local position-based control method developed in section 4.2



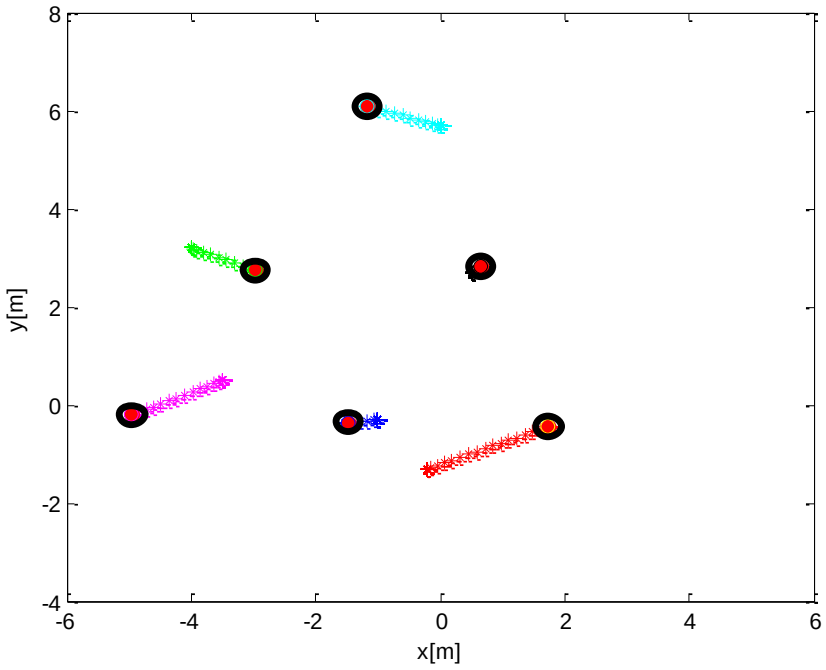
(a)



(b)



(c)



(d)

Fig. 4-8 The trajectories of CISCREAs to build the pyramid patterns with two ways (CP and PCA) respectively to compute their principal axes from the same initial distribution corresponding to Fig. 4-7. (a) the initial distributions of CISCREAs; (b),(c) the trajectories of pyramid building with CP method: (b) the built middle circle pattern, (c) the built pyramid pattern; (d) the trajectories of pyramid building with PCA method.

To evaluate the errors of the built pyramids, we calculated 4 kinds of errors: relative position errors x_{re}, y_{re} along x axis and y axis of the pyramid frame, relative distance d_{re} and angle errors φ_{re}. These relative errors are computed by comparing the measured values d_a, φ_a, (x_a, y_a) with the expected values d_s, φ_s, (x_s, y_s) of relative distance, relative angle and relative positions, shown in equation (4-10).

$$\begin{cases} x_s = d_s \sin\varphi_s \\ y_s = d_s \cos\varphi_s \\ x_a = d_a \sin\varphi_a \\ y_a = d_a \cos\varphi_a \\ x_{re} = x_a - x_s \\ y_{re} = y_a - y_s \\ d_{re} = d_a - d_s \\ \varphi_{re} = \varphi_a - \varphi_s \end{cases} \quad (4\text{-}10)$$

Then we define two radios $\frac{d_{re}}{d_s}$ and $\frac{\varphi_{re}}{\varphi_s}$ to judge whether the pyramid is built successfully or not at the convergent state. Considering that the AUVs will continue to coordinate their relative positions when moving forward to run tasks, the allowed relative error range can be set large for saving the convergent time. As a result, we evaluate the pyramid is achieved when $\frac{d_{re}}{d_s}$ and $\frac{\varphi_{re}}{\varphi_s}$ are both within $\pm 40\%$.

To test the stability of formation building with the presented local position-based control method, we repeat the formation building 100 times from different initial distributions with 6 CISCREAs. The results show that each CISCREA can converge to its destination and the pyramids can be built 100%, though the relative errors are large sometimes.

The relative distance error ratios $\frac{d_{re}}{d_s}$ and relative angle error ratios $\frac{\varphi_{re}}{\varphi_s}$ at convergent states are given in Fig. 4-9 (for the CP approach) and Fig. 4-10 (for the PCA approach). Besides, the fluctuations of relative positions errors (x_{re}, y_{re}) during the process of pyramid building based on CP and PCA are shown in Fig. 4-11 and Fig. 4-12. In Fig. 4-11 and Fig. 4-12, the fluctuations reflect the different steps of formation building. The method based on CP includes 3 steps: the first horizon stage stands for the searching of neighbors, during this period, robots keep their coordinates invariant (period 1 of Fig. 4-11(a) and (b)) and only turn around themselves; the second fluctuation to another horizon stage represents the step of building a circle pattern (period 2 of Fig. 4-11(a) and (b)); and the third decreasing to convergence stage corresponds to the step of pyramid building (period 3 of Fig. 4-11(a) and (b)). The PCA approach includes 2 steps without the second stage of

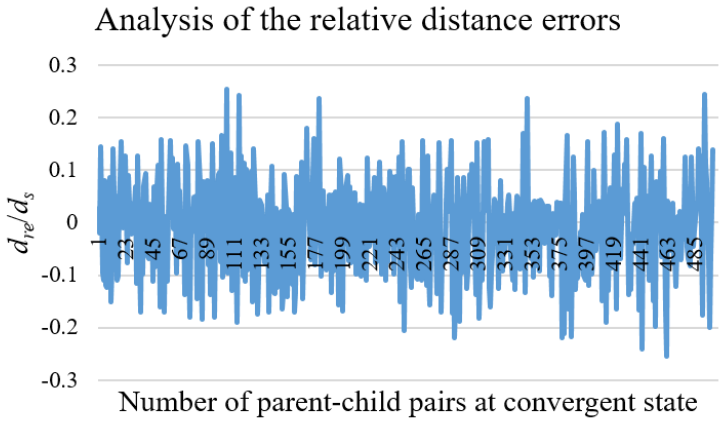
building a circle pattern. Besides, with the increasing of the number of AUVs in the formation, we notice that the relative errors become slightly larger which is probably due to the error accumulation when establishing common frame, shown in table 4-1 and table 4-2.

Table 4-1 Examples of relative errors when building pyramid patterns with 5, 6 and 7 AUVs 100 times respectively based on CP method.

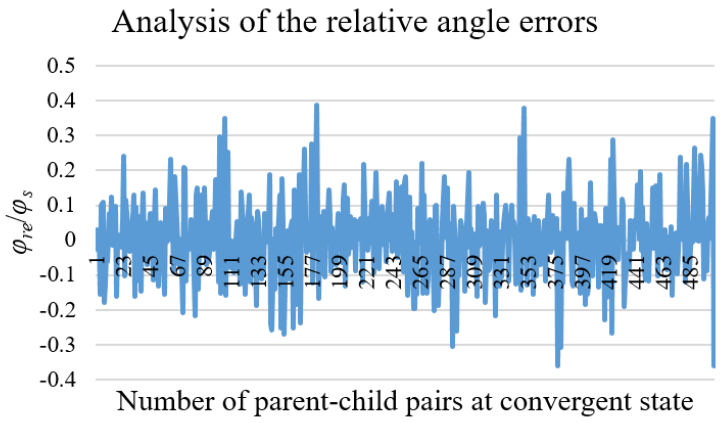
	d_{re}/d_s				φ_{re}/φ_s			
	aver	*stdeva*	*max*	*min*	*aver*	*stdeva*	*max*	*min*
5 robots	0.008	0.0554	0.2458	-0.2193	0.007	0.1044	0.3506	-0.3541
6 robots	-0.0063	0.0959	0.2541	-0.2549	0.0058	0.1097	0.3864	-0.3589
7 robots	0.0067	0.1143	0.3082	-0.3107	-0.013	0.09	0.41088	-0.392

Table 4-2 Examples of relative errors when building the pyramid patterns with 5, 6 and 7 AUVs respectively based on PCA method.

	d_{re}/d_s				φ_{re}/φ_s			
	aver	*stdeva*	*max*	*min*	*aver*	*stdeva*	*max*	*min*
5 robots	0.0089	0.0775	0.2188	-0.2619	0.0185	0.1118	0.3541	-0.403
6 robots	0.0005	0.1036	0.2478	-0.2455	-0.0337	0.1303	0.3673	-0.4222
7 robots	0.002	0.1258	0.3336	-0.327	0.026363	0.1653	0.4054	-0.4127

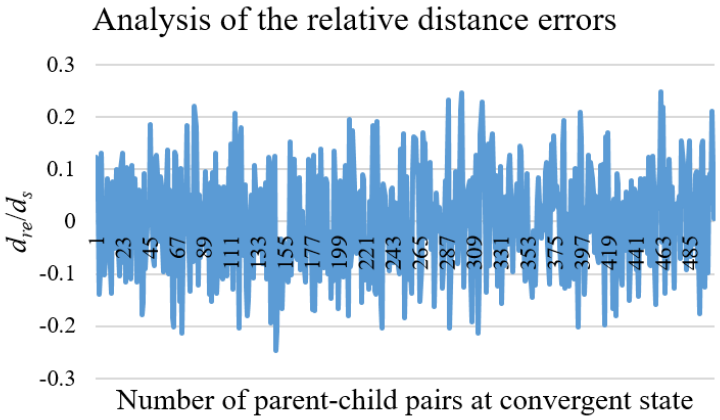


(a)

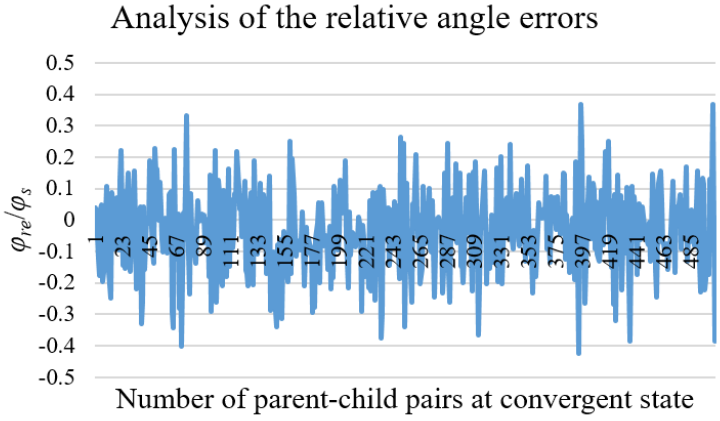


(b)

Fig. 4-9 Relative distance error ratios $\frac{d_{re}}{d_s}$ and relative angle error ratios $\frac{\varphi_{re}}{\varphi_s}$ at convergent state of pyramid building based on the CP approach by 6 AUVs (5 parent-child pairs) with explicit IDs

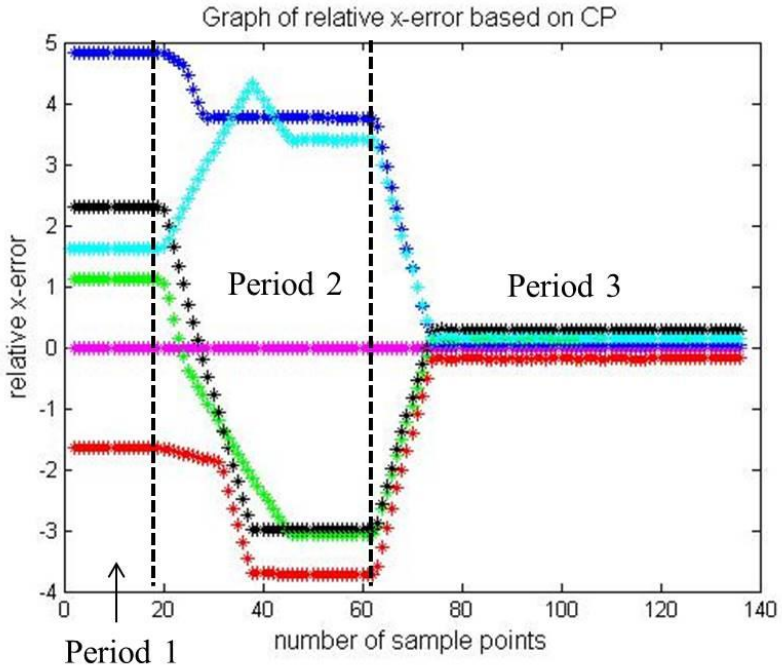


(a)

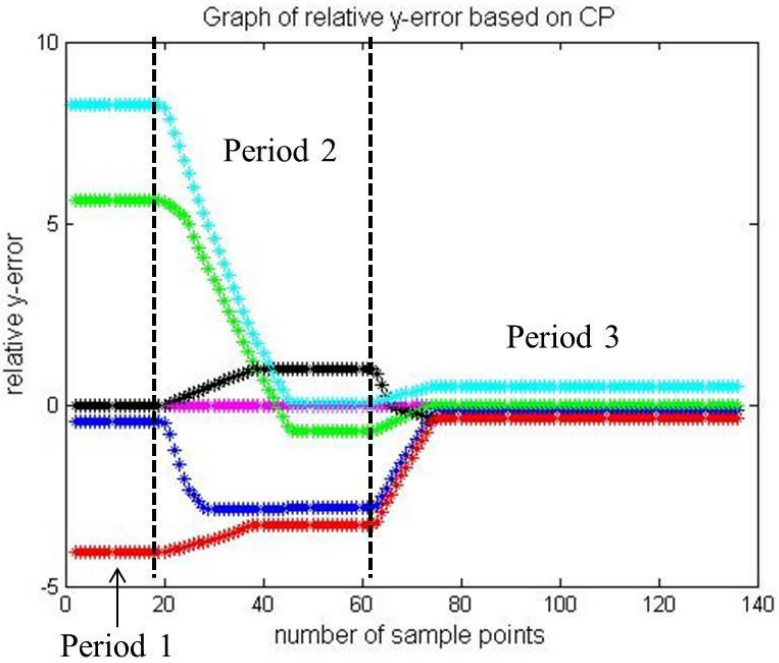


(b)

Fig. 4-10 Relative distance error ratios $\frac{d_{re}}{d_s}$ and relative angle error ratios $\frac{\varphi_{re}}{\varphi_s}$ at convergent state of pyramid building based on the PCA approach by 6 AUVs (5 parent-child pairs) with explicit IDs



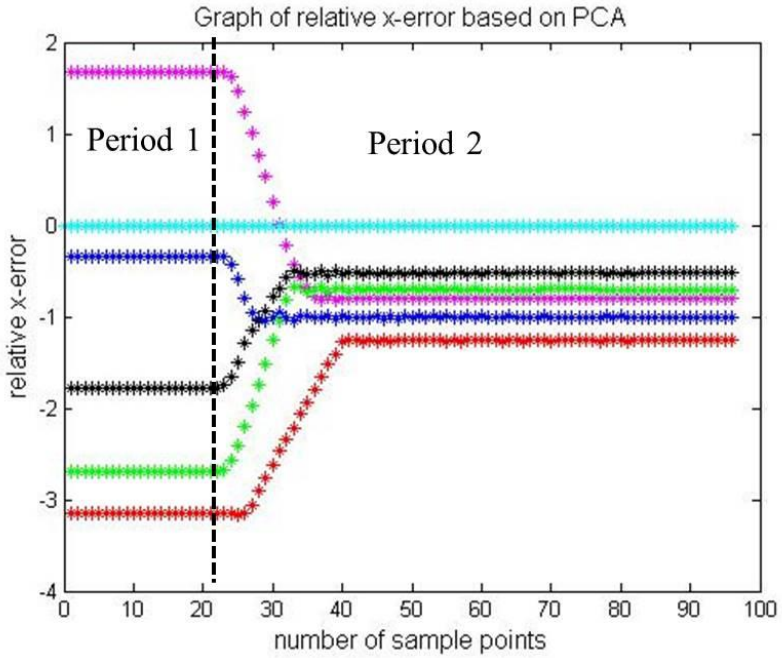
(a) The converging process of x_{re}



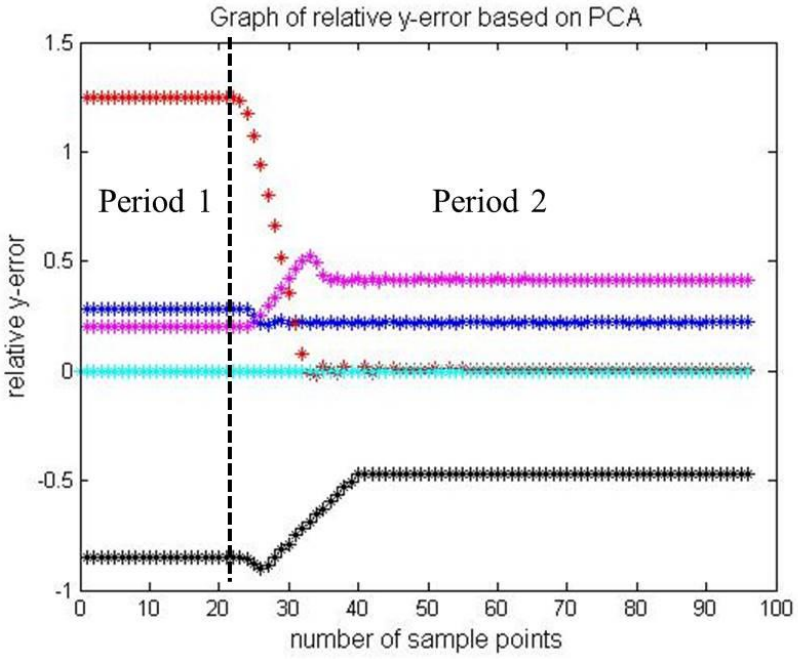
(b) The converging process of y_{re}

Fig. 4-11 Relative position errors (x_{re}, y_{re}) between each CISCREA and its parent based on the CP approach by 6 CISCREAs. For visual convenience, the points are sampled by taking one point

every 100 points from original data.



(a) The converging process of x_{re}



(b) The converging process of y_{re}

Fig. 4-12 Relative position errors (x_{re}, y_{re}) between each CISCREA and its parent based on PCA approach by 6 CISCREAs. For visual convenience, the points are sampled by taking one point every 100 points from original data.

From Fig. 4-9 to Fig. 4-12, we find that the local position-based control method is valid to build the pyramid pattern in the obstacle-free environment. And based on both ways (CP and PCA), the pyramid distribution can be identified. Moreover, the pyramid building can be achieved 100%

with the allowed error range $\frac{d_{re}}{d_s}$ and $\frac{\varphi_{re}}{\varphi_s}$ within $\pm 40\%$.

To evaluate the contribution of the image processing step to the relative errors, we run 100 times another simulation with 6 robots. In this simulation, the positions of the CISCREAs detected by cameras are given directly by Blender, not calculated in the image processing step (see section 3.2). All the relative errors are presented in table 4-3 and 4-4, where *_*img* stands for errors with the positions obtained by image processing, while *_*blender* represents errors with the positions given in Blender.

Compared the data in table 4-3 and 4-4, the average errors are small, the position errors and distance errors are within 25%, but the biggest angle error is about 40%. The big angle error is mainly caused by the relative positions calculated depending on image processing (like AUV overlapping, calibration etc.). Besides, because the expected relative angle φ_s is not big (30 degree), a small relative position error (x value or y value) may lead to big angle error.

And the accuracy obtained with the CP approach is slightly higher than that obtained by the PCA approach, the main reason is that when building the intermediate circle pattern, the positions of CISCREAs around the circle are finely tuned according to the CISCREA in the center. But for precisely this reason, the time consumption of the CP method is much more than that of the PCA method. The time spent for detecting the neighbors' positions is the same, about 75 seconds. The time consumptions to build pyramid patterns are shown in table 4-5. Repeated simulations show the consuming time fluctuates from 10% to 15%, depending on the initial distribution.

Table 4-3 Performance of the CP approach with 6 CISCREAs. For each error-criterion, we give the minimum, maximum, average and standard deviation in two cases (with or without image processing step)

	$\lvert x_{re}/x_s \rvert$	$\lvert y_{re}/y_s \rvert$	$\lvert d_{re}/d_s \rvert$	$\lvert \varphi_{re}/\varphi_s \rvert$
aver_img	0.0858	0.0695	0.08	0.0826
stdev_img	0.0757	0.0585	0.0553	0.0793
max_img	0.1507	0.1922	0.2541	0.3864
min_img	7.6298E-05	4.8761E-05	0.0008	0.0002
aver_blender	0.0168	0.0128	0.01530	0.0138
stdev_blender	0.0137	0.0059	0.0064	0.0089
max_blender	0.0713	0.0335	0.0451	0.0428
min_blender	4.4626E-05	0.0003	0.001	8.4327E-05

Table 4-4 Performance of the PCA approach with 6 CISCREAs. For each error-criterion, we give the minimum, maximum, average and standard deviation in two cases (with or without image processing step)

	$\lvert x_{re}/x_s \rvert$	$\lvert y_{re}/y_s \rvert$	$\lvert d_{re}/d_s \rvert$	$\lvert \varphi_{re}/\varphi_s \rvert$
aver_img	0.1101	0.0705	0.09154	0.1054
stdev_img	0.0956	0.0584	0.0567	0.0891

max_img	0.2078	0.2187	0.2478	0.4222
min_img	0.0002	0.0002	0.0047	9.5209E-05
aver_blender	0.03	0.0078	0.0177	0.0247
stdev_blender	0.0185	0.0068	0.0089	0.0164
max_blender	0.0775	0.03676	0.04469	0.0851
min_blender	8.62E-05	1.202 E-05	0.0004	1.5695 E-05

Table 4-5 Time consumption to build a pyramid pattern with CP and PCA

	Based on CP/s	Based on PCA/s
Time consumption	300 ± 30	100 ± 15

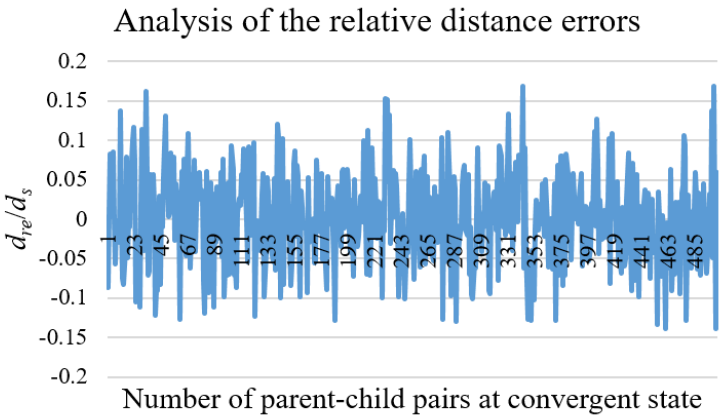
4.3.2 The pyramid building in case of implicit IDs

Results obtained in this case are similar to those in section 4.3.1 (see Fig. 4-13). With the same way, to verify the reliability of building a pyramid pattern, 6 CISCREAs are used to build the pyramid patterns 100 times by CP and PCA with different initial distributions in a bounded visual region. The results imply that the pyramid building can be achieved in all cases. And the time consumptions are 370 ± 15 s and 165 ± 15 s respectively, which is a little higher than those in section 4.3.1. The relative distance errors $\frac{d_{re}}{d_s}$ and angle errors $\frac{\varphi_{re}}{\varphi_s}$ at convergent state of these 100 simulations are shown in Fig. 4-14 and Fig. 4-15.

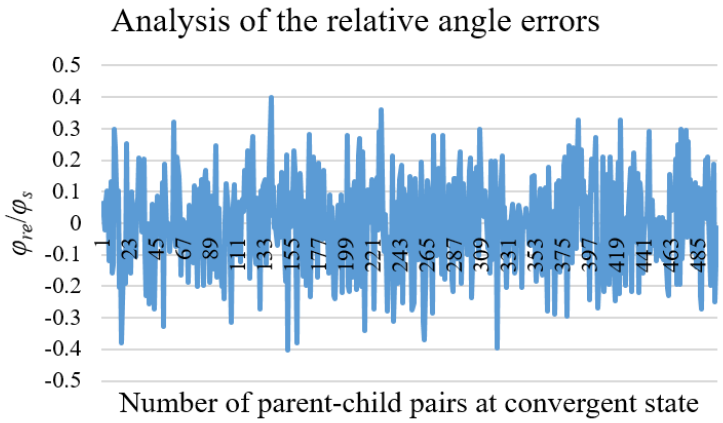
(a) An example of the pyramid built based on CP

(b) An example of the pyramid built based on PCA

Fig. 4-13 The pyramid patterns built by 6 CISCREAs with implicit IDs based on local position-based control method developed in section 4.2

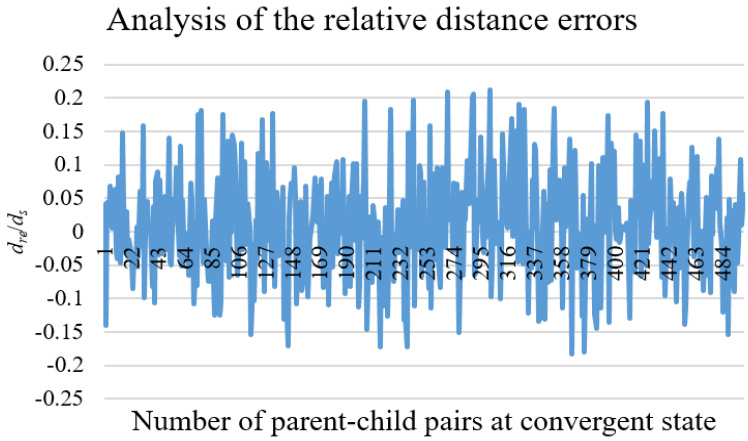


(a)

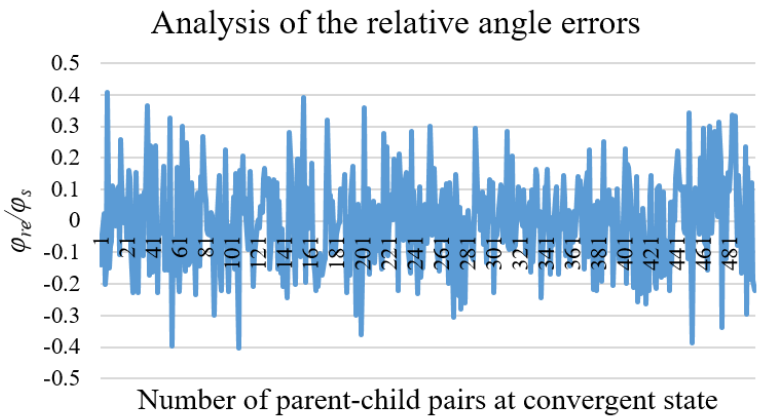


(b)

Fig. 4-14 Relative distance error ratios $\frac{d_{re}}{d_s}$ and relative angle error ratios $\frac{\varphi_{re}}{\varphi_s}$ at convergent state of pyramid building based on the CP approach by 6 AUVs (5 parent-child pairs) with implicit IDs



(a)



(b)

Fig. 4-15 Relative distance error ratios $\frac{d_{re}}{d_s}$ and relative angle error ratios $\frac{\varphi_{re}}{\varphi_s}$ at convergent state of pyramid building based on the PCA approach by 6 AUVs (5 parent-child pairs) with implicit IDs

Comparing with the results in section 4.3.1, we find that the relative distance error ratios $\frac{d_{re}}{d_s}$ and relative angle error ratios $\frac{\varphi_{re}}{\varphi_s}$ given in Fig. 4-14 and Fig. 4-15 are similar to the results shown in Fig. 4-9 and Fig. 4-10. Though it needs to spend more time to distinguish the implicit IDs of CISCREAs, the implicit IDs will be more convenient when there are many AUVs in the system. For example, if we use explicit IDs, when we add new AUVs to the system, more complex explicit ID should be designed, and the explicit ID may lead to the modification of the hydrodynamic models of AUVs. However, using implicit IDs has no this problem.

4.4 Chapter summary

For the obstacle-free unknown environment, this chapter describes the local position-based control method proposed to be used to build the pyramid patterns based on the visual system to sense the surroundings. Inspired from the global position-based control method, the initial distribution of AUVs is got in the constructed common frame by sharing the neighbors' information.

Subsequently, considering that the multi-AUV system lacks of the assistance of global sensors, it is very difficult to update the positions in the common frame on-line. To solve this problem, a collision avoidance strategy is proposed to optimize the pyramid pattern to make it tend to the initial distribution and to optimize the separations of sub-groups to make them be as disjoint as possible. In which, two ways (CP and PCA) are introduced to identify the principal axis of the pyramid pattern.

At last, in each combined sub-group, the non-intercrossing straight trajectories are planned based on lemma 4-1 to achieve the pyramid building without trajectory update in the limited communication condition.

The simulation results have verified the performance of the local position-based control method for building pyramid patterns, though the relative distance and angle errors between parent and child sometimes are large. And the CP and PCA methods both are valid to identify the distribution of pyramid. The accuracy of the built pyramid based on CP is slightly higher than based on PCA, but the PCA method saves a lot of time compared with the CP method. Generally, the PCA method is more practical.

5 Asynchronous discrete consensus-based formation control with local information

In the process of pyramid pattern building, the local position-based control method will fail, when one or more AUVs are stopped by the unexpected static or dynamic obstacles. In these cases, the distributed methods based on neighbors' information should be considered. Considering the communication difficulty in the water and the independent sensing property of each on-board video camera, each AUV can update its trajectory independently according to the neighbors' information got by image processing and the expected position relationship got in section 4. Then asynchronous discrete consensus-based control methods can be considered to accomplish the formation building. Since formation building can be seen as a "rendez-vous" problem, which is a consensus problem [85], it can be achieved by the consensus algorithm to coordinate the robots to get an agreement depending on their neighbors' states. When the AUVs reach the agreement, the formation is built. In this thesis, the neighbors' states are composed of local information: the relative positions in global frame got by on-board video camera and compass, or the relative distance vectors in local frames got by on-board video camera. As a result, the trajectories are updated by two ways: the consensus algorithm associated with displacement information and the consensus algorithm associated with the relative distance information.

The trajectory update based on the consensus algorithm is a discrete process only depending on the current neighbors' local information with no relations to the past, which results in that the convergent process can be seen as a Markov chain. Thus, Markov chain and transition matrix theories are described first.

5.1 Markov chain and transition matrix

Definition 5-1 A stochastic process $\{X_n, n \in T\}$, if for any integer $n \in T$ and arbitrary $i_0, i_1, ...i_{n+1} \in \mathbf{I}$, the conditional probability meets [125,126]:

$$P(X_{n+1} = i_{n+1} \mid X_n = i_n, X_{n-1} = i_{n-1}, ..., X_0 = i_0) = P(X_{n+1} = i_{n+1} \mid X_n = i_n),$$

then $\{X_n, n \in T\}$ is called as a Markov chain, and it is a first-order Markov chain

In other words, the state of next step is independent of the past and only relies upon the current state.

Definition 5-2 A conditional probability $P_{ij}(n) = P(X_{n+1} = j \mid X_n = i)$ defines the one-step transition matrix of Markov chain $\{X_n, n \in T\}$, where $i, j \in \mathbf{I}$.

Generally, the transition matrix is not only relevant to i, j, but also relevant to n. When $P_{ij}(n)$ is independent on n, the associated Markov chain has a stationary transition probability.

Definition 5-3 If $\forall i, j \in \mathbf{I}$, the transition matrix $P_{ij}(n)$ is not related to n, then the associated Markov chain is called as homogeneous Markov chain ($P_{ij}(n)$ noted as P_{ij}), otherwise, it is called as non-homogeneous Markov chain, noted as $P_{ij}(n)$.

First-order Markov chain is associated with the transition matrix $\mathbf{P}$. The element located at $(i, j)^{th}$ of $\mathbf{P}$ is P_{ij}. The elements of $\mathbf{P}$ meet two important properties:

(1) $\mathbf{P}$ is a non-negative matrix, noted as $\forall i, j \in \mathbf{I}, P_{ij} \geq 0$;

(2) $\forall i \in \mathbf{I}, \sum_{j \in \mathbf{I}} P_{ij} = 1$.

Theorem 5-1[125] If a Markov chain is irreducible and aperiodic, then it has a unique stationary distribution π. There is one unique left eigenvector (normalized such that the entries sum to 1) of its transition matrix $\mathbf{P}$ with eigenvalue 1. Moreover, $P_{ij}^n \to \pi(j)$ as $n \to \infty$ for all $i, j \in \mathbf{I}$.

This means that the transition matrix $\mathbf{P}$ of an irreducible and aperiodic Markov chain must converge to a stationary distribution π.

As a Markov chain can be expressed by a transition matrix $\mathbf{P}$, and one pattern can be seen as a graph *G*, there is a relationship between $\mathbf{P}$ and the adjacent matrix $\mathbf{A}$ of the graph *G* depending on lemma 2-1. To get $\mathbf{P}$ from the adjacent matrix $\mathbf{A}$ of the graph *G*, and to analyze the convergence of a Markov chain based on $\mathbf{P}$, some matrix theories are presented below.

According to the definition 2-8 (see section 2.1), the sum of each row of Laplacian matrix $\mathbf{L}$ is 0, then 0 is one eigenvalue of $\mathbf{L}$, the corresponding eigenvector is $\mathbf{1} = [1,1,...1]^T$. Then $\forall t > 0$, the sum of each row of matrix $e^{-\mathbf{L}t}$ is equal to 1, and 1 is one eigenvalue of $e^{-\mathbf{L}t}$, the corresponding eigenvector is also $\mathbf{1} = [1,1,...1]^T$. Besides, $e^{-\mathbf{L}t}$ is one non-negative matrix, $e^{-\mathbf{L}t}$ can be seen as a transition matrix according to the two properties in section 5.1.

Lemma 5-1 If there is at least one spanning tree included in the topology corresponding to the Laplacian matrix $\mathbf{L}$, then $e^{-\mathbf{L}t}$ has and only has one eigenvalue equal to 1, and the other eigenvalues are positive and their modulus are smaller than 1[85][126].

Lemma 5-2 An irreducible transition matrix $\mathbf{P}$ is primitive, if it has and only has one eigenvalue with maximum modulus, and its left and right eigenvectors $\mathbf{W}$ and $\mathbf{V}$ satisfy $\mathbf{w}^T\mathbf{P} = \mathbf{w}^T$, $\mathbf{Pv} = \mathbf{v}$ and $\mathbf{v}^T\mathbf{w} = 1$, then $\lim_{k \to \infty} \mathbf{P}^k = \mathbf{v}\mathbf{w}^T$, $\mathbf{v} = [1,1,...,1]_{N \times 1}^T$, $\mathbf{w}^T = [w_1, w_2, ...w_N]_{1 \times N}$, and

$\sum_{i=1}^{N} w_i = 1$ [85,86,90,126].

According to lemma 5-1, the transition matrix $\mathbf{P}$ can be obtained, and lemma 5-2 will be used in section 5.3 and 5.4 to compute the weights of each neighbors in the consensus algorithm.

5.2 A brief description of consensus algorithms

To accomplish the tasks, the cooperation among robots is necessary for the multi-robot systems. Then it needs to solve two key problems for the success of coordination strategy: the definition and management of shared information among robots and how to complete coordination control by using these information.

To share information usually means that one information (such as the relative positions among robots) of a fleet of robots needs to reach the agreement by continuous coordination, such problems are the consensus problems or agreement problems. The basic idea of consensus algorithm is to let all the robots update their information states, and let the states reach the agreement according to neighbors' information [85].

Let assume that $\mathbf{x}_i$ is the information state of robot-*i* to be coordinated. The information states can be positions, speeds, amplitudes, and decisive variables etc. According to the description in literature [85,86,89], one model of linear continuous consensus algorithm is:

$$\dot{\mathbf{x}}_i(t) = -\sum_{j \in J_i(t)} \alpha_{ij}(t)(\mathbf{x}_i(t) - \mathbf{x}_j(t)) \qquad (5\text{-}1)$$

Where, $J_i(t)$ is the set including all the neighbors robot-*j* of robot-*i* at moment *t*, $\alpha_{ij}(t)$ expresses the time-variant weights. In other words, the information state of each robot at time *t* is decided by its neighbors' information states at this moment.

Remark: The robots are allowed to not find any neighbors and/or fail to exchange information sometimes, in these cases, the robots keep their states and continue to re-detect their neighbors. But most of the time, robots can update their states according to their neighbors, otherwise, the agreement among all the robots cannot be reached, then the coordination of the multi-robot system will fail.

Then equation (5-1) can be expressed by matrix form: $\dot{\mathbf{x}}_i(t) = -\mathbf{L}\mathbf{x}(t)$, in which, $\mathbf{L}$ is the Laplacian matrix, $\mathbf{x}(t) = [\dot{x}_1(t), ..., \dot{x}_n(t)]^T$ [130].

Correspondingly, the discrete consensus algorithm is shown in equation (5-2) [87,130,131].

$$\dot{\mathbf{x}}_i[k+1] = -\sum_{j \in J_i[k] \cup \{i\}} \beta_{ij}[k]\mathbf{x}_j[k] \qquad (5\text{-}2)$$

Where, $\sum_{j \in J_i[k] \cup \{i\}} \beta_{ij}[k] = 1$, and $\beta_{ij}[k] > 0, \forall j \in J_i[k] \cup \{i\}$.

Equation (5-2) means that the next state of each robot is the weighted average of its neighbors'

states at current moment. In some point, one robot cannot find any neighbors to get the state information, then the state of this robot will be kept unchanged.

When $t \to \infty$, $\left\| \mathbf{x}_i(t) - \mathbf{x}_j(t) \right\| \to 0$ ($\forall i \neq j$) (continuous consensus algorithm), or when $k \to \infty$, $\mathbf{x}_j[k] \to 0$ ($j \in J_i[k] \cup \{i\}$) (discrete consensus algorithm), the consensus algorithm is convergent.

5.3 The trajectory update based on the consensus algorithm with the displacement-based method

In a consensus algorithm, as described in section 5.2, the state update of each robot depends on its neighbors' states. As searching neighboring AUVs based on the on-board video camera can be viewed as a discrete process, we study discrete consensus algorithm in this thesis. A common discrete consensus algorithm with the time-invariant undirected topology can be expressed below:

$$\mathbf{x}_i[k+1] = \sum_{j=1}^{N_i} P_{ij}[k]\mathbf{x}_j[k] \qquad (5\text{-}3)$$

Where, $\mathbf{x}_i[k+1]$ is the state of AUV-*i* at the $(k+1)_{th}$ sample loop, $P_{ij}[k]$ is the $(i,j)^{th}$ element of transition matrix $\mathbf{P}^k$ at the k_{th} sample loop, N_i is the neighbors' number of AUV-*i* and invariant, since this is a time-invariant undirected topology.

After accomplishing the match-up between the positions in both initial and pyramid distributions, the expected position relationship among AUVs can be got (see section 4.2). Then each AUV can compute the expected relative positions of its neighbors ((x_s, y_s)) as reference states. The trajectory update of each AUV is executed by continuously running the loop "search, compute, move": (1) searching neighbors as described in section 3.2.1, (2) computing the current relative positions ((x_a, y_a)) with the relative distances and angles got from images and the global orientations estimated by compass, and to update its trajectory by comparing with (x_s, y_s), (3) moving to the new state along the updated trajectory. Since the neighbors' information is extracted by each AUV's on-board camera and compass, its trajectory can be updated independently without any communication between AUVs. This implies that the formation building by using the discrete consensus algorithm can be considered as an asynchronous process.

Further, during the process of the pyramid building, each AUV is always in a dynamic state of searching for neighbors or moving along an updated trajectory, which makes each neighbor dynamic. Then, the sub-digraph $G_i[k_i]$ composed by AUV-*i* and its neighbors found at k_i loop is a time-variant digraph, and the whole digraph G composed by all AUVs is time-variant. Consequently, we introduce especially the asynchronous consensus algorithm with the time-variant digraph from

literature [127,128]. However, we cannot directly apply the methods in [127,128] to solve our problem. Both of them get the local information by means of communication on-line in continuous time.

In addition, the confidence of each edge is variable, as $G_i[k_i]$ is a time-variant digraph. Then we introduce weights to express the changeable confidences [130]. Since the discrete consensus algorithm can be regarded as a first-order Markov chain [85], we propose a new way based on on-step transition matrix to calculate these weights. The weights of all the neighbors are the limited values of the transition matrix $P_i[k_i]$ associated with the sub-digraph $G_i[k_i]$ constituted by AUV-*i* and its neighbors. When the sensors contain camera and compass, the attainable neighbors' information is the global relative positions, then an asynchronous discrete consensus algorithm associated with the displacement-based control is proposed to update trajectories and expressed in equation (5-4).

$$\mathbf{e}_i[(k+1)_i] = \sum_{j=1}^{N_i[k_i]} w_{ij}[k_i]\mathbf{e}_j[k_i] \qquad (5\text{-}4)$$

Where, k_i and $(k+1)_i$ are the k_{th} and $(k+1)_{th}$ updating loop of AUV-*i*. $N_i[k_i]$ is the number of AUVs in the sub-digraph $G_i[k_i]$, which includes the neighbors found in the k_i loop and AUV-*i* itself. $\mathbf{e}_i[(k+1)_i]$ stands for the updated trajectory, expressed by a relative position error. $\mathbf{e}_j[k_i] = \mathbf{p}_{ij}[k_i] - \mathbf{p}_{ij}^*$ is the relative error between the measured and expected relative positions of AUV-*i* and its neighbors AUV-*j*. $w_{ij}[k_i]$ represents the confidence of AUV-*j* to AUV-*i*. To compute $\mathbf{e}_i[(k+1)_i]$, AUV-*i* should know the $N_i[k_i]$, $\mathbf{e}_j[k_i]$ and $w_{ij}[k_i]$ at k_{th} loop, which is detailed below.

5.3.1 Calculation of the errors of relative positions $\mathbf{e}_j[k_i]$

To compute $\mathbf{e}_j[k_i]$, $\mathbf{p}_{ij}^*$ and $\mathbf{p}_{ij}[k_i]$ should be determined. Both of them are calculated in each AUV local frame given in section 3, shown in Fig. 3-10. $\mathbf{p}_{ij}^*$ is computed according to the definition of pyramid pattern, the expected distance d_s and angle φ_s, and the expected position relationship (see section 2.2).

To get $\mathbf{p}_{ij}[k_i](x_{ij}[k_i], y_{ij}[k_i])$, AUV-*i* should execute its k_i loop, where AUV-*i* searches its neighbors step by step. In each step, AUV-*i* captures one image (see one example in Fig 3-11(a)). And the step length should not be too large for making sure to get all the information in each AUV's surroundings. Then all the neighbors are recognized depending on their yellow parts, and the positions in images $(u_{ij}[k_i], v_{ij}[k_i])$, sizes $h_{ij}[k_i]$ and explicit IDs (in this chapter, we only use

the AUVs with explicit IDs to verify the performance of the proposed control method) are extracted, as explained in section 3.2. Subsequently, the relative distances $d_{ij}[k_i]$ ($d_{ij}[k_i]=\left\|\mathbf{p}_{ij}\right\|$) and relative angles $\varphi_{ij}[k_i]$ can be estimated from a single image according to the distance model (equation (3-21)) and angle model (equation (3-22)). Combined with the global orientation $\theta_{ij}[k_i]$ estimated by the compass of AUV-*i* at the moment of capturing this image, the relative position $\mathbf{p}_{ij}[k_i](x_{ij}[k_i], y_{ij}[k_i])$ of AUV-*j* in local frame $o_{lj}x_{lj}y_{lj}$ can be got by equation (3-23).

5.3.2 Calculation of the weights of neighbors $\mathbf{w}_i[k_i]$

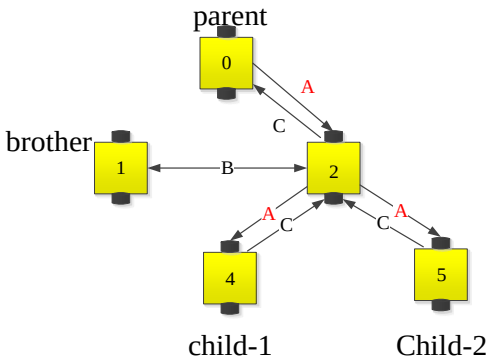
The weight of each neighbor (AUV-*j*) of AUV-*i* is computed according to the sub-digraph $G_i[k_i]$ composed by AUV-*i* and its neighbors. Since $G_i[k_i]$ is a time-variant digraph, the corresponding Laplacian matrix $\mathbf{L}_i[k_i]$ is also time-variant. As a result, the Markov chain associated with the discrete consensus algorithm is a non-homogeneous Markov chain, whose transition matrix $\mathbf{P}_i[k_i]$ ($\mathbf{P}_i[k_i]=e^{-\mathbf{L}_i[k_i]t}$) at k_{th} loop is also time-variant. The digraph is strongly connected with our hypothesis (see chapter 2), then the Laplacian matrix is irreducible according to lemma 2-1, so that the corresponding transition matrix $\mathbf{P}_i[k_i]$ is primitive. Based on lemma 5-1 and lemma 5-2, $\lim_{t\to\infty}\mathbf{P}_i[k_i]=\lim_{t\to\infty}e^{-\mathbf{L}_i[k_i]t}=\mathbf{1}\mathbf{w}_i[k_i]^T$. Thus, the elements of limited vector $\mathbf{w}_i[k_i]$ can be considered as the weights at k_{th} loop of AUV-*i*.

To guarantee the calculated weight vector $\mathbf{w}_i[k_i]$ are effective in each loop, each sub-digraph $G_i[k_i]$ should be constructed with different structures depending on the situations. To do so, two kinds of sub-digraphs are proposed: a family sub-digraph and a former-latter sub-digraph, illustrated in Fig. 5-1. If the found neighbors include the parent AUV, then the family sub-digraph is adopted, otherwise, the former-latter sub-digraph is adopted.

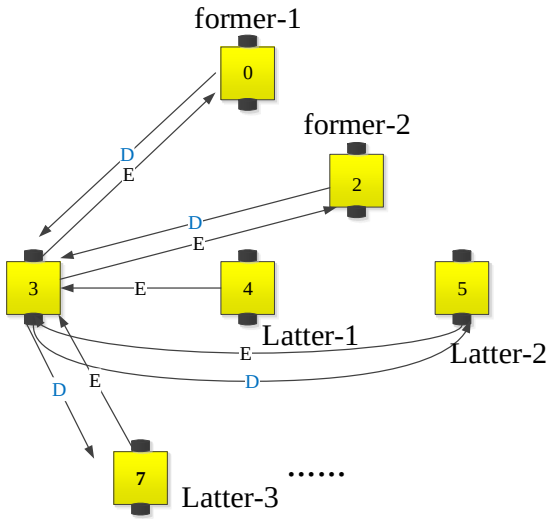
For the family sub-digraph (Fig. 5-1(a)), the weight of parent AUV (noted as *A*) must be set much larger than the weight of brother AUV (noted as *B*), meanwhile, the weight of brother AUV (*B*) is bigger than that of the child AUV (noted as *C*). This property is decided by the leader-follower model: to guarantee this model to be maintained well, the confidence of AUV to be followed is higher than that of follower. These confidences are the elements located on the *m*-row or *m*-volume of adjacent matrix $\mathbf{A}_i[k_i]$ corresponding to the sub-graph $G_i[k_i]$, *m* is the index of AUV-*i* in the matrix. Usually $m=1$ in the family sub-digraph, since the elements of $\mathbf{A}_i[k_i]$ are arranged with the orders: parent, AUV-*i* itself, brother and child, shown in equation (5-5), where $A?\ \ B>C$,

and the other elements of $\mathbf{A}_i[k_i]$ are equal to 0.

$$a_{1j} / a_{j1} = \begin{cases} A/C, \text{if} & \text{AUV-}j \text{ is the parent of AUV-}i \\ B/B, \text{if} & \text{AUV-}j \text{ is the brother of AUV-}i \\ C/A, \text{if} & \text{AUV-}j \text{ is the child of AUV-}i \end{cases} \quad (5\text{-}5)$$



(a) One family sub-digraph of AUV-2



(b) One former-latter sub-digraph of AUV-3

Fig. 5-1 Examples of two kinds of sub-digraphs (the number on the AUV represents the order of AUV in the pyramid distribution)

But there are some unsatisfying situations, in which, family sub-digraph leads to bad results: (1) the parent is not detected because of the overlapping among AUVs, or low quality of images, etc; (2) the relative position between AUV-*i* (child) and AUV-*j* (parent) has been in the tolerant error range ($\mathbf{p}_{ij} - \mathbf{p}_{ij}^{*} \to \boldsymbol{\varepsilon}$), while the relative position between AUV-*j* and its parent (AUV-*h*) is out of range ($\mathbf{p}_{jh} - \mathbf{p}_{jh}^{*}$? $\boldsymbol{\varepsilon}$). If the state is still updated according to the family sub-digraph, the updated state in situation (1) will become with large error or even not correct because the confidences of children and brothers are not higher than that of itself; the pyramid pattern built in situation (2) will fail because the AUV-*i* believes that it has arrived at the final position and stops.

To deal with these two situations, we replace the family sub-digraph by the former-latter sub-

digraph to generate a new Laplacian matrix, which makes it possible to build the pyramid pattern successfully and to check whether the AUV-*i* truly arrives at its destination or not yet. As presented in Fig. 5-1(b), the parent of AUV-3 (AUV-1) is not recognized, so that the AUVs of this sub-digraph are divided into two sets: the former set with the AUVs whose orders are smaller than itself (i.e. AUV-0, AUV-2 in Fig. 5-1(b)), and the latter set with the AUVs whose orders are larger than itself (i.e. AUV-4, AUV-5 and AUV-7). Then a new way to define a_{mj} or a_{jm} (the elements of $\mathbf{A}_i[k_i]$ are sorted according to the orders of AUVs in ascending order, *m* is changeable depending on the neighbors' orders) is proposed in equation (5-6).

$$a_{mj} / a_{jm} = \begin{cases} D / E, \text{if AUV-}j \text{ is in the former set of AUV-}i \\ E / D, \text{if AUV-}j \text{ is in the latter set of AUV-}i \end{cases} \quad (5\text{-}6)$$

Where, $D=q|od_s\text{-}od_f|$, od_s and od_f are orders of AUVs in the former/latter set and the order of itself respectively, q is a coefficient, $q \geq 1$, $D > E$. The other elements of $\mathbf{A}_i[k_i]$ are set equal to 0.

Subsequently, the Laplacian matrix $\mathbf{L}_i[k_i]$, $\mathbf{L}_i[k_i] = [l_{ij}] \in \mathrm{R}^{N_i[k_i] \times N_i[k_i]}$ corresponding to the adjacent matrix $\mathbf{A}_i[k_i]$ can be computed. Based on lemma 5-2, the normalized eigenvector $\mathbf{w}_i[k_i]$ of the eigenvalue 1 of matrix $\lim_{t\to\infty} e^{-\mathbf{L}_i[k_i]t}$ can be calculated, and $\sum_{j=1}^{N_i[k_i]} w_{ij}[k_i] = 1$.

For example, setting the confidences of equation (5-5) to $A = 20$, $B = 3$, $C = 1$, the adjacent matrix $\mathbf{A}_2[k_2]$, Laplacian matrix $\mathbf{L}_2[k_2] = \Delta[k_2] - \mathbf{A}_2[k_2]$ associated with Fig. 5-1(a) are respectively:

$$\mathbf{A}_2[k_2] = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 20 & 0 & 3 & 1 & 1 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{L}_2[k_2] = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -20 & 25 & -3 & -1 & -1 \\ 0 & -3 & 3 & 0 & 0 \\ 0 & -20 & 0 & 20 & 0 \\ 0 & -20 & 0 & 0 & 20 \end{pmatrix}$$

Subsequently, the weight vector associated with $\mathbf{L}_2[k_2]$ can be computed:

$$\mathbf{w}_2[k_2] = [0.9049, 0.0453, 0.0453, 0.0023, 0.0023]^T.$$

5.3.3 Formation control model based on displacement-based control method

As a preliminary approach, a linear discrete single-integrator system of *N*-AUVs is considered,

$$\overset{\vee}{\Delta p}_i[(k+1)_i] = \mathbf{u}_i[(k+1)_i] \qquad (5\text{-}7)$$

Where, $\overset{\vee}{\Delta p}_i[(k+1)_i] \in \mathrm{R}^n$, $\mathbf{u}_i[(k+1)_i] \in \mathrm{R}^n$ are respectively the relative position error and the control input of AUV-*i* at the $(k+1)_i$ loop. As we only consider 2D situation with all AUVs located at the same depth, $\overset{\vee}{\Delta p}_i[(k+1)_i] = (\Delta x_i[(k+1)_i], \Delta y_i[(k+1)_i])$, and $n = 2$.

Inspired by the model of fish schooling which includes three kinds of behaviors: approach, avoidance and parallel [68,69,70], we introduce two kinds of them (approach and avoidance) into our control input to avoid collisions among AUVs and obstacles. These two kinds of behaviors are embodied by the "$\pm$" of relative position errors. The control input $\mathbf{u}_i[(k+1)_i]$ can be estimated by equation (5-8).

$$\mathbf{u}_i[(k+1)_i] = \sum_{j=1}^{N_i[k_i]} w_{ij}[k_i](\mathbf{p}_{ij}[k_i] - \mathbf{p}_{ij}^*) \qquad (5\text{-}8)$$

In this case, we set $\left\|\mathbf{p}_{ij}^*\right\| \geq r$ to avoid collision. r is the radius of the neighborhood of each robot, we set r equal to the length of the AUV. If AUV-*j* is within the neighborhood of AUV-*i*, noted as $\left\|\mathbf{p}_{ij}[k_i]\right\| \leq \left\|\mathbf{p}_{ij}^*\right\|$, they will be repulsive to each other to preclude collision, otherwise, they will approach to build a pyramid pattern.

In the below part, we give a "theorem of convergence" and its proof.

Theorem 5-2 For the system (5-7) with the controller (5-8), the AUVs build a pyramid pattern successfully and asymptotically if all the sub-digraphs of AUVs G_i and the whole digraph $G(V, E)$ at every moment, are strongly connected even if they are time-variant.

Proof: Assume that each AUV is at least in the vision range of another one AUV and no AUV is broken. To make sure that each sub-digraph is strongly connected, if one AUV does not detect neighbors because of overlapping, distinguishing ID wrongly, etc., it will continuously and slightly adjust its position until it finds at least one neighbor. Then all the AUVs build the pyramid pattern depending on their sub-digraphs. Now, we will proof that the pyramid will converge according to theorem 2.

(1) Each sub-digraph of AUV-*i* has at least a spanning tree.

As each sub-digraph of AUV-*i* is strongly connected, it has at least a spanning tree depending on definition 2-5 (see subsection 2.1) and literature [129].

(2) Each sub-digraph of AUV-*i* can converge to an equilibrium state (the relative error is in a tolerant range).

As each sub-digraph of the AUV-*i* has at least a spanning tree, the matrix $\mathbf{L}_i[k_i]$ associated

with the sub-digraph should have exactly one zero eigenvalue based on lemma 5-1 and 5-2. Then the transition matrix is irreducible (see lemma 5-1) and also primitive according to lemma 5-2. Depending on theorem 5-1 of Markov chain (see the section 5.1), each sub-digraph of the AUV-*i* can converge to an equilibrium state.

(3) A mixture of finite convergent sub-digraphs is global convergent.

Each sub-digraph of AUV-*i* is convergent, which can be expressed by equation (5-9).

$$\lim_{t\to\infty}(\mathbf{P}_i[k_i]=e^{-\mathbf{L}_i[k_i]t})=\mathbf{v}_i[k_i]\mathbf{w}_i^T[k_i] \tag{5-9}$$

Where, $\mathbf{v}_i[k_i]=[1,1,...,1]^T_{N_i[k_i]\times 1}$. Assuming that the largest number of neighbors of all the sub-graphs of the AUV-*i* is $N_{i\max}$, then all the Laplacian matrices, and the associated transition matrices are noted as equation (5-10) and (5-11):

$$\hat{\mathbf{L}}_i[k_i]=\begin{pmatrix}\mathbf{L}_i[k_i] & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{pmatrix}_{N_{i\max}\times N_{i\max}} \tag{5-10}$$

$$\lim_{t\to\infty}e^{-\hat{\mathbf{L}}_i[k_i]t}=\hat{\mathbf{v}}_i[k_i]\hat{\mathbf{w}}_i^T[k_i]^T_{N_{i\max}\times N_{i\max}} \tag{5-11}$$

Where $\hat{\mathbf{v}}_i[k_i]=[1,1,...,1]^T_{N_{i\max}\times 1}$, $\hat{\mathbf{w}}_i[k_i]=[w_{i1}[k_i],w_{i2}[k_i],...,w_{iN_i}[k_i],0,...,0]_{1\times N_{i\max}}$. Consequently, the iteration of all the sub-graphs of the AUV-*i* can be represented by

$$\lim_{t\to\infty}\sum_{k_i=0}^{M_i}e^{-\hat{\mathbf{L}}_i[k_i]t}=\sum_{k_i=0}^{M_i}\hat{\mathbf{v}}_i[k_i]\hat{\mathbf{w}}_i^T[k_i]^T_{N_{i\max}\times N_{i\max}} \tag{5-12}$$

Where M_i denotes the number of all loops of AUV-*i* before reaching its convergence. As a result, a mixture of finite convergent sub-digraphs is still global convergent.

(4) Asynchronous behavior has no influence to global convergence

One precondition is given: the whole digraph $G(V,E)$ is strongly connected all the time, though the connected edges of the whole digraph are changeable because of the independences among all the agents.

As the equilibrium points of a linear consensus strategy are not relevant to the intermediate states but have relations with the initial state and the control inputs [132], the asynchronous behaviors between the AUV-*i* and its neighbors will not influence the convergence property.

Once all the AUVs converge to their equilibrium states, the initial graph can converge to a pyramid pattern, where $\mathbf{u}_i[k_i]<\varepsilon$, and $\mathbf{p}_i[k_i]$ converges to $\mathbf{p}_{ij}^*$ asynchronously.

5.3.4 Simulation results with the displacement-based control method

For a multi-AUV system, to build a pyramid formation with the displacement-based control method, the steps are summarized below:

(1) Searching neighbors and getting the neighbors' information: each CISCREA searches its neighbors with its on-board single camera in its neighborhood.

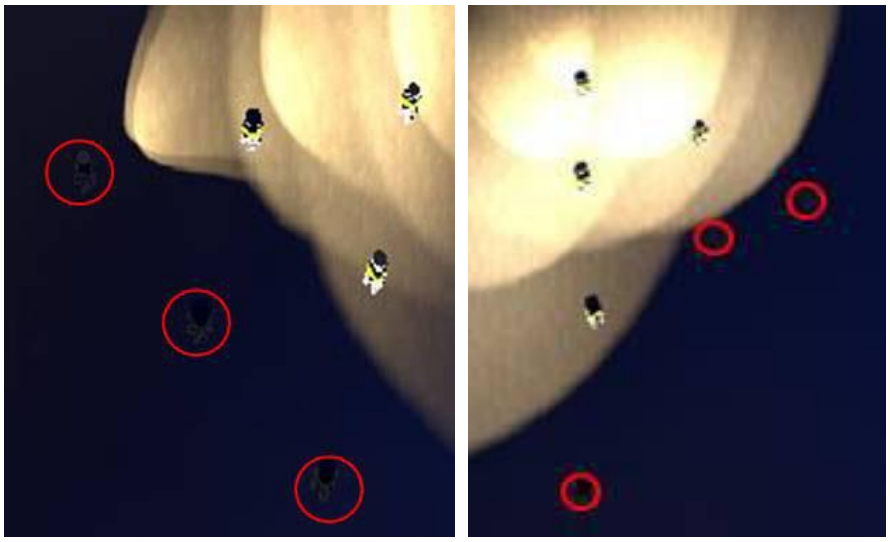
(2) Sharing neighbors' information and constructing the common frame: each CISCREA shares the neighbors' information by acoustic communication to get all the neighbors' information of all the CISCREAs, then each CISCREA can construct the common frame and estimate the initial distributions.

(3) Identifying the expected position relationship among AUVs: the method based on PCA (see section 4.2.1) is adopted to identify and optimize the pyramid distribution, then the expected position relationship can be obtained with the match-up of the positions in both initial and pyramid distributions achieved according to the collision avoidance strategy (see section 4.2.2).

(4) Updating the trajectories: the leader moves to its destination along the straight trajectory as a reference position, without trajectory updating; the other CISCREAs repeat the loop: "search, compute, move" independently and asynchronously until all of them arrive at their destinations.

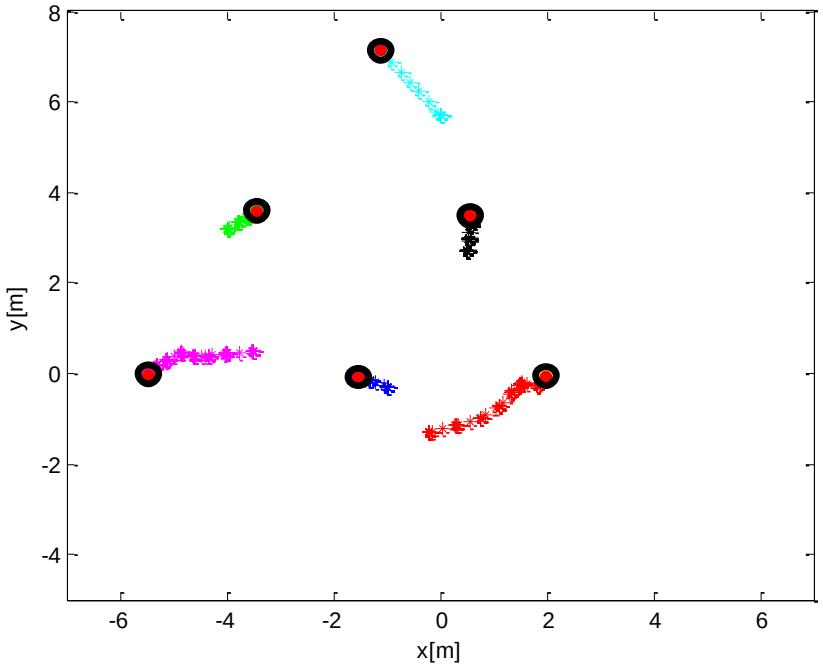
When all the CISCREAs reach their destinations in the pyramid pattern, the pyramid pattern is built successfully.

Limited by the configuration of my PC, at most 7 CISCREAs can be used to simulate the formation building in Blender. Fig. 5-2 and 5-3 present two built pyramid patterns as examples with 6 and 7 CISCREAs, and their trajectories of the CISCREAs. Similarly, the CISCREAs located at the last row are hardly to be seen.

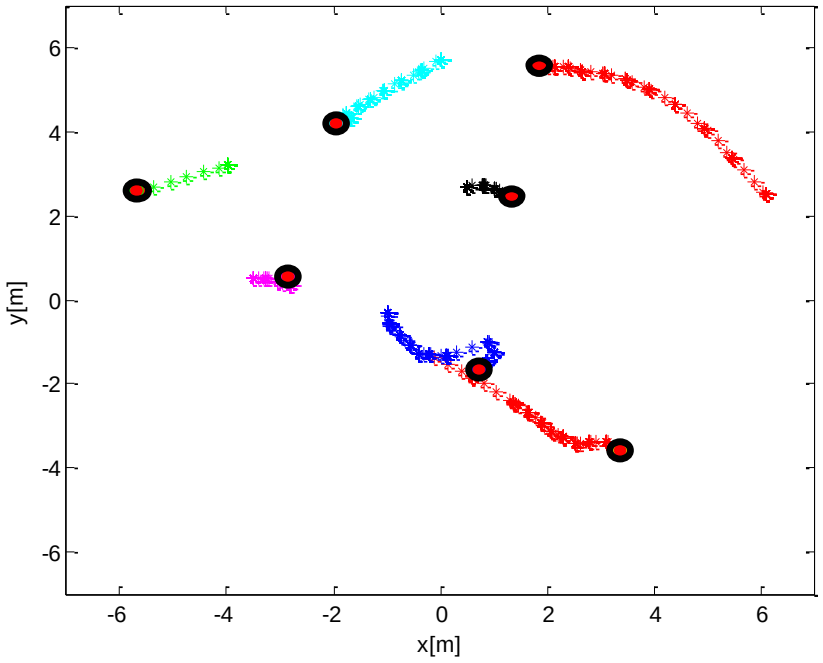


(a) with 6 CISCREAs (b) with 7 CISCREAs

Fig. 5-2 Two examples of pyramid built with the displacement-based control method



(a) Trajectories of 6 CISCREAs of Fig. 5-2(a)



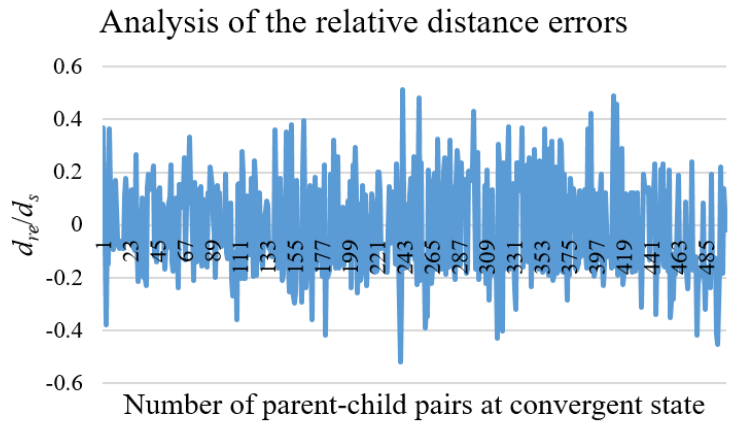
(b) Trajectories of 7 CISCREAs of Fig. 5-2(b)

Fig. 5-3 Trajectories of the AUVs of Fig. 5-2 during the pyramid building

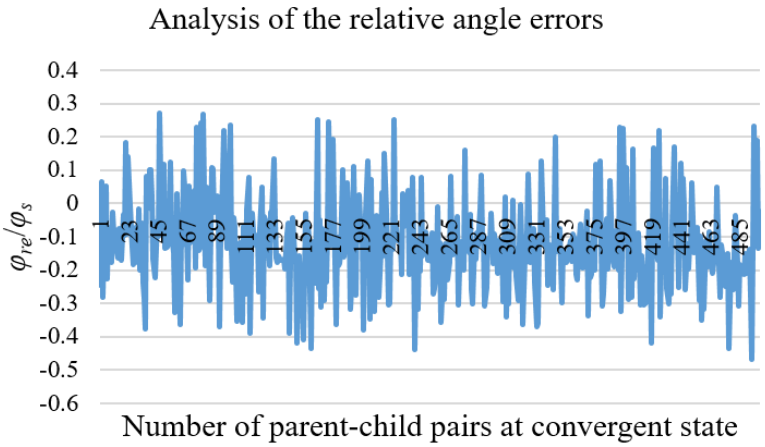
To evaluate the performance of the formation building based on the discrete consensus algorithm associated with the displacement-based control method, the relative position errors x_{re}, y_{re}, the relative distance error d_{re} and relative angle error φ_{re} between the parent and child

computed by equation (4-10) are also introduced. We repeat the simulations more than 100 times with 6 AUVs to estimate the precision of pyramid patterns built with the asynchronous discrete consensus algorithm. In the 100 times, the initial distributions of 6 AUVs are randomly given in a bounded range. The relative error ratios $\frac{d_{re}}{d_s}$ and $\frac{\varphi_{re}}{\varphi_s}$ at the convergent states are presented in Fig. 5-4.

Besides, we also simulate the experiments another 100 times with positions in Blender to estimate the contribution of the image processing step and movement step to the errors. The comparison results of relative errors using the relative positions from the image process $*\_img$ and from the Blender $*\_blender$ are presented in table 5-1.



(a)



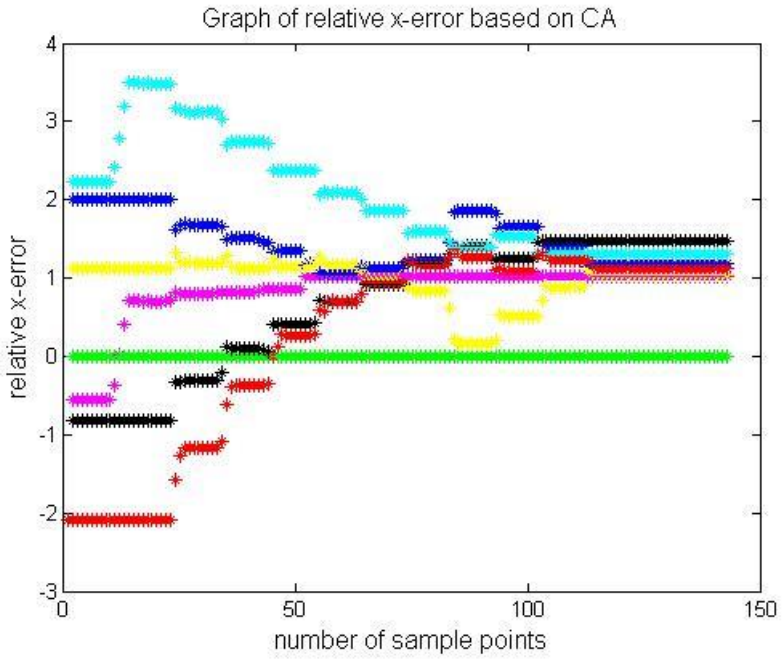
(b)

Fig. 5-4 Relative error ratios of distances $\frac{d_{re}}{d_s}$ and angles $\frac{\varphi_{re}}{\varphi_s}$ at convergent states based on displacement-based method by 6 AUVs with the relative positions obtained by image processing.

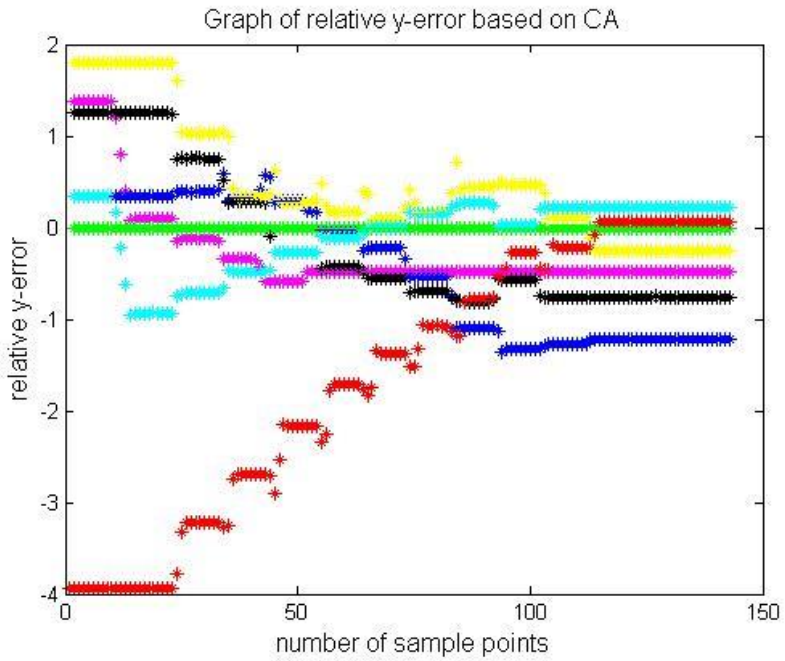
Table 5-1 Performance of the displacement-based method. For each error-criterion, we give the average, standard deviation, the maximum and the minimum in two cases (with and without image processing step)

	$\lvert x_{re}/x_s \rvert$	$\lvert y_{re}/y_s \rvert$	$\lvert d_{re}/d_s \rvert$	$\lvert \varphi_{re}/\varphi_s \rvert$
aver_img	0.1239	0.2004	0.1965	0.1783
stdev_img	0.0982	0.1298	0.1020	0.1037
max_img	0.4823	0.7696	0.6687	0.5210
min_img	0.0011	0.0014	0.0190	0.0003
aver_blender	0.1185	0.1418	0.1493	0.1447
stdev_blender	0.0929	0.0894	0.0667	0.0898
max_blender	0.5651	0.5423	0.4719	0.5782
min_blender	0.0002	0.0007	0.0149	0.0008

To present the error changing trends during the process of pyramid building, the relative position errors (x_{re}, y_{re}) associated with Fig. 5-2(b) and Fig. 5-3(b) are shown in Fig. 5-5. The drawn points are sampled by taking one point every 100 points from original data. In each loop, the time of moving to the new position is a small part compared with the time of searching neighbors. Therefore, the sampled movement points are much less than the sampled points in the searching period, which leads to that the curves presented look like piecewise functions.



(a) The convergent process of x_{re}



(b) The convergent process of y_{re}

Fig. 5-5 Relative position errors (x_{re}, y_{re}) between parent AUV and AUV itself based on the displacement-based method. The points are sampled by taking one point every 100 points from original data. In each loop, the sampled movement points are much less than the sampled points in the searching period because of the short movement time. Therefore, most sample points are the points of searching neighbors, that is the reason of gaps: only a few of movement points are extracted as sample points.

To verify the performance of the proposed algorithm in the obstacle environment, the asynchronous discrete consensus algorithm is also tested in the obstacle environment in Matlab. Here, we assume the obstacles can be detected and the relative positions between obstacles and AUVs can be estimated by AUVs. And the weights of obstacles w_{ob} are all set to $w_{ob} = -1$, and the avoidance distance threshold is set as $d_{ob}^{*} = l_{AUV}$, where l_{AUV} is the length of AUV. Then in the same way, we set the relative distance noises generated by image processing to $\delta_d \in \pm 10\% d_s$, and set the kinetic noises and hydrodynamic noises to $\delta_{td} \in \pm 10\% d_s$, $\delta_{t\varphi} \in \pm 10\% \varphi_s$. One example built by 6 robots with 2 obstacles is presented in Fig. 5-6.

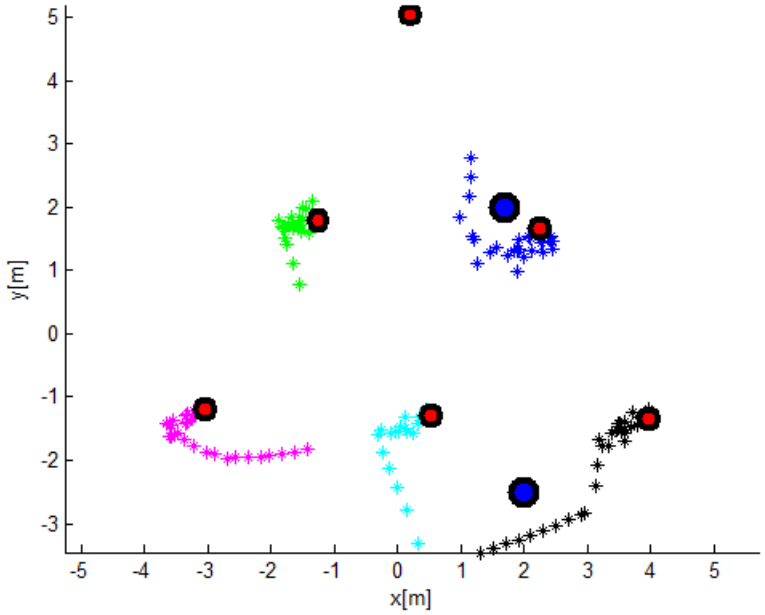


Fig. 5-6 One example built by 6 robots with 2 obstacles (the red points are the final positions of AUVs, while the blue points are the positions of two obstacles) with the relative distance noises generated by image processing $\delta_d \in \pm 10\% d_s$, and the kinetic noises and hydrodynamic noises $\delta_{td} \in \pm 10\% d_s$, $\delta_{t\varphi} \in \pm 10\% \varphi_s$.

From Fig. 5-5 and table 5-1, it implies that the convergent errors are large. Though the relative errors got by directly reading the accurate positions from Blender are slightly smaller than those estimated by the image processing, the errors are still large, more than 50%. One reason is that to save time by reducing the loop times of each AUV, the allowed error range is set big. Besides that, except for the image processing, the kinematic control, AUV hydrodynamic uncertainty also have contributions to the large convergent errors. From Fig. 5-6, we find that the pyramids can be built based on the proposed algorithm associated with the relative position information in the obstacle environment as long as the obstacles can be detected and the relative positions between robots and obstacles can be estimated.

5.4 The trajectory update based on the consensus algorithm with the distance-based method

When only cameras are available without compasses, the displacement-based control method described in section 5.3 is invalid to build a pyramid. In this case, a discrete consensus algorithm with the distance-based control method is proposed to update the trajectories of robots. The same as the section 5.3, the asynchronous discrete consensus algorithm with weighted time-varied digraph

is used to build the pyramids depending on the distance-based method, shown in equation (5-13).

$$\mathbf{e}_i[(k+1)_i] = \sum_{j=1}^{N_i[k_i]} w_{ij}[k_i](\mathbf{d}_{ij}[k_i] - \mathbf{d}_{ij}^*) \qquad (5\text{-}13)$$

Where, $\mathbf{d}_{ij}[k_i] - \mathbf{d}_{ij}^*$ is the relative distance error between AUV-*i* and AUV-*j* at the k_{th} loop of AUV-*i*, $\mathbf{d}_{ij}^*$ and $\mathbf{d}_{ij}[k_i]$ are respectively the expected distance vector and measured distance vector at the k_{th} loop. $\mathbf{e}_i[(k+1)_i]$ is the updated trajectory of AUV-*i*. According to that any vector in a plane can be expressed by at least two non-relevant vectors, to be able to calculate the distance vector $\mathbf{d}_{ij}[k_i]$, $N_i[k_i]$ should meet the condition: $N_i[k_i] \geq 2$, as illustrated in Fig. 5-7. To get $\mathbf{e}_i[(k+1)_i]$, the weights $w_{ij}[k_i]$ and the relative distance error $\mathbf{d}_{ij}[k_i] - \mathbf{d}_{ij}^*$ have to be estimated.

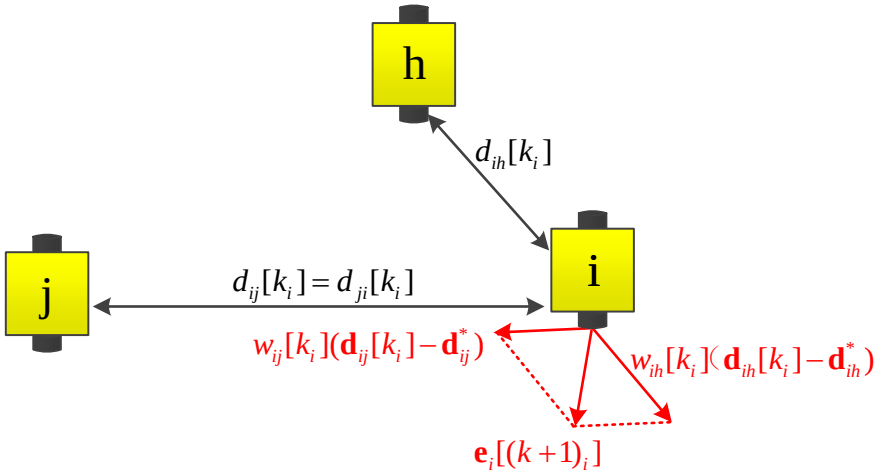


Fig. 5-7 Illustration of the trajectory update of AUV-*i* with the distance-based method. In the k_{th} loop, AUV-*i* finds two neighbors AUV-*j* and AUV-*h*, then its trajectory $\mathbf{e}_i[(k+1)_i]$ can be updated according to equation (5-13).

5.4.1 Calculation of the distance vector errors

Without the compass, each AUV is not able to estimate its global orientation. But the relative distances $d_{ij}[k_i]$ and relative angles $\varphi_{ij}[k_i]$ can still be estimated from images according to equations (3-19) and (3-20). Then $\mathbf{d}_{ij}[k_i] - \mathbf{d}_{ij}^*$ can be computed by equation (5-14).

$$\mathbf{d}_{ij}[k_i] - \mathbf{d}_{ij}^* = \frac{\mathbf{d}_{ij}[k_i]}{d_{ij}[k_i]}(d_{ij}[k_i] - d_{ij}^*) \qquad (5\text{-}14)$$

Where $\mathbf{d}_{ij}[k_i]=\begin{pmatrix} d_{ij}[k_i]\cos\varphi_{ij}[k_i] \\ d_{ij}[k_i]\sin\varphi_{ij}[k_i] \end{pmatrix}$.

5.4.2 Calculation of the weights $\mathbf{w}_i[k_i]$

To get the weight of each neighbor, a different former-latter sub-digraph is introduced (see Fig. 5-8). Different from section 5.3, the former set only includes the AUVs located in the former layers of AUV-*i* (the concept of layer is defined in Fig. 2-3, and the layer can be computed according to equation (4-5)). The elements of the latter set are the AUVs located in the latter layers of AUV-*i*. The other AUVs located in the same layer of AUV-*i* make up a new set, noted as middle set. As previously described, the weights of the elements in the former set are higher than those in the middle set, while the weights of the elements in the middle set are higher than those in the latter set. The equation (5-15) re-defines the values of the elements of adjacent matrix $\mathbf{A}_i[k_i]$.

$$a_{ij}/a_{ji}=\begin{cases} B/C, \text{if AUV-}j \text{ is in the former layer of AUV-}i \\ B/B, \text{if AUV-}j \text{ is in the same layer of AUV-}i \\ C/D_{ij}, \text{if AUV-}j \text{ is in the latter layer of AUV-}i \\ 0, \text{the others} \end{cases} \quad (5\text{-}15)$$

However, since the state of each robot is the distance vector error, the influences of robots in the former set on the updated trajectory are not much higher than those in the middle set and those in the latter set. As a result, we set the value of B not too large, and $B \geq C$. Thus, in equation (5-15), $B=\min\{N_i[k_i]-1, q\}$, $D_{ij}=|i-j|*q$, and $1 \leq q \leq 3$, usually $q=1$. For example, the adjacent matrix $\mathbf{A}_2[k_2]$ associated with Fig. 5-8 is shown below:

$$\mathbf{A}_2[k_2]=\begin{pmatrix} 0 & 0 & C & 0 & 0 & 0 \\ 0 & 0 & B & 0 & 0 & 0 \\ B & B & 0 & C & C & C \\ 0 & 0 & D_{24} & 0 & 0 & 0 \\ 0 & 0 & D_{25} & 0 & 0 & 0 \\ 0 & 0 & D_{27} & 0 & 0 & 0 \end{pmatrix}$$

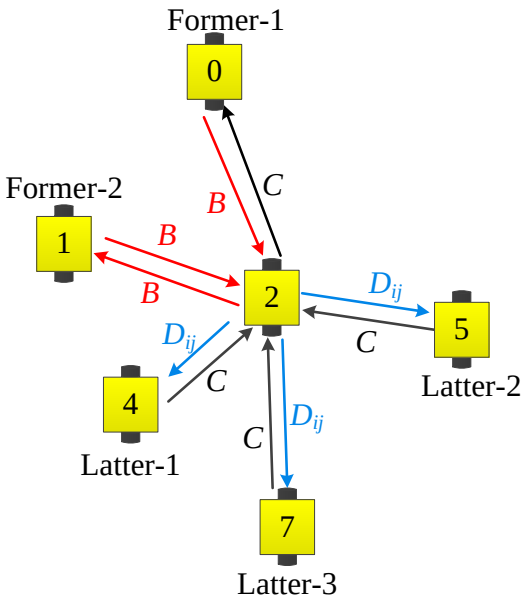


Fig. 5-8 One former-latter sub-digraph of AUV-2 in the k_2 loop in the case of the approach based on the distance-based control method

In the same way as in section 5.3.2, the limited transition matrix $\mathbf{P}_i[k_i]$ ($\lim_{t\to\infty} e^{-\mathbf{L}_i[k_i]t} = \mathbf{1}\mathbf{w}_i[k_i]^T$) can be obtained depending on the adjacent matrix, and the elements of $\mathbf{w}_i[k_i]^T$ are regarded as the weights $w_{ij}[k_i]$ of equation (5-13).

Considering that $\mathbf{e}_{ii} = \mathbf{0}$, and the weight of a parent (noted as the AUV-*n*) should be the largest, the weight $w_{in}[k_i]$ is set to $w_{in}[k_i] = w_{in}[k_i] + w_{ii}[k_i]$. For example, the variables of equation (5-15) are set to $q = 1$ and $C = 1$, then the Laplacian matrix $\mathbf{L}_2[k_2]$ and weight vector $\mathbf{w}_2[k_2]$ associated with the adjacent matrix $\mathbf{A}_2[k_2]$ and Fig. 5-8 are presented below:

$$\mathbf{L}_2[k_2] = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ -1 & -1 & 5 & -1 & -1 & -1 \\ 0 & 0 & -2 & 2 & 0 & 0 \\ 0 & 0 & -3 & 0 & 3 & 0 \\ 0 & 0 & -5 & 0 & 0 & 5 \end{pmatrix}$$

$$\mathbf{w}_2[k_2] = [0.2484, 0.2484, 0.2484, 0.124, 0.0826, 0.0496]$$

Considering that AUV-0 is the parent of AUV-2, then $w_{20}[k_2] = w_{20}[k_2] + w_{22}[k_2]$, and $w_{20}[k_2] = 0.4968$.

5.4.3 Formation control model based on distance-based control method

Similar to the case of a displacement-based control, a linear discrete single-integrator system of *N*-AUVs is considered, and equation (5-13) is re-written as:

$$\overset{\cdot}{\Delta} d_i[(k+1)_i] = \mathbf{u}_i[(k+1)_i] \tag{5-16}$$

$$\mathbf{u}_i[(k+1)_i] = \sum_{j=1}^{N_i[k_i]} w_{ij}[k_i](\mathbf{d}_{ij}[k_i] - \mathbf{d}_{ij}^*) \tag{5-17}$$

Where, $\overset{\cdot}{\Delta} d_i[(k+1)_i]$ is the estimated updated trajectory of AUV-*i* in its $(k+1)_{th}$ loop. $\mathbf{u}_i[(k+1)_i]$ is the updated control input of AUV-*i*, which can be computed by equation (5-13).

We given below a theory and its proof, which allows us to affirm that the pyramid pattern can be achieved depending on the system (5-16) and control input (5-17).

Theorem 5-3 For the system (5-16) and the control input (5-17), the AUVs can build a given pattern asymptotically and asynchronously with our system structure if the digraph including all the robots is strongly connected at every moment even if the sub-digraphs are time-variant.

Proof:

(1) In each k_i loop, AUV-*i* will gradually approach its final position with its strongly connected sub-digraphs.

The formation building process based on the asynchronous discrete consensus algorithm via the distance-based control method is also regarded as a Markov chain, the convergence of each robot to its destination has been proofed in section 5.3.3. As a result, each AUV-*i* will approach its destination asynchronously.

(2) The AUVs will finally converge to the given pattern.

Though the connected edges of the whole digraph G are changeable because of the movement of all the AUVs, the whole digraph G is strongly connected according to the assumption. This is the precondition of the convergence of the pyramid pattern.

For each AUV, after finite loops k_{mi}, they will arrive in the neighborhood of their final position with the proof of step 1. Then in the next loops ($k_i > k_{mi}$), the found neighbors will be fixed, which will compose fixed strongly connected sub-digraphs with a constant irreducible and aperiodic transition matrix. Then according to the theorem 5-1 in section 5.1, the AUVs will converge to the given pattern.

(3) The asynchronous behaviors among AUVs have no influence to the convergence of the pyramid pattern.

As mentioned in [131], with the linear consensus algorithm, only the initial condition of a vertex contributes to the equilibrium value if and only if the vertex has a directed path to all the other nodes in the graph. This implies that only the initial states have influences on the equilibrium

state, so that the asynchronous behaviors are uncorrelated to the convergent property.
Therefore, after each AUV converges to its final position, the pyramid pattern will be built.

The update steps of multi-AUV system are summarized below (see Fig. 5-6 as an example):

(1) AUV-*i* searches its neighbors around itself by its video camera. When finding one AUV-*j*, its relative distance $d_{ij}[k_i]$ and relative angle $\varphi_{ij}[k_i]$ are calculated according to the equations (3-19) and (3-20). Then with all the relative distances, the AUV-*i* will judge whether it has arrived at its destination or not by the equation (5-24).

$$\delta d_i[k_i] = \sum_{j=1}^{N_i[k_i]} (d_{ij}[k_i] - d_{ij}^*) < \varepsilon_i \qquad (5\text{-}24)$$

Where, ε_i is the given error threshold, associated with $N_i[k_i]$.

If equation (5-24) is satisfied, AUV-*i* believes that it has arrived at its destination in the pyramid pattern. Otherwise, it will continuously update its trajectory $\dot{\Delta} d_i[(k+1)_i]$ by following the steps (2)-(4).

(2) The sub-digraph-*i* $G_i[k_i]$ is constructed, and the associated weights $\mathbf{w}_i[k_i]$ are computed.

(3) All the distance vector errors $\mathbf{d}_{ij}[k_i] - \mathbf{d}_{ij}^*$ ($j = 1, ..., N_i$) are calculated with equation (5-14) according to the relative distances $d_{ij}[k_i]$ and the relative angles $\varphi_{ij}[k_i]$.

(4) Then $\dot{\Delta} d_i[(k+1)_i]$ can be updated with the equations (5-16) and (5-17). With the updated trajectory, AUV-*i* can move to a new position along this updated trajectory (a short line). Then according to the steps (1)-(4), AUV-*i* runs its next loop.

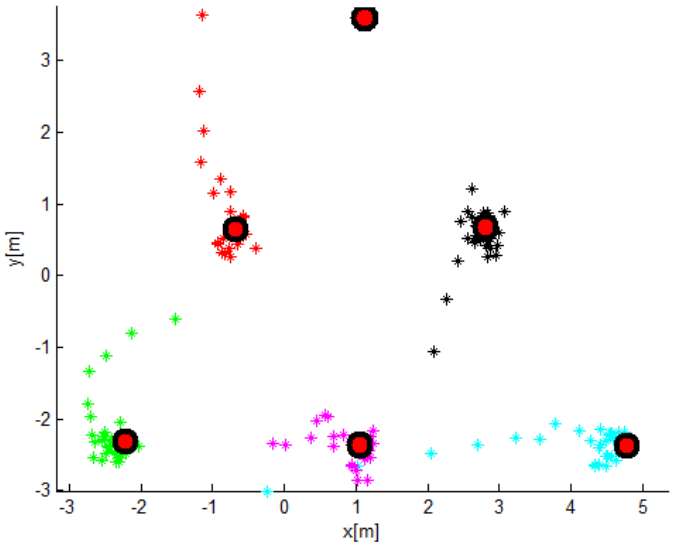
In the same way, the other AUVs can also get their updated states.

5.4.4 Simulation results with the distance-based control method

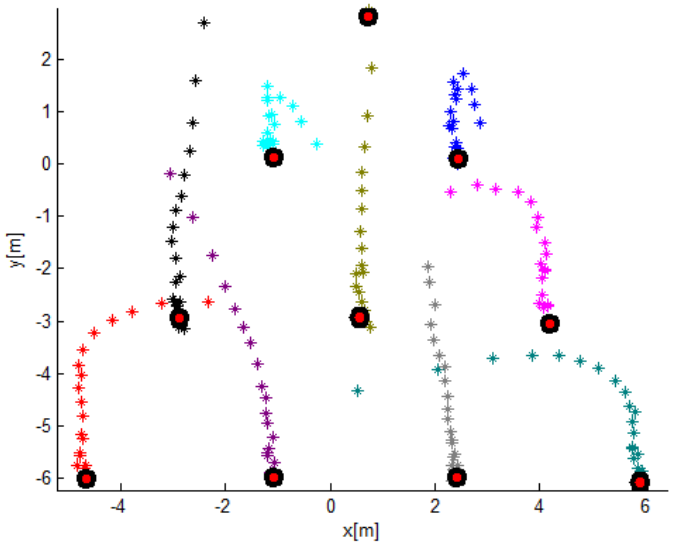
With the same reason of section 5.3.4, at most 7 CISCREAs are used to simulate the formation building in Blender. Compared with Blender, the simulations can be done with more AUVs in Matlab to verify the performance of the proposed formation control method, though the simulation environment constructed in Matlab is just a 2D environment, and the process of image processing cannot be simulated. Thus, the simulations in this section are done in Matlab with more AUVs, where one point is used to stand for an AUV. The expected relative distance and relative angle between each AUV and its parent are also set to: $d_s = 3.5m$, $\varphi_s = 30^\circ$. With the assumption that the depth of focus is 5m, the AUVs can see each other when $d_{ij} \le 5m$.

All the relative distances in the initial distributions are set to within a bounded range for meeting the condition: each of AUV can be detected by at least one of the other AUVs. The formation building steps are given in section 5.4.3. Referencing to the information errors extracted

by image processing in Blender, and the added movement errors of AUVs described in section 4.3 and 5.3.4, we add the relative distance errors generated by image processing ($\delta_d \in \pm 10\% d_s$), and the movement control noises ($\delta_{td} \in \pm 10\% d_s, \delta_{t\varphi} \in \pm 10\% \varphi_s$). According to the errors of section 5.3.4, the convergent conditions are set to: $d_{re} = 40\% d_s$, $\varphi_{re} = 40\% \varphi_s$ (The same as in section 5.3.4, considering that the AUVs will continuously adjust their positions when moving on to accomplish the tasks, the convergent errors are set large to reduce the time consumption). Fig. 5-9 presents two simulation results with 6 and 10 AUVs respectively in Matlab.



(a) The pyramid built with 6 AUVs

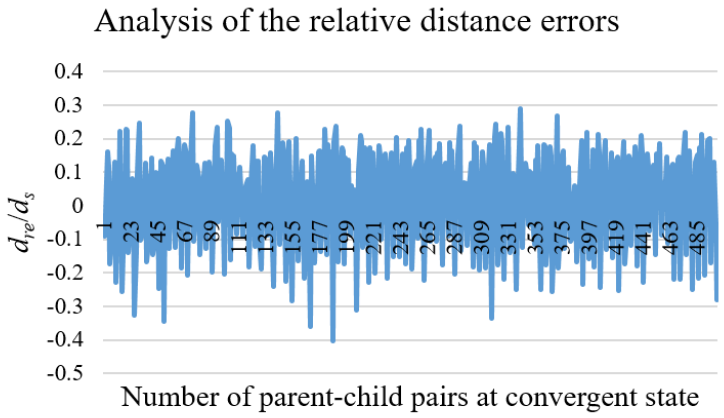


(b) The pyramid built with 10 AUVs.

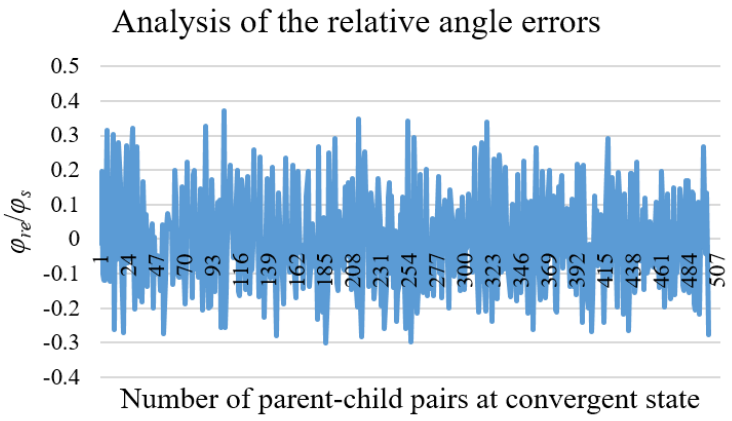
Fig. 5-9 Trajectories of AUVs from initial state to final state based on the asynchronous discrete consensus algorithm associated with the distance-based control method with the convergent condition within $\pm 40\%$ in Matlab. In the simulations, we set the relative distance noises generated

by image processing to $\delta_d \in \pm 10\% d_s$, and set the kinetic noises and hydrodynamic noises to $\delta_{td} \in \pm 10\% d_s$, $\delta_{t\varphi} \in \pm 10\% \varphi_s$.

To evaluate the performance of the discrete consensus algorithm with the distance-based control method, the relative distance error ratio $\frac{d_{re}}{d_s}$ and relative angle error ratio $\frac{\varphi_{re}}{\varphi_s}$ between AUV and its parent AUV got by equation (4-10) are introduced. With the noises $\delta_d, \delta_{td}, \delta_{t\varphi}$, 6 CISCREAs are used to repeatedly build the pyramid 100 times, the error ratios $\frac{d_{re}}{d_s}$ and $\frac{\varphi_{re}}{\varphi_s}$ are measured at the convergent state are given in Fig. 5-10.



(a) The relative distance errors



(b) The relative angle errors

Fig. 5-10 The relative distance and angle error ratios ($\frac{d_{re}}{d_s}$ and $\frac{\varphi_{re}}{\varphi_s}$) between AUVs and their parent-AUVs at the convergent state based on the discrete consensus algorithm with the distance-

based control method

From Fig. 5-10, we note that both relative error ratios at the convergent state are within 40%, which means that all the pyramids are built successfully in the 100 times. The accuracy of this method is similar to the asynchronous discrete consensus algorithm with the displacement-based control method.

Actually, the accuracy of the built pyramid can be higher by setting the allowed error to be slightly smaller. In this case, each AUV needs more loops to update its trajectory and needs more time to reach the convergent state. Then in Matlab, we set the convergent condition within $\pm 20\%$ to build the pyramids. The AUVs can achieve a convergent pyramid pattern, one example is shown in Fig. 5-11, but the time consumption is about 3 times of the pyramid building with the convergent condition within $\pm 40\%$. Similar with the Fig. 5-9(a), Fig. 5-11 shows the built pyramid by 6 AUVs with the convergent condition within $\pm 20\%$, where the time consumption is 3.2 times of that with the convergent condition within $\pm 40\%$. In this case, the method will lose its practical value. Further, considering that the relative distances among AUVs will be continuously adjusted to maintain the formation, then the allowed relative errors can be set a bit of large. As a result, some accuracy is sacrificed to save time.

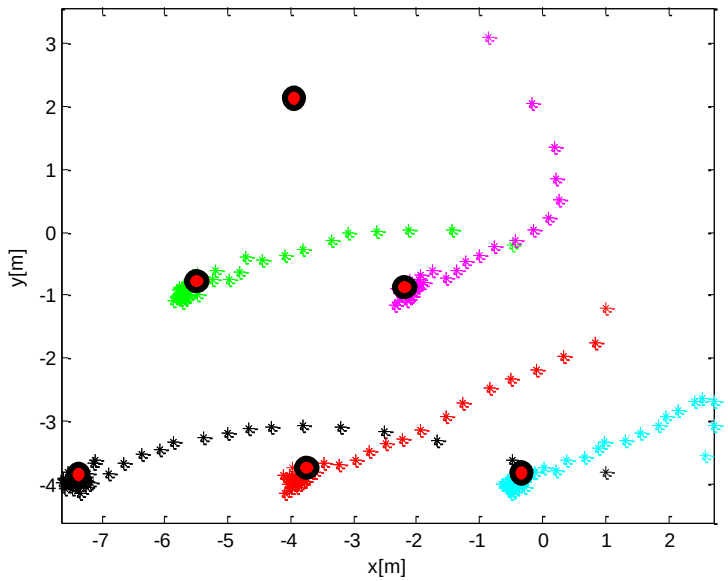


Fig. 5-11 One example of the built pyramid with 6 AUVs with the convergent condition within $\pm 20\%$

Similar to the section 5.3.4, the performance of the asynchronous discrete consensus algorithm with the distance information is also tested in the obstacle environment. The weights of obstacles w_{ob} are also set to $w_{ob} = -1$, and the avoidance distance threshold is set as $d_{ob}^{*} = l_{AUV}$. Then

adding the noises $\delta_d \in \pm 10\% d_s$, $\delta_{td} \in \pm 10\% d_s$, one example built by 6 robots with 2 obstacles is presented in Fig. 5-12.

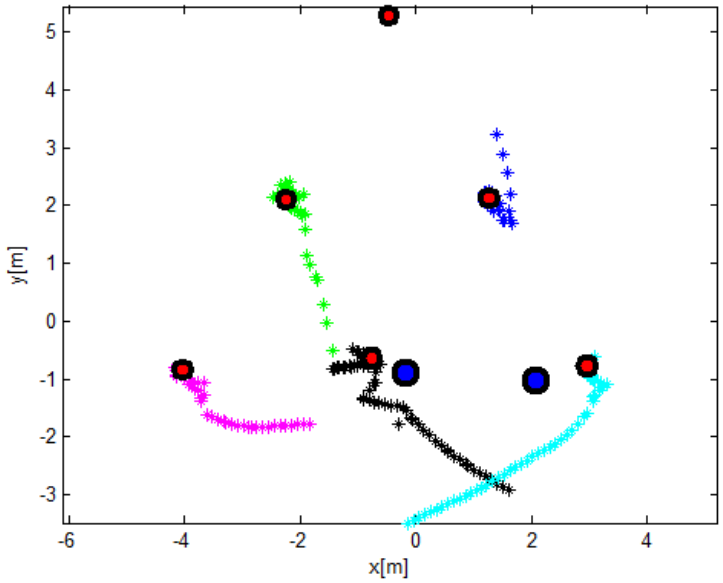


Fig. 5-12 One example built by 6 robots with 2 obstacles with the relative distance noises generated by image processing $\delta_d \in \pm 10\% d_s$, and the kinetic noises and hydrodynamic noises $\delta_{td} \in \pm 10\% d_s$, $\delta_{t\varphi} \in \pm 10\% \varphi_s$.

From Fig. 5-12, we notice that the pyramids can be built based on the proposed algorithm associated with the relative distance information in the obstacle environment as long as the obstacles can be detected and the relative distances between robots and obstacles can be estimated.

5.5 Chapter summary

Due to the regular unknown environment, an asynchronous discrete consensus-based formation control method with local information is proposed based on the visual system. With the expected formation relationship among AUVs, the AUVs can update their trajectories independently based on the information sensed by its on-board visual system until the formation building is achieved.

For the formation building in general unknown environment, the consensus algorithm is introduced. Depending on the obtained neighbors' information (global relative positions or local distance vector), two kinds of asynchronous formation control methods (the displacement-based and distance-based methods) are studied to complete the formation building. Combining with the properties of the multi-AUV system in this thesis, each AUV is always in a dynamic state, then the topology constituted by each AUV and its neighbors can be described by a time-variant digraph, then the discrete consensus algorithm with the weighted time-variant digraph is proposed. Moreover,

the trajectory update (the next state) is only relevant to the current state, then the first-order non-homogeneous Markov chain is introduced to identify the weights of neighbors. To guarantee the effectivity of the weights, two kinds of sub-digraphs are given for different situations: family and former-latter structures.

Based on the information mentioned above, the theorems of formation building associated with the displacement-based and distance-based control methods are proposed respectively. And we have proven that the formation building will be successful with these two kinds of asynchronous discrete consensus algorithms when the topologies constituted by all the AUVs are strongly connected, even if the topology is weighted time-variant. The feasibility and stability of the methods have been verified in Blender and Matlab.

6 Experiments and analysis of the formation building

In the simulation environments of Blender and Matlab, the proposed coordination strategy has been verified fully. The experiments of the formation building in the real context are mainly to test and verify the feasibility of the coordination strategy, to evaluate its performance and to provide some information for further study. As the experiments at sea with underwater vehicles are extremely complex to perform, an intermediate experimental assessment of the method is done on land in this thesis.

In this thesis, we focus on the formation structure, information extraction from images and formation control, not on the communication problem and the kinematic control of AUV in depth. The indoor environment is well-suited to evaluate the image processing methods (recognition algorithm, information extraction method) and the formation coordination strategy with the shared local visual information. As a result, the experiments are carried out with three humanoid robots (NAO robots) in doors with limited communication (only once by wifi) to test the recognition of robot, the extraction of local information, and the formation control experiments with the shared information for simulating the experiments of multi-AUV system in the given depth (2D environment).

6.1 Presentation of the experimental platform of humanoid robot

To get as close as possible to the underwater conditions, we realize the experiments in these conditions: limited communication frequency, limited communication distance, artificial lighting noise and single vision system to sense the surroundings etc. 3 humanoid robots (NAO robots, shown in Fig. 6-1) are used to build the pyramid pattern (actually a triangular pattern) for evaluating the performance of the coordination strategy. All the NAOs are totally identically equipped with V4 version. Implicit IDs are used to recognize the NAOs.

To simulate the limited communication situation and the formation building based on the vision system, the forehead camera of NAO is regarded as the main sensor to percept the surroundings, which is used: (1) to extract the neighbors' local information; (2) to provide visual feedback for the kinematic control of NAOs. Assuming that acoustic communication has been achieved, to simply the process, the NAOs do not directly communicate with each other, but share the information on a center computer (Intel(R)Xeon(R)CPU(4processors) at 3.1GHz) by WiFi.

The experiments are done in a big lab with many features in surroundings, shown in Fig. 6-2. Considering that the NAOs have no compasses, we only evaluate the performances of the local position-based control method and the asynchronous limited communication control method with the distance-based control to construct the pyramid formation, since these two methods do not need to estimate the global orientation in real-time.

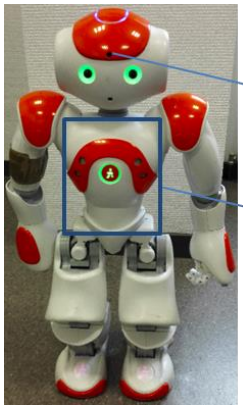
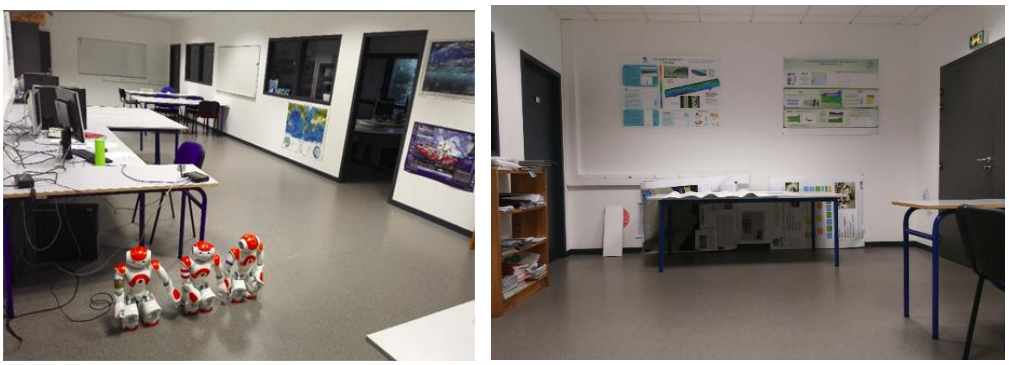


Fig. 6-1 NAO humanoid robot



(a) (b)

Fig. 6-2 Experimental environment with enough space for robots to move and many features for the positioning of robots

6.2 Acquirement of the neighbor information for NAOs

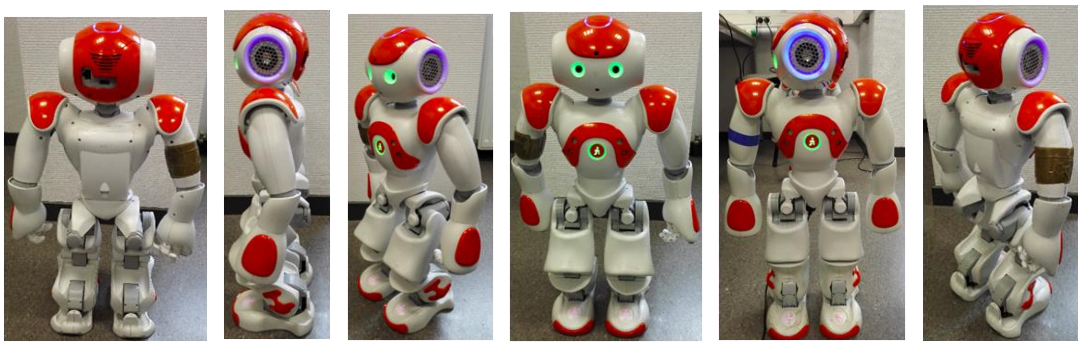
As described as in chapter 3, each NAO needs to percept its neighbors by its on-board camera, which means the information is extracted mainly depending on the forehead camera of each NAO (see Fig. 6-1). The parameters of the forehead camera are shown in table 6-1. Then image processing becomes a significant part to get the local information.

Table 6-1 The parameters of NAO forehead camera

Version	V4.0
Camera model	MT9M114
Camera output performance	960p@30fps(YUV422 color space)
View field	72.6° DFOV(60.9° HFOV, 47.6° VFOV)
Range of focus	30cm-infinite
Type of focus	Fixed focus

6.2.1 Recognition of NAOs

Different from the CISCREA, the recognition method of NAOs should be modified, since the different visual features will be extracted when a NAO is located at different orientation θ (the angle between the torso and the north orientations) with different headings φ (the angle of head yaw). Fig. 6-3 presents some examples of the visual features of the same NAO located at different orientations. To recognize the NAOs correctly, a mask-base (M-base) is created to store these different features. Then a 2D cross-correlation is used to select the mask in M-base, which is the most similar to the visual features extracted from the current image [133].

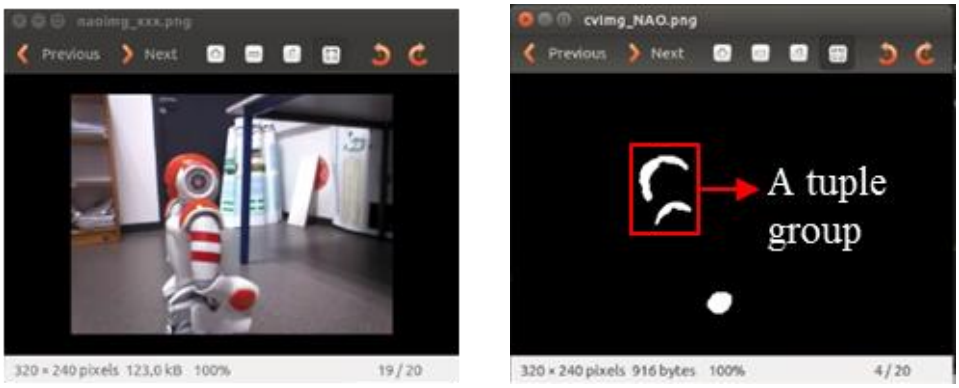


(a) (b) (c) (d) (e) (f)

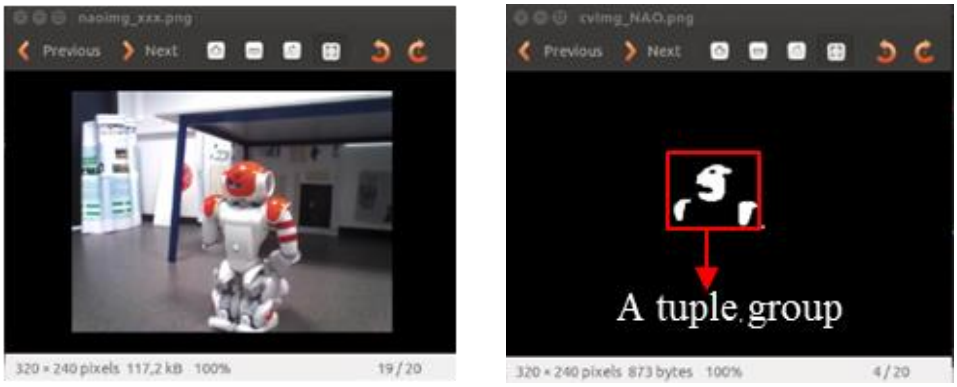
Fig. 6-3 The examples of the same NAO's visual features captured with different relative orientations θ. (a) $\theta = 0^{\circ}, \varphi = 0^{\circ}$; (b) $\theta = 90^{\circ}, \varphi = 0^{\circ}$; (c) $\theta = 120^{\circ}, \varphi = 0^{\circ}$;(d) $\theta = 180^{\circ}, \varphi = 0^{\circ}$; (e) $\theta = 180^{\circ}, \varphi = 90^{\circ}$; (f) $\theta = 320^{\circ}, \varphi = 0^{\circ}$.($\theta$ is the orientation, the angle between the torso and the north direction, while φ is the heading, the angle of head yaw.)

1. Generation of a mask

As shown in Fig. 6-3, the orange regions of a NAO can be detected and distinguished from the surroundings by being segmented from the perceived image in the HSV color space with the orange range $(2, 170, 50)$ ((Hmin, Smin, Vmin)) and $(10, 255, 255)$ ((Hmax, Smax, Vmax)): except that the orange regions are transformed into the white color, the other colors are transformed into the black color, shown in Fig. 6-4. Then the head and shoulders (or the torso) constitute one mask.



(a) (b)



(c) (d)

Fig. 6-4 Image segmentation for recognizing a NAO: detection of the orange tuples

(a), (c) The origin color images. (b), (d) The corresponding segmented images.

2. Construction of the M-base

Ideally, this M-base should consist of all the features of a NAO got from different distances d, different headings φ and different orientations θ. To save time, the masks in this M-base are got from the fixed distance $d = 0.6m$, with the head yaw keeping to zero degree: $\varphi = 0$, as the effective visual perception range is $[0.5m, 1.45m]$. To get an accurate M-base, the masks are generated in a simulated software "V-rep", shown in Fig. 6-5, according to the following criteria:

(1) Constructing two NAOs (NAO-*i* and NAO-*j*) totally according to the parameters of the real NAOs [134], both NAOs maintain the same posture: "StandZero".

(2) NAO-*i*, NAO-*j* both stand on the y axis of the "V-rep" frame with the position (0,0) and (0,0.6*m*) respectively.

(3) The orientation of NAO-*i* is fixed, always along the y axis of the "V-rep" frame ($\theta_i = 0$), shown in Fig. 6-5, while NAO-*j* turns around continuously with a step length: $\theta_{stp} = 5^\circ$ (in Fig. 6-5, the orientation of NAO-*j* is $\theta_j = 180^\circ$).

(4) At each orientation of NAO-*j*, NAO-*i* captures one image of NAO-*j*, and generates one mask with the features located in the center part of the image. After NAO-*j* turns around, 72 ($360^\circ / 5^\circ$) masks are generated, which makes up the M-base, some examples are shown in Fig. 6-6.

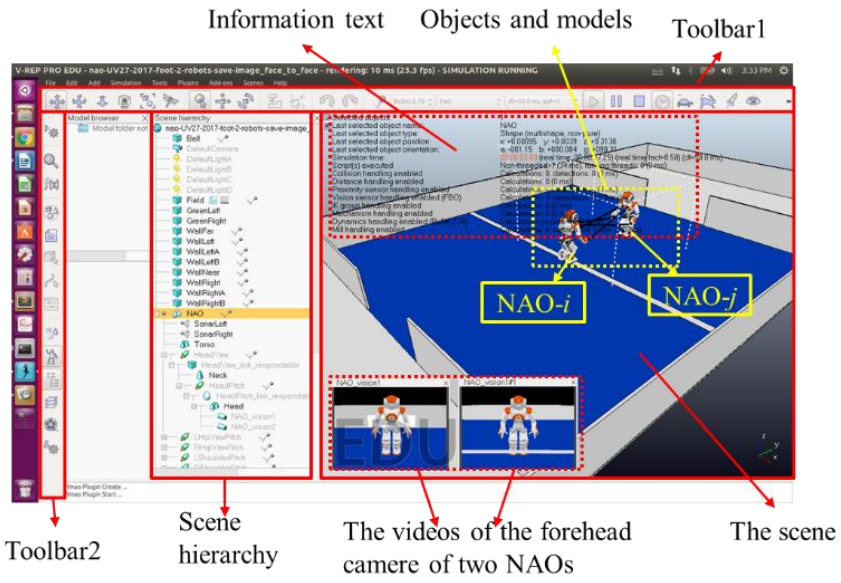


Fig. 6-5 The generation of the M-base in V-rep

(a) (b) (c) (d) (e)

Fig. 6-6 Some masks of the M-base corresponding to the (a)-(d), (f) of Fig. 6-3.

(a) $\theta = 0^{\circ}$; (b) $\theta = 90^{\circ}$; (c) $\theta = 120^{\circ}$; (d) $\theta = 180^{\circ}$; (e) $\theta = 320^{\circ}$.

3 Recognition steps

To recognize the pose and estimate the distance to another NAO from one image, the 2D discrete cross-correlation (equation (6-1)) is used to select the mask in M-base that is the most similar to the observed tuple.

$$(f * g)[x, y] = \sum_{m=0}^{w} \sum_{n=0}^{h} f[m, n] g[x + m, y + n] \qquad (6\text{-}1)$$

Here, $f[m, n]$ stands for one mask in the M-base, while $g[x, y]$ represents a tuple group extracted from one image. w and h are the width and height of the tuple group $g[x, y]$ respectively.

The larger the value of $(f * g)$ is, the higher the similarity between $f[m, n]$ and $g[x, y]$ will be. Then we define a confidence factor $\lambda = \sqrt{\frac{a}{b}}$ (the possibility to be a NAO) to judge whether $g[x, y]$ is a NAO or not. Here, $a = (f * g)[x, y]$ and $b = (f * f)[x, y]$.

However, usually, the segmented images include more than just the NAOs' features because of other orange objects or light noises in each image, as shown in Fig. 6-7(b). To reduce the noises,

the qualities of the images should be improved. In this work, the noises are filtered by one Gaussion filter combined with some morphological operations (closing and opening), and one improved image is shown in Fig.6-7(c). Like the Fig.6-7(c), there are usually more than three tuples in the images because more than one NAOs sometimes are shown in the pictures and light noises sometimes cannot be reduced sufficiently for reserving the NAOs' features (the orange regions) as much as possible. In this case, to find the real tuple group, every three tuples (or two tuples if in the "side" postures) will construct one group $g_j[x, y]$, all the tuple groups in one image will assemble to a set $S = \{g_j[x, y] \mid j \in 1, 2, ...C_n^3 (or \quad C_n^2)\}$, n is the number of tuples detected from one image.

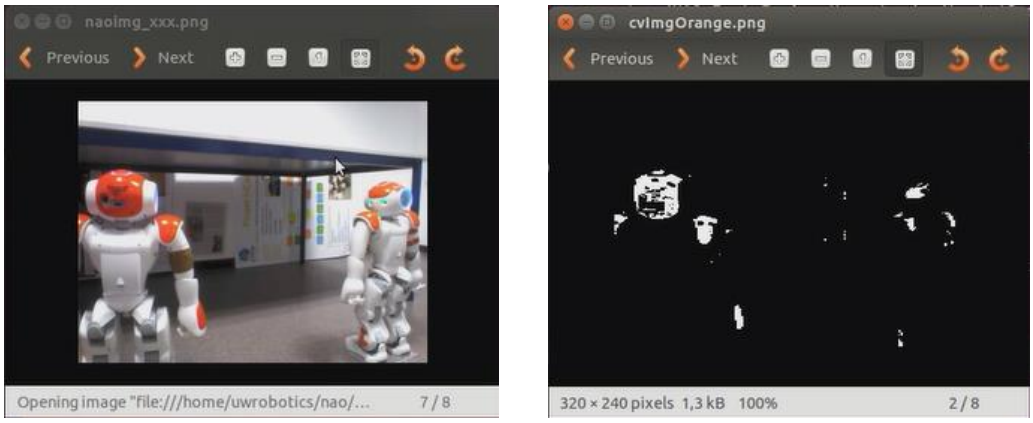


Fig. 6-7 The tuples extracted from one image. (a) One image got from the forehead camera. (b) The segmented image in the HSV space. (c) The segmented image after the image enhancement of (b) with one Gaussian filter and the closing and opening operations of morphological operations.

Algorithm 2 determines which mask in the M-base corresponds to the visual features extracted from the current image. It carries out an exhaustive search on every tuple group in the image, every pose and every distance. Each distance corresponds to a scale factor r_g on the masks computed at a fix distance of $d = 0.6m$. The association of the mask in M-base and the tuple group $g_j[x, y]$

in the image showing the highest confidence $\lambda_{\max}$ is selected.

If $\lambda_{\max}$ is larger than a given threshold ($\varepsilon_{\text{NAO}} = 0.6$ in our study), then we decide that the tuple group can be associated with a NAO. Its position in the image is that of the tuple group $g_j[x, y]$, meanwhile, the associated scale r_{g_j} will also be stored.

Remarks: As all the masks in the M-base are generated in the image center part, as a result, all the tuple groups are valid only when they are located in the center part of an image.

Algorithm 2 Recognition of NAOs with all the tuple groups in one image

for $g_j[x, y]$ in tuple groups: **do**
  for mask in M-base: **do**
    for scale (r_{g_j}) in range [0.01,1.5] with a step 0.01: **do**
      dim = size of mask × r_{g_j}
      if the size of dim is smaller than the size of the tuple group $g_j[x, y]$: **then**
        break
      end if
    end for
    mask_new = mask × dim, to change the size of mask to become smaller than the size of $g_j[x, y]$, λ_{g_j} is calculated by comparing the mask_new with the tuple group $g_j[x, y]$, and is stored in the list L_λ
  end for
  $\lambda_{g_j} = \max\{L_\lambda\}$ is the evaluation of $g_j[x, y]$ and is stored in the list $L_{\lambda_{\max}}$
end for
$\lambda_{\max} = \max\{L_{\lambda\max}\}$ is the evaluation of a NAO existence or not in this image

6.2.2 Extraction of neighboring information

When the neighboring NAOs are recognized from the images, the neighboring information (relative positions or relative distance vectors) should be calculated, which includes the relative distances and angles (and the orientation of NAO itself). As there is only one on-board camera, to obtain the relative distances and angles, a distance model and an angle model should be put forward and calibrated. To get the orientation of NAO, a visual compass is introduced to make up for the lack of a real one.

1. Identification of the orientation

The visual compass uses the forehead camera to get the offset of orientation ($\Delta\theta$) at each rotation step by combining the comparative results of two adjacent images and the values measured by accelerometer and gyroscope [135]. To guarantee the validity of visual compass, there should be enough corresponding features in the two adjacent images. It means that the surroundings should include abundant features, and each rotation step cannot be too large. Fig. 6-8 shows one comparison

result of two adjacent images. Then the current orientation will become:

$$\theta_c = \theta_l + \Delta\theta \tag{6-2}$$

Here, θ_c and θ_l are the current and latest orientations of the rotating NAO relative to the initial orientation. It implies that the orientation is a relative angle, whose value depends on the initial orientation given.

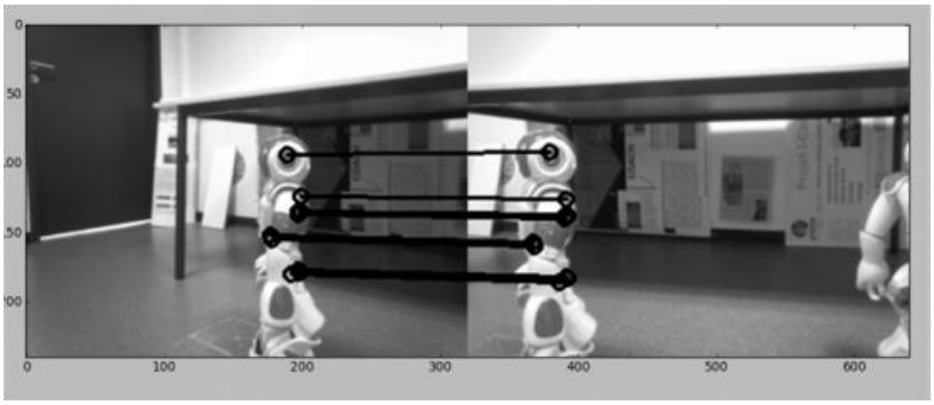
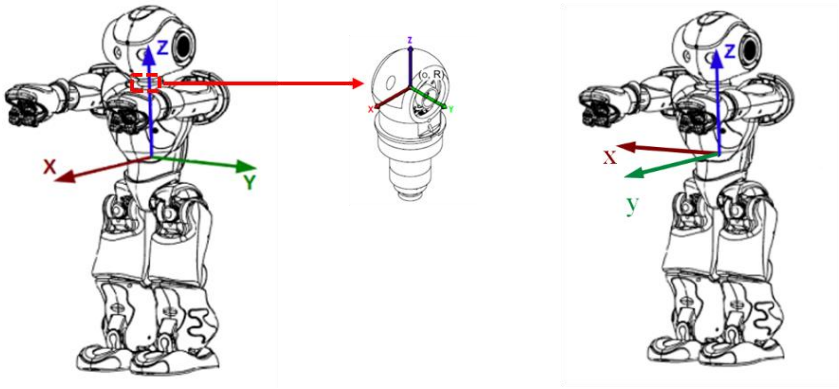


Fig. 6-8 The corresponding feature points in two adjacent images, left image stands for the current image, while the right one is the latest image.

2. Establishment of the angle model

Before establishing the angle model, the NAO frame (noted as NAO-*i*) and the neck frame (noted as neck-*i*) should be described firstly. Shown in Fig. 6-9(a), in the NAO-*i* (the left one), the x axis is looking forward in the horizon plane, y axis is perpendicular to the x axis in the horizon plane, the origin point is the mass center, while z axis is perpendicular to the xy plane. And the NAO-*i* is relatively fixed to its torso, which is changeable following the movement of the NAO. The axes of the neck-*i* are totally parallel to those of the NAO-*i*, the head yaw φ_{Nij} in the NAO-*i* will be equal to φ_{nij} (the angle of head yaw between the neck joint and x axis of neck-*i*) in the neck-*i*.

Associated with each NAO-*i*, a local frame is constructed, shown in Fig. 6-9(b), noted as the local-*i*. Its y axis and origin point coincide with the x axis and origin point of NAO-*i*, its x axis is along the inverse direction of the y axis of the NAO-*i*. Like the NAO-*i*, the local-*i* is also relatively fixed with the NAO's torso.



(a) (b)

Fig. 6-9 The NAO frame (NAO-*i*), neck frame (Neck-*i*): (a), and the local frame (local-*i*): (b). In the NAO-*i*, the x axis is looking forward in the horizon plane, y axis is perpendicular to the x axis in the horizon plane, and the origin point is the mass center, while z axis is perpendicular to the xy plane. The Neck-*i* is parallel to the NAO frame. In the local-*i*, the origin point and its y axis coincide with the origin point and x axis of the NAO-*i* respectively, its x axis is along the inverse direction of the y axis of the NAO-*i*.

The angle model expresses the relative angle between one neighbor NAO-*j*, and NAO-*i* in the local-*i*, φ_{lij}. The heading φ_{nij} can be read from the neck joint. Considering that only the tuple groups at the center part of one image will be used to recognize the NAO in our study, the heading of NAO-*i*, $\varphi_{Nij}=\varphi_{nij}$, can be seen as the relative angle between one neighbor NAO-*j* and NAO-*i* in the NAO-*i* according to the method in section 6.1.1. Then the angle model can be expressed as:

$$\varphi_{lij} = \varphi_{Nij} + 90^{\circ} \qquad (6\text{-}3)$$

3. Establishment of the distance model

As mentioned in section 6.1.1, each extracted NAO is associated with a scale r_{g_j} to get the maximum similarity value. Fig. 6-10 shows the camera model, in which, $O_i u_i v_i$ is the image frame, *u* axis and *v* axis are along the width and height of the image respectively, while $O_{ci} x_{ci} y_{ci} z_{ci}$ is the camera frame of forehead camera of the NAO-*i*, O_{ci} is the optic center, its z axis, $O_{ci} z_{ci}$, is along the direction of optic axis and is approximately parallel to the x axis of the NAO-*i*, x axis $O_{ci} x_{ci}$ and y axis $O_{ci} y_{ci}$ are parallel to the *u* axis and *v* axis respectively. Assuming that A_{ij} and A_{uv} are the projection's areas of NAO-*j* in the $O_{ci} x_{ci} y_{ci}$ (translated to the position of NAO-*j*, p_{ij}, the blue area shown in Fig. 6-10) and in the $O_i u_i v_i$, we can get an ideal relationship between the

distance d_{ij} and the area A_{uv} of NAO-*j* in the image, shown in equation (6-4). Equation (6-5) describes the relationship between d_{ij} and d_f ($d_f = 0.6m$ is the distance of generating the M-base).

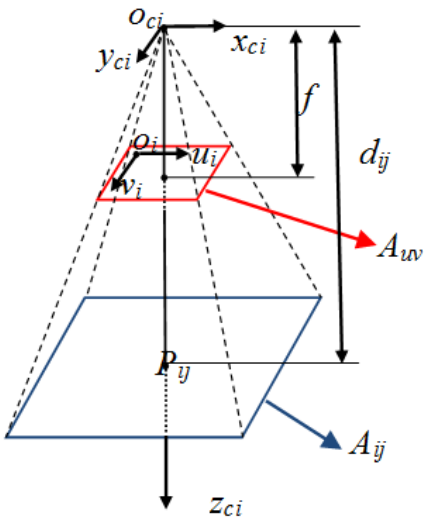


Fig. 6-10 Camera model. $o_i u_i v_i$ is the image frame, *u* axis and *v* axis are along the width and height of the image respectively, while the $o_{ci} x_{ci} y_{ci} z_{ci}$ is the camera frame of forehead camera of the NAO-*i*, o_{ci} is the optic center, its z axis is along the direction of optic axis and the z axis is parallel to the x axis of the NAO-*i*, x axis and y axis are parallel to the *u* axis and *v* axis respectively. p_{ij} is the mass center of NAO-*j* in the NAO-*i*, *f* is the focus of forehead camera of NAO-*i*. A_{ij} and A_{uv} are the projections of NAO-*j* in the plane $o_{ci} x_{ci} y_{ci}$ and in the CCD $o_i u_i v_i$. The distance between NAO-*i* and NAO-*j* is d_{ij}.

$$\begin{cases} \dfrac{d_{ij}^2}{(f+\delta_{ij})^2} = \dfrac{A_{uv}}{A_{ij}} = \dfrac{r_{ij}^2 wh}{wh} \\ \dfrac{d_f^2}{(f+\delta_f)^2} = \dfrac{A_{uv2}}{A_{ij}} = \dfrac{r_f^2 wh}{wh} \end{cases} \tag{6-4}$$

$$\frac{d_{ij}}{d_f} = \frac{f+\delta_f}{f+\delta_{ij}} \cdot \frac{r_{ij}}{r_f} = \frac{f+\delta_f}{f+\delta_{ij}} \cdot \frac{1}{r_{g_j}} \tag{6-5}$$

Where, δ_{ij} and δ_f are the uncertain factors because $g_j[x, y]$ cannot exactly be at the center of one image. r_{ij} and r_f are the scales between the size of A_{uv}, A_{uv2} (the projection area in the image when $d_f = 0.6m$) and A_{ij} respectively. And as $d_f = 0.6m$ is the distance for the generation

of M-base. Then the distance d_{ij} is deduced to have an inverse proportion relationship with the scale r_{g_j} of NAO-*j*, expressed by equation (6-6) got from equation (6-5).

$$d_{ij} = \frac{a'}{r_g} + b' \qquad (6\text{-}6)$$

Where, a' is fluctuated around 0.6, standing for the distance to generate the M-base, while b' represents uncertain factors, a' and b' compose a coefficient pair. Since the NAO-*j* is recognized according to the different masks of the M-base, a' and b' corresponding to each mask are not the same. As a result, a distance base (D-base) is built to store these coefficient pairs, shown in table 6-2, while the building process of D-base is presented in the calibration of the distance model.

4. Calibration of the distance model

To get the accurate calibration results, we calibrate the distance model in the simulation software "V-rep" with the two built NAOs, shown in Fig. 6-5. The steps to sample the points is given:

(1) NAO-*i* and NAO-*j* both stand on the y axis of the "V-rep" frame, NAO-*j* has a fixed position $(0,1.0m)$, while NAO-*i* can change its position in the interval $[(0,0.5m),(0,-0.45m)]$ with a back step length $0.05m$ along y axis to keep the distance in the range $[(0,0.5m),(0,-0.45m)]$, 20 positions to be considered.

(2) NAO-*j* is set to every orientation associated with the M-base, whereas, NAO-*i* keeps along the y axis of the "V-rep" frame.

(3) For every orientation of NAO-*j*, NAO-*i* recognizes NAO-*j* and gets the scale r_{g_j} at each position mentioned in (1). Every distance d_{ij} and the reciprocal of the scale $\frac{1}{r_{g_j}}$ form a sample pair. Then 20 sample pairs are obtained, called a sample set.

(4) After the NAO-*j* turns around, 72 sample sets associated to 72 masks in the M-base will be generated. The sample pairs in each set are given into Matlab, a curve is fitted with the tool "cftool" (Curve Fitting Toolbox), and one coefficient pair (a', b') of equation (6-6) can be got. Finally, 72 coefficient pairs are got and they make up a D-base.

Fig. 6-11 gives an example of fitted curve with $\theta = 180^\circ$. Table 6-2 presents part of coefficient pairs (a', b') calibrated in the D-base.

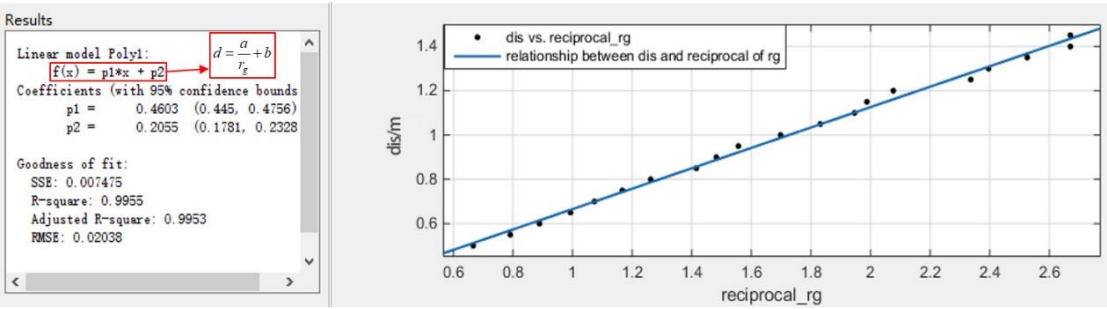


Fig. 6-11 An example of fitting for the distance model with $\theta = 180^{\circ}$. Here $f(x), x, p_1, p_2$ represent d, $\frac{1}{r_g}$, a', b' respectively.

Table 6-2 Part coefficient pairs of D-base (a', b') associated to different orientations

a'	b'	θ
0.5044	0.1095	0
0.5167	0.09141	5
0.5146	0.09774	10
0.5068	0.1037	15
0.5032	0.1053	20
0.4908	0.1258	25
…	…	…

6.3 The performance test of NAO

Considering that the valid distance range to the mutual recognition between NAOs is $[0.5m, 1.45m]$, and the horizon view angle of forehead camera is 60.9°, the expected relative distance and angle between each NAO and its follower in the pyramid pattern are set as: $d_s = 0.8m$, $\varphi_s = 30^{\circ}$. Differently to the simulations, some unexpected problems occur due to the environmental noises, such as the disturbance of objects with the same or similar colors as NAO, the impact of the light refection on the features, and the joints and sensors' errors caused by the robot aging. To solve these problems, a series of tests are lead and associated solutions are given.

6.3.1 Recognition of NAOs at "side" postures

Differently from the other postures, every tuple group in the "side" postures is composed by two tuples. Consequently, we should give a way to distinguish the "side" postures and recognize NAOs at "side" postures, since the "side" postures generate fewer features than the other postures.

Shown in Fig. 6-6(b), the widths of the tuple groups in "side" postures are smaller than those

in other postures. As a result, the ratio $r_w = \frac{w_f}{w_i}$ is used to judge the types of the postures, in which, w_f and w_i are the widths of the visual features and the image respectively. In our experiments, if $r_w < 0.225$, we believe the NAO stands sideways.

Since only the head and one shoulder are detected in the "side" postures, only two tuples are extracted, then sometimes it fails to recognize the NAOs. The reason of failure is that the M-base is generated with $\varphi = 0^\circ$ in an ideal simulation environment without any noises. However, in the reality, there are great differences between the features with $\varphi = 0^\circ$ and those with $\varphi = 90^\circ$ in the "side" postures. Then we provide a supplementary "side" M-base got in the real environment with the light noise, which includes 2 masks for the same posture but with two different kinds of light noises. The posture parameters are: the heading $\varphi = 90^\circ$, the distance $d_f = 0.6m$, and the orientation $\theta = 270^\circ$, shown in Fig. 6-12.

(a) (b)

Fig. 6-12 Two added masks with "side" postures got in the real environment with two different kinds of light noises. The posture parameters are: the heading $\varphi = 90^\circ$, the distance $d_f = 0.6m$, and the orientation $\theta = 270^\circ$.

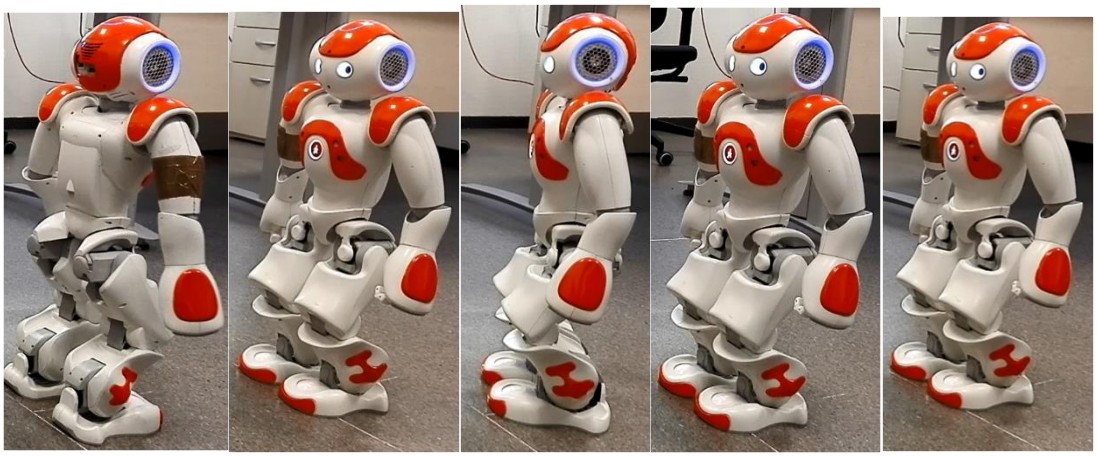
6.3.2 Evaluation of the visual compass

To make the rotation of NAOs accurate, PD control method (shown in equation (6-7)) is adopted to control the rotation of NAOs, in which, the rotation angle estimated by visual compass is regarded as a feedback to correct the rotated angle. Given that the relative angle estimated by visual compass is $\Delta\hat{\theta}_i[j]$, the whole rotation process is separated into several steps with an initial step length $\theta_{step} = 15^\circ$.

$$\begin{cases} \Delta\theta_i[j] = K_{\mathrm{P}}\hat{\theta}_{step}[j] + K_{\mathrm{D}}(\hat{\theta}_{step}[j] - \hat{\theta}_{step}[j-1]) \\ \hat{\theta}_i[j] = \hat{\theta}_i[j-1] + \Delta\hat{\theta}_i[j] \\ \hat{\theta}_i[0] = 0 \end{cases} \quad (6\text{-}7)$$

Where, $K_{\mathrm{P}} = 1.0$, $K_{\mathrm{D}} = 0.2$ are the proportion and derivated factors respectively, $\hat{\theta}_{step}[j] = \min\{\theta_{step}, \eta_i - \hat{\theta}_i[j-1]\}$, $\hat{\theta}_i[j]$ stands for the rotated angle from beginning to the j_{th} step, η_i is the expected rotation angle.

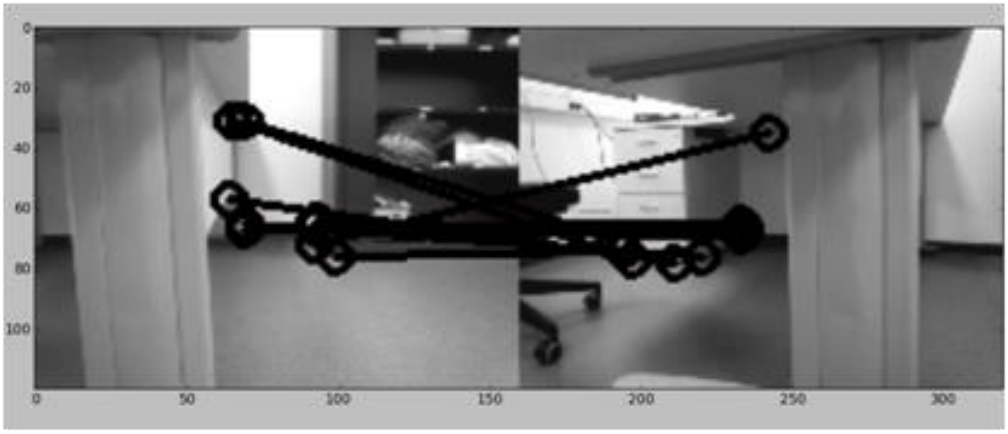
To evaluate the precision of the visual compass, the behavior of turning around 180° is tested repeatedly by NAO itself or by the visual compass providing a feedback at a given position (a line written by a chalk on the ground), shown in Fig. 6-13(a). The results are also shown in Fig. 6-13, in which, Fig. 6-13(b) and (c) show the good and bad rotation results by NAO itself, Fig. 6-13(d) and (e) give the rotation results by using the visual compass to provide the feedback. We find that: (1) the bad rotation results are mainly caused by the heavy sways of NAOs due to NAOs aging; (2) the results obtained with the feedback of visual compass are more accurate than those obtained by itself, especially the sways of NAOs have little influence on the rotation. But the visual compass also has some shortcomings, illustrated in Fig. 6-14: (1) when there are few features in one image, the orientation offset will be wrong because of the lack of corresponding features, shown in Fig. 6-14(a); (2) when there are similar features in the adjacent images, such as the corners of a table, wrong results will occur, like Fig. 6-14(b); (3) when one step length is too large, similarly to (1), no features will be matched up, which leads to get one false result, such as Fig. 6-14(c) with one step length of 40°.



(a) (b) (c) (d) (e)

Fig. 6-13 Rotation results obtained without or with a visual compass. (a) The initial position. (b), (c) good and bad rotation results without the visual compass. (d) and (e) the rotation results when the visual compass is introduced.

(a)



(b)

(c)

Fig. 6-14 Three examples of the situations leading to wrong estimations of orientations by the visual compass. (a) Few features in one or two images. (b) Similar but not the same features in two images. (c) Large rotation step length (40°).

6.3.3 The movement control of NAO

Since the planned trajectories are straight lines or the fold-lines, the linear movement ability of NAOs should be tested and improved. To do so, NAOs repeatedly move on different distances along a given line (a chalk line on the ground, rewritten by a red dashed line, shown in Fig. 6-15(a)).

The results are presented in table 6-3 (some results show big errors caused by the heavy sway due to the aged joints).

Table 6-3 The moving distances and associated orientation errors

dis/*m*	e_{orien}/*radian*
0.2	-0.0211
0.2	0.0957
0.3	0.3734
0.3	-0.0379
0.4	0.0125
0.4	-0.1279
0.5	0.1374
0.5	0.1698
0.7	0.5292
0.7	0.2274
0.8	0.334
0.8	0.4465
1	0.4493
1	0.4062

From table 6-3, we can observe that the movement trajectory can be seen as a straight line when the step length of moving forward is no more than 0.2m.

We assume the feedback of the inertial unit is accurate, the movement distance is considered as accurate. Then in the distance control, the step length of moving forward is set as $d_{step} = 20cm$. To make sure the movement trajectory is approximate to a straight line, rotation control is introduced to correct the orientation of NAOs in each step (but $K_{\mathrm{D}} = 0$), shown in equation (6-8).

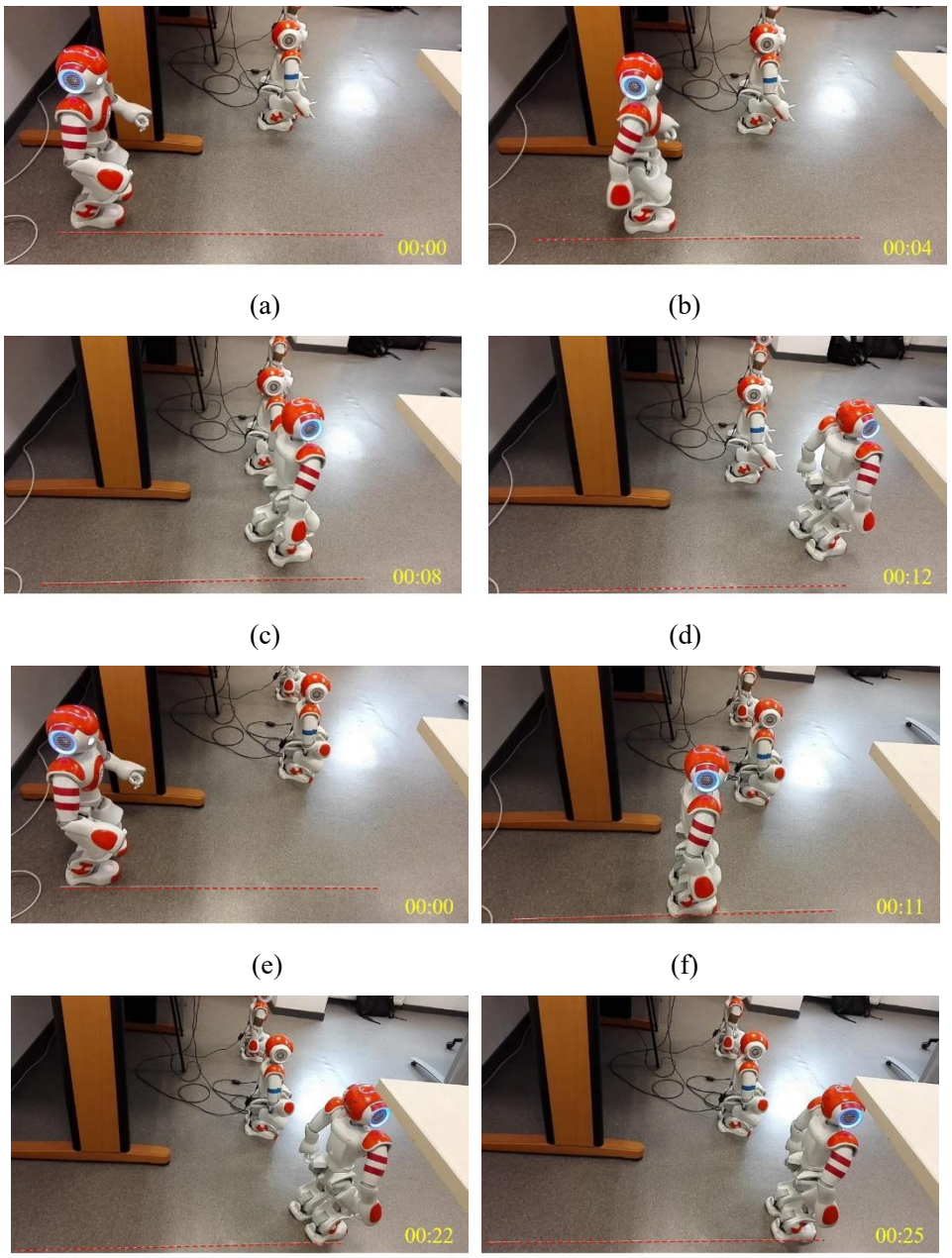
$$\begin{cases} \Delta d_i[j] = \min\{d_{step}, d_i - d_i[j-1]\} \\ d_i[j] = d_i[j-1] + \Delta d_i[j] \\ d_i[0] = 0 \\ \Delta\theta_i[j] = K_{\mathrm{P}}\hat{\theta}_{step}[j] \end{cases} \tag{6-8}$$

Where, $d_i[j]$ is the movement distance from the starting position to the j_{th} step.

To test the capabilities of the distance control, NAOs repeatedly run 0.8*m* along a line with two kinds of movement ways respectively: 1) one by itself without the distance control, whose results are shown in Fig. 6-15(a) - (d); 2) the other one with the distance control, whose results are shown in Fig. 6-15(e) - (h). From Fig. 6-15, we can find that when the step length is $d_{step} = 20cm$, the NAOs are able to move straight well (see Fig. 6-15 (b), (f)).

Occasionally, it will generate a big orientation deviation, shown in Fig. 6-15(g). In this case, the rotation control is used to correct its orientation (Fig. 6-15(h)). But when NAO move a long distance by itself without the distance control, the trajectories are curves, shown in Fig. 6-15(c) and

(d).



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 6-15 The results of NAOs moving 0.8m along a line (A chalk line, rewritten with red dashed line). (a)-(d) The movement results by itself. (e)-(h) The movement results by the distance control. (a) and (e) are the initial postures of both tests. (b) is the posture of moving forward 20cm by itself. (c) and (d) shows the deviation of the movement by NAO itself. (f) is the posture of moving forward 40cm after the second step with $d_{step} = 20cm$. (g) presents the deviation of the orientation even moving forward 20cm. (h) is the correction result of (g) by using the orientation feedback of distance control at the end of this step.

6.4 Pyramid building with NAOs

6.4.1 Pyramid building with the local position-based method

After the estimation and improvement of the properties of NAOs, the pyramid building is done with the local position-based method according to the steps below:

(1) The origin orientations of all the NAOs are set identically manually to keep all the fixed frames being parallel, because of the lack of compass.

(2) Each NAO searches its neighbors around itself independently and gets neighbors' information (i.e. relative positions) with the proposed integrated image processing method described in section 6.1.

(3) All the NAOs communicate with each other by WiFi to exchange their neighbors' information in a center PC, after all the NAOs finish the step (2).

(4) The NAOs distinguish the IDs of their neighbors by themselves according to the neighbor-check method given in section 3.2.3.

(5) The NAO with the largest number of neighbors is chosen as the center, its fixed frame is considered as the common frame. Then the positions of the other NAOs can be translated into this common frame, and the initial distribution is then known.

(6) The proposed collision avoidance strategy is used to design an optimized pyramid pattern being closed to the initial distribution and to match up the positions in both the initial and the pyramid distributions in the pyramid frame to generate the non-intercrossing straight trajectories (η_i, d_i).

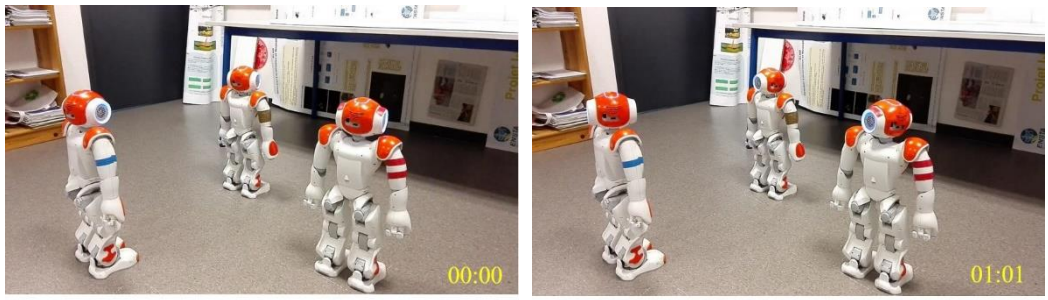
(7) Finally, with the kinematic control (distance control and rotation control), the leader will go to its destination firstly, then the followers will move along their trajectories to arrive at their destinations. After turning back to the pyramid orientation, the pyramid building is completed.

Remarks: Because the NAOs share the same algorithms, (4)-(7) steps are done inside each NAO independently.

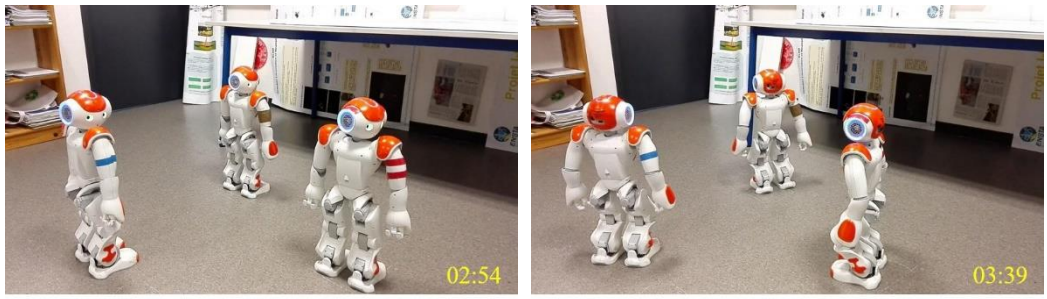
When searching the neighbors, the forehead camera of each NAO needs to scan the surroundings a round. As the range of head yaw is $\varphi \in [-119.5^\circ, 119.5^\circ]$ in the NAO frame, the NAOs can get all their neighbors' information after scanning twice with the forehead camera at the orientations of $\theta = 0^\circ$ and $\theta = 180^\circ$ respectively in the fixed frames. The rotation to $\theta = 180^\circ$ from $\theta = 0^\circ$ is optimized by the rotation control.

After having shared neighbors' information by WiFi, the NAOs will plan trajectories (η_i, d_i) with the proposed local position-based method. After that, the NAOs move to their destinations along their trajectories. During this process, rotation control is used to rotate to face the destination, and the distance control is then adopted to move straight approximately until they arrive at the

destination. One example is shown in Fig. 6-16.

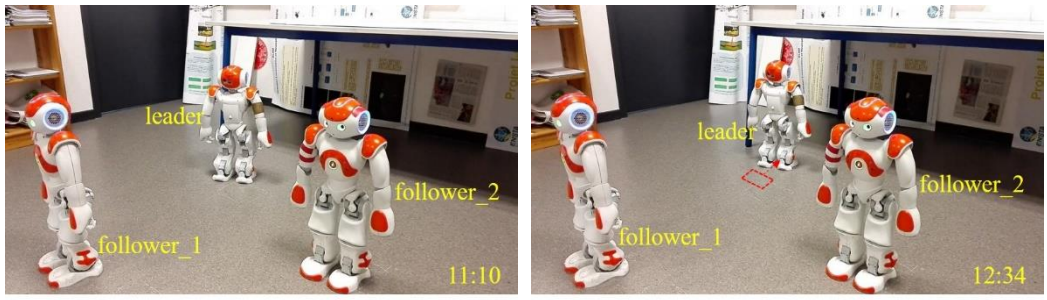


(a) (b)

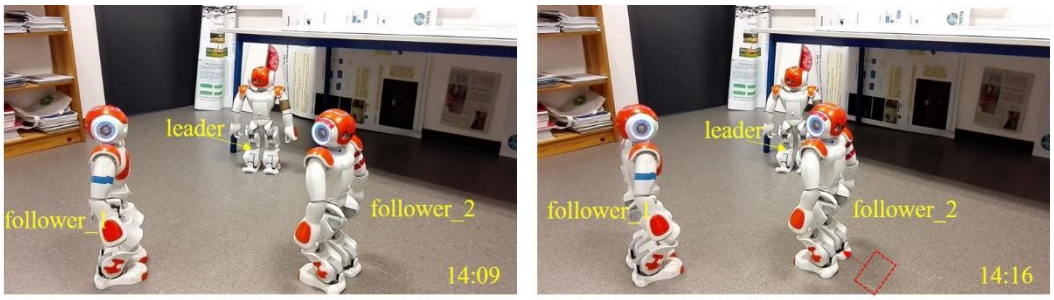


(c) (d)

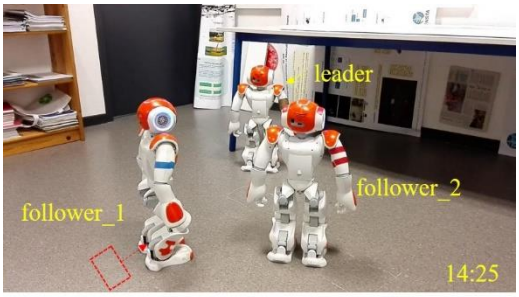
(e) (f)



(g) (h)



(i) (j)



(k)

(l)

Figure 6-16 An example of pyramid building. (a) Initial distribution of NAOs. (b) The NAOs search neighbors at $\theta = 0^{\circ}$ with the $\varphi \in \left[-119^{\circ}, 0^{\circ}\right]$. (c) The NAOs search neighbors at $\theta = 0^{\circ}$ with the $\varphi \in \left[0^{\circ}, 119^{\circ}\right]$. (d) The NAOs are rotating to $\theta = 180^{\circ}$ from $\theta = 0^{\circ}$ with the rotation control. (e) The NAOs search neighbors at $\theta = 180^{\circ}$. (f) NAOs are exchanging information among NAOs by WiFi through the PC, then the leader and followers are identified, the trajectories are planned well with the proposed method. (g) The leader is rotating to the destination with the rotation control. (h) The leader has arrived at its destination with the distance control. (i) The leader rotates to the pyramid orientation and the followers begin rotating to face to their destinations with the rotation control. (j) Follower_2 arrives at its destination, while follower_1 still rotates to its destination. (k) Follower_2 has rotated to the pyramid orientation and follower_1 has arrived at its destination. (l) After follower_1 rotates to the pyramid orientation, the pyramid building is achieved.

6.4.2 Pyramid building with the asynchronous limited communication distributed control method

Because of the lack of compass, it is hard to update the global orientation of NAO in real-time. Then in this section, the pyramids are built only with the asynchronous discrete consensus algorithm associated with the distance-based control method. As the trajectories should be updated continuously using this method, time consumption is high (see section 5.3.4). To save time, the convergent condition of pyramid building is defined by a relative big allowed error. Since the distance-based control method has weak constraints on the orientation, the orientations of NAOs have big errors at the preliminary built pyramid, sometimes the angle errors are even unacceptable. Then a fine-tuning step is necessary to get a satisfying pyramid pattern.

A. Pyramid building

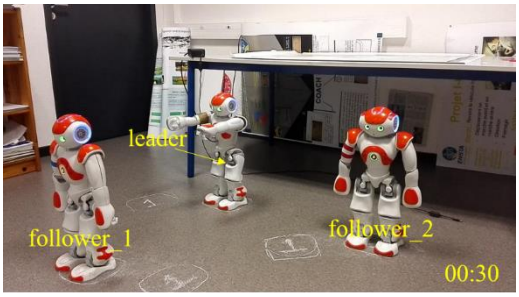
The process of the pattern building with the distance-based control method is presented below:

(1) Similar to the local position-based method, each NAO searches its surroundings to find at least two neighbors and estimate their relative distance vectors (if not, the NAOs will repeat the loop “search neighbors”).

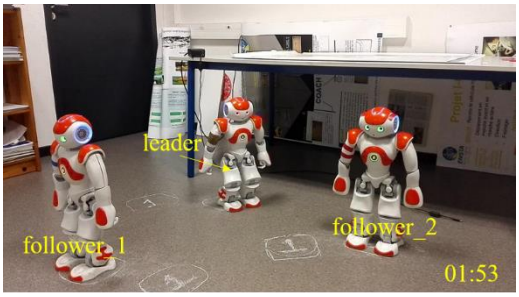
(2) All the NAOs share their information via the central PC by WiFi, then a specific pyramid pattern will be designed. And each NAO will get the expected position relationship, leader and followers (see Fig. 6-17(a)).

(3) The position of leader is seen as the position in pyramid pattern and it remains unchanged. Then the leader turns on the spot to faces to the followers. Since all the NAOs are homogeneous, to simplify the recognition process, the eyes of leaders are set to be red, while the eyes of followers are set to be green, shown in Fig. 6-17(b).

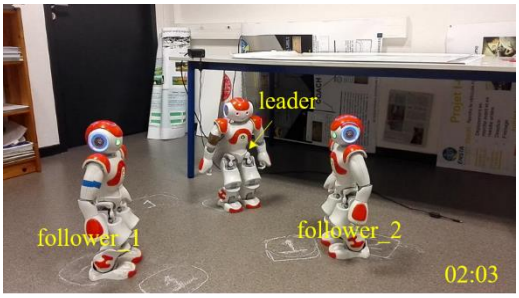
(4) The followers plan and update their trajectories according to equations (5-17) and (5-18), until they reach their destinations and complete the preliminary pyramid building, shown in Fig. 6-17 (c) - (h).



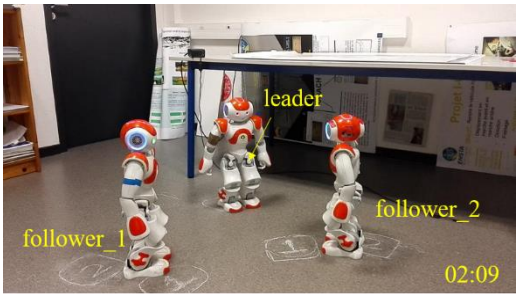
(a)



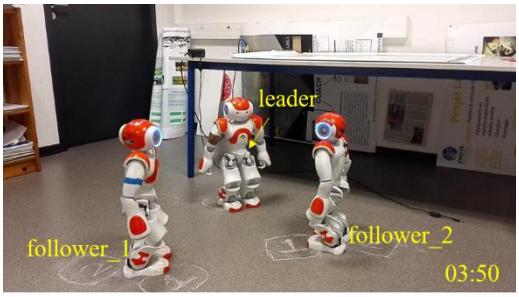
(b)



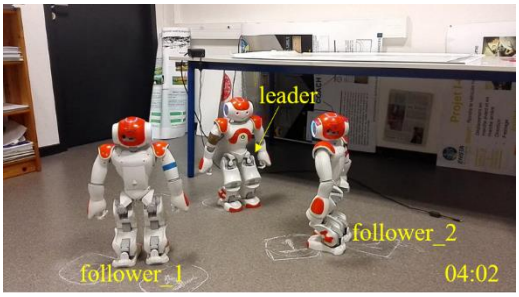
(c)



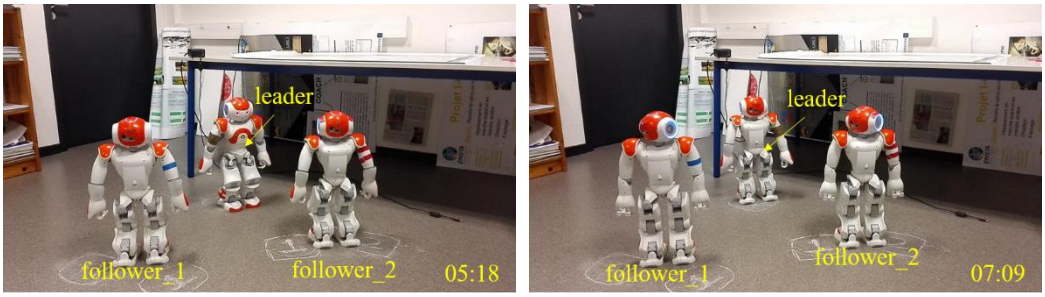
(d)



(e)



(f)

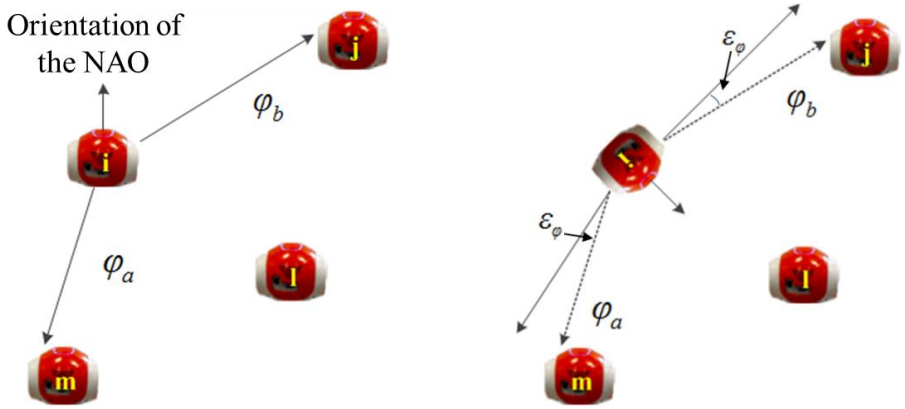


(g) (h)

Fig. 6-17 The process of building a preliminary pyramid pattern. (a) After searching neighbors around themselves and exchanging information by WiFi, each NAO gets the position relationship with the other NAOs and knows its role. (b) The leader rotates to face the followers with red eyes, while the followers' eyes are set to be green for the distinguishment. (c) Both followers move to build the pattern by running the loop ("search, compute, move") independently and follower_2 finishes its first loop. (d) Follower_1 finishes its first loop. (e) Follower_2 finishes its second loop. (f) Follower_1 finishes its second loop and arrives at its destination. (g) Follower_2 finishes its third loop and arrives at its final position. (h) Leader turns back and fine-tune its position and orientation.

Different from the local position-based method, to update their trajectories, the NAOs should turn around and scan their surroundings to find new neighbors and get the current distance vectors independently in every loop with the asynchronous limited communication distributed control method. But the duration of turning around by itself is about 5 minutes. The worse thing is that the battery of the NAO can only maintain about 30-40 minutes so that the pyramid building sometimes cannot be accomplished due to the lack of electricity.

Actually, after the first scanning, the NAO can know the distribution area of its neighbors in the NAO-*i* at this moment shown in Fig. 6-18 (a). Hence, we give a larger range, $\omega_i[(k+1)_i]=[\varphi_a-\varepsilon_\varphi,\varphi_b+\varepsilon_\varphi]$, to cover the area of its neighbors in the next loop. (ε_φ is introduced to decrease the effect of errors resulted from the visual compass, from the image processing, and from the movement of its neighbors, $\varepsilon_\varphi=20^\circ$ in our experiments). In its next loop, the NAO will rotate to face to this area and only scan this area $\omega_i[(k+1)_i]$, shown in Fig. 6-18(b).



(a) (b)

Fig. 6-18 Illustration graph of the scanning area of NAO-*i*. (a) The neighbors' area got from the camera-*i* in the k_i loop $[\varphi_a, \varphi_b]$. (b) The scanning area evaluated to cover the real neighbors' area in the $(k+1)_i$ loop $[\varphi_a - \varepsilon_\varphi, \varphi_b + \varepsilon_\varphi]$.

B. Fine-tuning of the pyramid pattern

Since the consensus algorithm based on the distance vectors has no constraints on the angle, the final relative angle errors sometimes are unacceptable. Therefore, a fine-tuning of the pyramid pattern is necessary, where the allowed errors are set as: $e_d = 0.1m$, $e_\varphi = 10^\circ$, which are the values between one robot and the parent robot.

We assume that the distance error is along the line connecting each NAO (NAO-*i*) and the robot it following (NAO-*h*) but with a big relative angle error $e_{\varphi_{ih}} = \varphi_{ih} - \varphi_s$, shown in Fig. 6-19(a). With the same system of equation (5-16), the control input becomes:

$$\mathbf{u}_i[(k+1)_i] = \mathbf{e}_{ih}[k_i] \qquad (6\text{-}9)$$

Then the fine-tuning steps are given below:

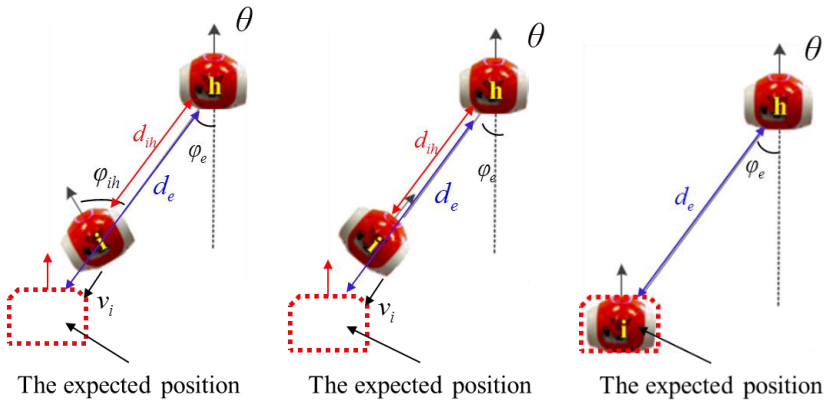
(1) Each NAO-*i* finds the NAO-*h* in its surroundings and faces to NAO-*h*, shown in Fig. 6-19(b). Then their relative angle φ_{ih} and relative distance d_{ih} are recorded, and the associated relative errors $e_{\varphi_{ih}} = \varphi_{ih} - \varphi_s$ and $e_{d_{ih}} = d_{ih} - d_s$ are calculated.

(2) If $e_{d_{ih}} < e_d$, it implies that the position of NAO-*i* is its destination, then the (4) step runs directly, otherwise, the NAO-*i* moves forward or backward along the line connecting the NAO-*i* and NAO-*h* according to the equation (6-9) to reduce the distance error, shown in Fig. 6-19(b).

(3) Since the movement of each robot mainly depends on itself, the distance control is an open control, we repeat step (1) until $e_{d_{ih}} < e_d$.

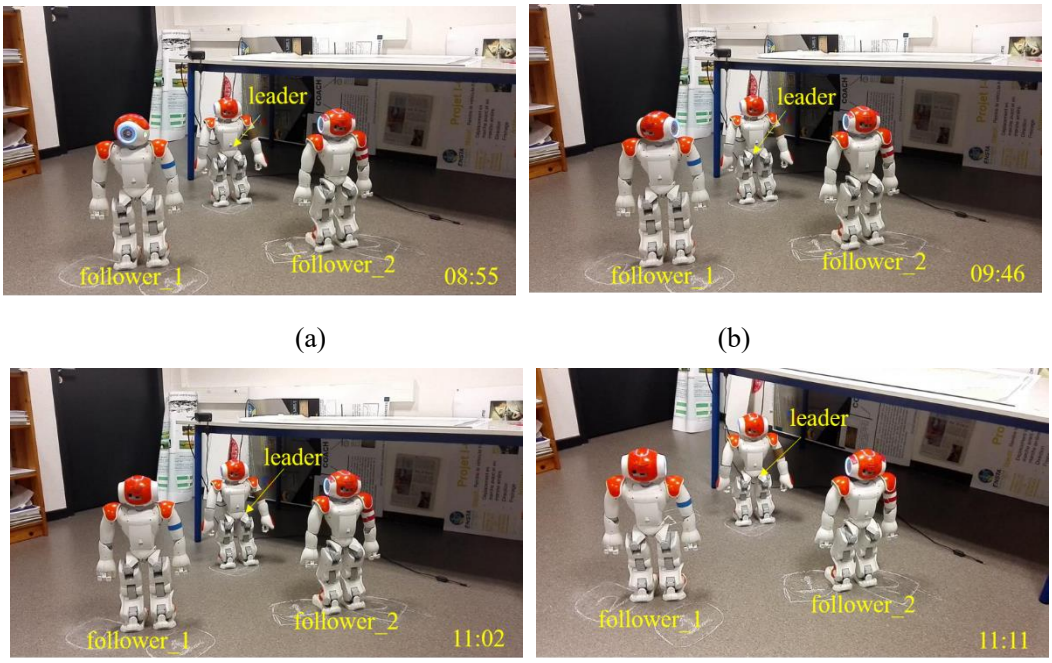
(4) The NAO-*i* turns the offset angle φ_s back to the orientation of the NAO-*h* θ, and stops,

shown in Fig. 6-19(c). After fine-tuning, the pyramid pattern is built satisfactorily, with $e_{d_{ih}} < e_d$ and $e_{\varphi_{ih}} < e_{\varphi}$ both in the allowed range, shown in Fig. 6-20.



(a) (b) (c)

Fig. 6-19 Illustration of the fine-tuning process. (a) The relative position and orientation between NAO and its parent in the preliminary triangle pattern. (b) NAO-*i* moves along the line connecting the NAO-*i* and NAO-*h* to reduce the distance error. (c) The final relative position and orientation between NAO and its parent in the final triangle pattern.



(a) (b)

(c) (d)

Fig. 6-20: The fine-turning process of the final triangle pattern after the basically pattern building shown in Fig. 6-17(h). (a) With the leader and brother's information, follower_1 finds that it is in the allowed range, while follower_2 fine tunes its position. (b) Follower_2 finishes its second loop within the tolerant range. (c) Follower_1 completes its re-check part and makes sure it arrives at the destination. (d) The triangle pattern is built well.

6.5 Chapter summary

In this chapter, the indoor experiments of multi-AUV system are designed, and the multi-mobile system is composed by 3 NAOs. The success of the pyramid building with 3 NAOs in the indoor environment demonstrates the feasibility and convergence of the proposed integrated image processing method (recognition algorithm and information extraction from images), the local position-based method, and the asynchronous limited communication distributed control method (the distance-based control method).

In the experiments, to get the neighboring NAOs' local information from a single image, the integrated image processing method is proposed:

① to recognize NAOs, a M-base is constructed to store the abundant features of NAOs. Considering there are only size transformations without rotation transformations among the features of NAOs in different images, a cross-correlation method is introduced to recognize the NAOs.

② to get the local information from a single image, the relative distance and angle models are established.

③ to overcome the lack of actual compass, the orientation of the NAO is estimated according to a pair of images with a visual compass and the information got by IMU.

To reduce the errors caused by the unstable movement due to the sensors and joints aging etc, visual sensor is introduced to estimate the rotation angle of NAO and to provide the angle feedback, and PD control method is used to control the rotation angle. Though the movement distances cannot be estimated directly, step-open-control is adopted to reduce the errors as possible as we can, and rotation control is introduced to correct the movement orientation at each step.

But the pyramid is sometimes built with big errors or even failed in some cases. The reasons include:

① the failure of recognition step due to the overlapping of NAOs and/or light noise in the images;

② the error of visual compasses because of the dynamic features (the mobile neighboring NAOs) in two adjacent images;

③ the large movement errors of the NAO caused by the NAOs aging. Further, the initial range among NAOs cannot be too large because of the limited recognition range for each NAO ($[0.5m,1.45m]$). Usually, the initial distances among NAOs are within $[0.6m,1.2m]$.

However, the stability of the pyramid building by NAOs can be improved by :

1) perfecting the recognition algorithm by increasing the features of masks in the M-base (such as colors, contour factors, etc.) and by reducing the influence of the light noise in the images;

2) improving the movement performance of NAOs by updating the feature extraction method of the visual compass to prevent the emergence of the situation in Fig. 6-14(b), and by adopting the closed-loop control to the distance control with the video cameras providing the feedback (such as

the size of features).

The indoor experiments have simulated the experiments with the constant ocean current in 2D underwater environment (all the AUVs locate at the same depth of oceans). The experiments have indirectly verified the feasibility and effectivity to build pyramids with the proposed coordination strategy when the AUVs information can be extracted well and the AUVs movements can be controlled well.

7 Conclusion and prospects

7.1 The main conclusions

The purpose of this thesis is to propose a kind of limited communication coordination strategy to allow the multi-AUV systems to achieve the formation building depending on the local information got by the on-board cameras (and compasses).

The proposed coordination strategy includes two parts: (1) defining a planar pyramid pattern; (2) studying and proposing corresponding formation building methods: ① for the obstacle-free environment, a local position-based method considering a collision avoidance strategy is studied and developed; ② for the general environment, asynchronous limited communication distributed control methods is proposed: according to the different kinds of information got by different sensors, two kinds of consensus algorithms are put forward to update the trajectories of robots: a. the asynchronous discrete consensus algorithm associated with the displacement-based control method; b. the asynchronous discrete consensus algorithm associated with the distance-based control method (the topologies of the both kinds of consensus algorithms are time-variant weighted digraphs).

(1) The definition of the planar pyramid pattern

To improve the scanning effectivity of multi-AUV systems, an inverted "V" pattern with large coverage is introduced. To improve the anti-interference ability of the pattern, redundant AUVs are filled in the internal part of the inverted "V" pattern, which can strengthen the constraints of AUVs by making the second smallest eigenvalue of Laplacian matrix to be larger. At the same time, these redundant AUVs can reduce the communication distance and the communication difficulty, and can be suitable to the short work range of underwater cameras. Finally, a planar pyramid pattern is defined.

(2) The presentation of the formation building methods

① The local position-based control method

In the limited communication condition, for the obstacle-free environment, the collision avoidance trajectories can be planned beforehand to prevent the trajectories update and reduce the time consumption. To do so, a common framed based on the local position relationship is established and the initial distribution can be obtained. Based on the graph theories, a collision avoidance strategy with the time complexity$O\ (n\log n)$ is proposed to optimize the pyramid distribution and optimize the corresponding positions of AUVs in the pyramid pattern, and then to get the expected position relationship.

To optimize the pyramid distribution based on the initial distribution, two methods are given to identify the orientation of the principal axis of the pyramid pattern: the method based on the CP and the method based on the PCA. To avoid complex algebra computation when optimizing the corresponding positions of AUVs in the pyramid pattern, the positions in both initial and pyramid

distributions are divided into the sub-groups with one or two positions as disjoint as possible according to the graph theories. The sub-groups with same orders located at the same area of both distributions are combined together to be the combined sub-groups. In each combined sub-group, each AUV identifies its destination in the pyramid distribution. The orders in pyramid pattern and IDs of AUVs compose to the expected position relationship among AUVs. And the connected lines between the positions of AUVs in both distributions are non-intercrossing, which are the non-intercrossing straight trajectories.

③ Asynchronous limited communication distributed control method

When the AUVs are stopped because of the static or dynamic obstacles, the local position-based control method becomes invalid. In these cases, each AUV needs to update its trajectory according to the neighbors' positions got by its on-board camera. To achieve it, the asynchronous limited communication distributed control method is proposed: with the displacement-based control method and with the distance-based control method. In the limited communication environment, each AUV independently senses its surrounding and computes and updates its trajectory according to the expected position relationship, until all the relative positions reach the agreement.

The two approaches (displacement-based and distance-based approaches) present some simulations: a. both methods update the trajectories by the asynchronous discrete consensus algorithm with the time-variant weighted digraph; b. each AUV updates its trajectory depending on the neighbors information got by its on-board sensors, and it does not need communication when updating trajectories. The differences between the two methods: a. in the displacement-based method, the neighbors information are the relative positions in global frame got by the on-board camera and compass, while in the distance-based method, the neighbors information are the distance vectors in each AUV's local frame got only by the on-board camera; b. each AUV at least senses one neighbors in the displacement-based method, while the AUV needs to detect at least two neighbors in the distance-based method.

Compared with the local position-based method, these two methods are more robust to the disturbances, the AUVs can re-plane their trajectories depending on its neighbors when meeting the obstacles. Nevertheless, the shortcoming of these two methods is high time cost because of the continuously updating of trajectories. And the cost of reducing the time consumption is to reduce the accuracy of formation building.

(3) Experimental results and analysis

The proposed coordination strategy is verified by the simulations first. Then its performance is tested by the indoor experiments with three NAOs.

The simulations are mainly done in a 3D simulated deep sea environment similar to the real underwater environment, in which the AUVs are modeled in 1:1 scale according to the real CISCREA. The authentic hydrodynamic model is introduced to the kinematic control. The repeated simulations demonstrate the feasibility and stability of the local position-based control method and the asynchronous limited communication distributed control method (with the displacement-based

method). To overcome the limitations in Blender (no more than 7 AUVs can be put in the fleet), and to test the algorithm performance with more AUVs to build the pyramid patterns, the distance-based control method is tested in Matlab with 6-10 AUVs. All the simulations show that the pyramid patterns can be built successfully, which demonstrates the feasibility and convergence of the coordination strategy.

To further evaluate the performance of the algorithm, the indoor experimental environment is constructed with limited communication similar to the underwater environment. Then 3 NAOs (the most we can have) are used to test the performance of the coordination strategy. Though the pyramid building sometimes fails because of the errors of recognition algorithm and visual compass, and the NAOs' movement, the pyramids can be built successfully in most situations.

(4) The creativities of this thesis

① Defining a pyramid pattern. Based on the inverted "V" pattern, the redundant AUVs are added to the internal part of the inverted "V" pattern to enforce the constraints among AUVs based on the property of the second smallest eigenvalue of Laplacian matrix. The redundant AUVs shorten the communication distance, which is suitable to the short working range of underwater camera and enhances the robustness. Unlimited by the number of AUVs, the pyramid pattern can cover a large range.

② Proposing the local position-based control method. Inspired by the position-based control method and graph theories, a collision avoidance with time complexity $O(n\log n)$ is proposed. Based on this strategy, each AUV can plan non-intercrossing straight trajectory to achieve the pyramid building in the obstacle-free environment with the shared relative positions perceived by cameras and compasses through the only once underwater acoustic communication.

③ Proposing the asynchronous limited communication distributed control method. With the known expected position relationship, an asynchronous discrete consensus algorithm with the time-variant weighted digraph is developed to update the trajectories. Each AUV independently constructs a time-variant sub-digraph depending on the local information got by cameras (and compasses), then estimates the time-variant weights and updates its trajectory. With the strongly connected topology, the convergence of the method is proofed.

7.2 The prospects

Developing the low-cost multi-AUV system is very meaningful for the research and exploration of oceans. As a part of the study of multi-AUV systems, we propose the following improvements.

(1) The image processing method should be more intelligent by:

① perfecting the image pre-processing method to reduce the influence of noises;

② recognizing correctly the AUVs from the overlapping situation to get precise positions and sizes;

③ improving the accuracy of visual compass by getting more features from images and matching up more features from two adjacent images.

(2) The performance of the formation building methods could be improved:

① The time consumption of asynchronous limited communication distributed method should be reduced.

② The robustness of the algorithm should be improved.

③ The methods should be tested and improved in the environment including static and dynamic obstacles.

(3) More real experiments should be planed:

① Experiments on land: the coordination strategy should be tested and enhanced by using more smart mini-cars.

② Experiments in the water: BlueROV2 is intended to be used to build pyramids in the water to test the performance of the algorithms.

(4) Formation maintaining and rebuilding control methods

With the built formation, multi-AUV systems should execute given tasks. Except for the formation building, the subsequent questions we would explore is how to maintain the formation and how to rebuild the formation when the formations are broken.

Reference

[1] Lapierre, Lionel. Underwater Robot Part I: current systems and problem pose. Mobile Robotics Towards New Applications. 2006: 335-360.

[2] Barron M G. Ecological impacts of the deep water horizon oil spill: implications for immunotoxicity [J]. Toxicologic Pathology, 2012, 40(2):315-320.

[3] Kathy Marks, Daniel Howden. The world's rubbish dump: a garbage tip that stretches from Hawaii to Japan. 2008. https://www.independent.co.uk/environment/green-living/the-worlds-rubbish-dump-a-tip-that-stretches-from-hawaii-to-japan-778016.html.

[4] Matsuno F, Sato N, Kon K, et al. Utilization of robot systems in disaster sites of the great eastern Japan earthquake [J]. Field and service Robotics, 2014, 92:1-17.

[5] How much of the ocean have we explored? 2018, https://oceanservice.noaa.gov /facts/ exploration.html.

[6] Blidberg, D. R. The development of autonomous underwater vehicles (AUVs); a brief summary [C]. In: Proceedings of the ICRA'01, Seoul, Korea, 2001: 1-12.

[7] Curtin, T., Bellingham, J., Catipovic, J., & Webb, D. Autonomous oceanographic sampling networks [J]. Oceanography, 1993, 6(6).

[8] G. Griffiths. Ocean science applications for autonomous underwater vehicles – the work plan for autosub-1 for 1997–2000 and beyond [J]. Unmanned underwater vehicle showcase, 1997, 9: 24-25

[9] Jos Melo, Anbal Matos. Survey on advances on terrain based navigation for autonomous underwater vehicles J], Ocean Engineering, 2017, 139: 250-264.

[10] Heidemann J, Stojanovic M, Zorzi M. Underwater sensor networks: applications, advances and challenges [J]. Philosophical Transactions, 2012, 370:1-20.

[11] Kamgar-Parsi, G., B. Johnson. D.L. Folds, and E.O. Belcher, High-resolution underwater acoustic imaging with lens-based systems [J]. Int. J. Imaging Sys. Technol., 1997, 8:377-385.

[12] Gracias N, Ridao P, Garcia R, et al. Mapping the moon: using a lightweight AUV to survey the site of the 17th century ship 'La Lune' [C]. Oceans. IEEE, 2013:1-8.

[13] J. H. Smart, Underwater optical communication systems Part 1: variability of water optical parameters [C]. Proc. IEEE MILCOM, 2005: 1140–46.

[14] J. Partana, J. Kurosea, and B. N. Levinea, A Survey of practical issues in underwater networks [J]. ACM Mobile Comp. Commun. Rev., 2007, 11:23–33.

[15] H. Riksfjord, O. T. Haug, and J. M. Hovem, Underwater acoustic networks — survey on communication challenges with transmission simulations [C]. Proc. 3rd IEEE SENSORCOMM, June 2009, pp. 300–5.

[16] Au W W, Nachtigall P E, Pawloski J L. Acoustic effects of the ATOC signal (75 Hz, 195 dB) on dolphins and whales [J]. Journal of the Acoustical Society of America, 1997, 101(1):2973-2977.

[17] Che X, Wells I, Dickers G, et al. Re-evaluation of RF electromagnetic communication in underwater sensor networks [J]. Communications Magazine IEEE, 2010, 48(12):143-151.
[18] Reza and J. Harms, Robust grid-based deployment schemes for underwater optical sensor networks [C]. Proc. 34th IEEE Conf. LCN, Oct. 2009: 641–48.
[19] D. Anguita, D. Brizzolara, and G. Parodi, Optical communication for underwater wireless sensor networks: A VHDL-implementation of a physical layer 802.15.4 Compatible [C]. Proc. IEEE Oceans Europe, 2009.
[20] S. Arnon and D. Kedar, Non-Line-Of-Sight underwater optical wireless communication network [C]. J. Optical Soc. Amer., 2009: 530–39.
[21] J. H. Smart, Underwater optical communication systems Part 2: basic design considerations [C]. Proc. IEEE MILCOM, 2005: 1700–1705.
[22] David Szondy. DARPA's TUNA program completes first phase. https://newatlas.com/darpa-tuna/47274/.
[23] Yang R. Modeling and robust control approach for underwater vehicles, Ph.D. dissertation. Ocean University of China, 2015.
[24] Kemp M, Hobson B, Meyer J, et al. MASA: a multi-AUV underwater search and data acquisition system [C]. Oceans. IEEE, 2002:311-315.
[25] Fratantoni D M, Haddock S H D. Introduction to the autonomous ocean sampling network (AOSN-II) program [J]. Deep Sea Research Part II Topical Studies in Oceanography, 2009, 56(3–5):61-61.
[26] Killeen T L. Ocean observatories initiative (OOI): advanced research tools for the ocean sciences [J]. Marine Technology Society Journal, 2010, 44(6):15-17.
[27] Schmidt H, Fox W L, Maguer A, et al. Generic oceanographic array technologies (GOATS)'98 - bi-static seabed scattering measurements using autonomous underwater vehicles [J]. 1998.
[28] Kalwa J. The GREX-Project: Coordination and control of cooperating heterogeneous unmanned systems in uncertain environments [C]. Oceans. IEEE Xplore, 2009:1-9.
[29] Birk A, Antonelli G, Caiti A, et al. The CO3AUVs (Cooperative Cognitive Control for Autonomous Underwater Vehicles) project: Overview and current progresses [C]. Oceans. IEEE, 2011:235-239.
[30] Liao Y. Japanese ARENA plan [J]. Ship & boat, 2005(4):20-25.
[31] Yan Z, Wang L, Wang T, et al. Polar Cooperative navigation algorithm for multi-unmanned underwater vehicles considering communication delays [J]. Sensors, 2018, 18(4):1044.
[32] Jaruwatanadilok S. Underwater wireless optical communication channel modeling and performance evaluation using vector radiative transfer theory [J]. IEEE Journal on Selected Areas in Communications, 2008, 26(9):1620-1627.
[33] Ergul O, Dinc E, Akan O B. Communicate to illuminate: State-of-the-art and research challenges for visible light communications [J]. Physical Communication, 2015, 17:72-85.
[34] Cui R, Li Y, Yan W. Mutual information-based multi-AUV path planning for scalar field

sampling using multidimensional RRT* [J]. IEEE Transactions on Systems Man & Cybernetics Systems, 2017, 46(7):993-1004.

[35] Annamalai A S, Sutton R, Yang C, et al. Robust adaptive control of an uninhabited surface vehicle [J]. Journal of Intelligent & Robotic Systems, 2015, 78(2):319-338.

[36] Ugrinovskii V A. Distributed robust filtering with H ∞, consensus of estimates [J]. Automatica, 2011, 47(1):1-13.

[37] H. M. La W. Sheng J. Chen. Cooperative and active sensing in mobile sensor networks for scalar field mapping [J]. IEEE Trans. Syst. Man Cybern. Syst. 2015, 45(1): 1-12.

[38] Hanff H, Kloss P, Wehbe B, et al. AUVx-A novel miniaturized autonomous underwater vehicle [C]. Oceans. IEEE, 2017.

[39] Fiorelli, E., Leonard, N.E., Bhatta, P., Paley, D.A., Bachmayer, R., Fratantoni, D.M.: Multi-AUV control and adaptive sampling in monterey bay [J]. IEEE journal of Oceanic Engineering, 2006, 31(4): 935–948.

[40] Jin, Y., Guo, H., Meng, Y.: A hierarchical gene regulatory network for adaptive multirobot pattern formation[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),2012, 42(3): 805–816.

[41] Li X., Zhu D., Qian Y.: A survey on formation control algorithms for multi-AUV system [J]. Unmanned Systems, 2014, 2(4): 351–359.

[42] Yu, S., Barca, J.C.: Autonomous formation selection for ground moving multi-robot systems [C]. In: Advanced Intelligent Mechatronics (AIM), 2015 IEEE International Conference on, 2015: 54–59.

[43] Lewis M.A.,Tan K.H.: High precision formation control of mobile robots using virtual structures [J]. Autonomous Robots, 1997, 4(4):387–403.

[44] Yu G., Brad S., Giampiero C., Marcello R., Larry R., Srikanth G., Sheng W.. Design and flight testing evaluation of formation control laws [J]. IEEE Transactions on Control Systems Technology, 2006, 14(6):1105–1112.

[45] Leonard N.E., Paley D.A., Lekien F., Sepulchre R., Fratantoni D.M., Davis R.E.: Collective motion, sensor networks, and ocean sampling [C]. Proceedings of the IEEE, 2007:48–74.

[46] Saad S. B., Zerr, B., Probst I., Dambreville F. Hybrid coordination strategy of a group of cooperating autonomous underwater vehicles [C]. IFAC-PapersOnLine, 2015, 48(5): 47–52.

[47] Girard A, Karlhedrick J. Formation control of multiple vehicles using dynamic surface control and hybrid systems [J]. International Journal of Control, 2003, 76(9-10):913-923.

[48] Mariottini G L, Morbidi F, Prattichizzo D, et al. Vision-Based Localization for Leader–Follower Formation Control [J]. IEEE Transactions on Robotics, 2009, 25(6):1431-1438.

[49] Cheng L, Wang Y J. Fault tolerance for communication-based multirobot formation [C]. International Conference on Machine Learning and Cybernetics. IEEE, 2004:127-132.

[50] Zhu F，Yin X．Formation control method of multiple mobile robots imported communication [J]．Application research of computer, 2007, 24, (10):59-60.

[51] Cheng L, Wang Y. Communication-based multiple mobile robots rigid formation control [C]. Control, Automation, Robotics and Vision Conference, 2004:729-734.
[52] Consolini L, Morbidi F, Prattichizzo D, et al. Brief paper: Leader-follower formation control of nonholonomic mobile robots with input constraints [J]. Automatica, 2008, 44(5):1343-1349.
[53] Morbidi F, Mariottini G L, Prattichizzo D. Brief paper: Observer design via Immersion and Invariance for vision-based leader-follower formation control [M]. Pergamon Press, Inc. 2010.
[54] Chen J, Sun D, Yang J, et al. Leader-follower formation control of multiple non-holonomic mobile robots incorporating a receding-horizon scheme [J]. International Journal of Robotics Research, 2010, 29(6):727-747.
[55] Balch T, Arkin R C. Behavior-based formation control for multi-robot teams [J]. IEEE Transactions on Robotics & Automation, 1998, 14(6):926-939.
[56] Lyons D M, Arkin R C, Jiang S, et al. Performance verification for behavior-based robot missions [J]. IEEE Transactions on Robotics, 2017, 31(3):619-636.
[57] Lewis M A, Tan K H. High precision formation control of mobile robots using virtual structures [J]. Autonomous Robots, 1997, 4(4):387-403.
[58] Ren W, Beard R W. Formation feedback control for multiple spacecraft via virtual structures [J]. Control Theory and Applications, IEEE Proceedings -, 2004, 151(3):357-368.
[59] Min G P, Jeon J H, Min C L. Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing [C]. IEEE International Symposium on Industrial Electronics, 2001. Proceedings. ISIE. IEEE, 2001, 3:1530-1535.
[60] Min C L, Min G P. Artificial potential field based path planning for mobile robots using a virtual obstacle concept [C]. IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2003:735-740.
[61] Khatib O. Real-Time obstacle avoidance for manipulators and mobile robots [M]. Autonomous Robot Vehicles. Springer New York, 1986:500-505.
[62] Oh, K.K., Park, M.C., Ahn, H.S.: A survey of multi-agent formation control [J]. Automatica, 2015, 53: 424–440.
[63] Ren W, Cao Y. Distributed coordination of multi-agent networks [M]. Springer, 2011.
[64] Chiem S.Y., Cervera E.: Vision-based robot formations with bezier trajectories [C]. In: in Eighth Conference on Intelligent Autonomous System, 2004.
[65] Fredslund J., Mataric M.J. A general algorithm for robot formations using local sensing and minimal communication [J]. IEEE transactions on robotics and automation, 2002, 18(5): 837–846.
[66] Dimarogonas D V, Kyriakopoulos K J. A connection between formation infeasibility and velocity alignment in kinematic multi-agent systems [J]. Automatica, 2008, 44(10):2648-2654.
[67] Ji M, Egerstedt M. Distributed coordination control of multi-agent systems while preserving connectedness [J]. IEEE Transactions on Robotics, 2007, 23(4):693-703.
[68] Aoki, I.: A simulation study on the schooling mechanism in fish [J]. Bulletin of the Japanese Society of Scientific Fisheries, 1982, 48(8): 1081–1088.

[69] Huth A., Wissel C. The simulation of the movement of fish schools [J]. Journal of theoretical biology, 1992, 156(3): 365–385.
[70] Inada Y., Kawachi K. Order and flexibility in the motion of fish schools. Journal of theoretical biology [J], 2002, 214(3): 371–387.
[71] Nikou, C. K. Verginis, and D. V. Dimarogonas, Robust distance-based formation control of multiple rigid bodies with orientation alignment [C]. IFAC, 2017: 15458–15463.
[72] Dimarogonas D V, Johansson K H. On the stability of distance-based formation control [C]. Decision and Control, 2008:1200-1205.
[73] Dimarogonas D V, Johansson K H. Further results on the stability of distance-based multi-robot formations [C]. American Control Conference, 2009:2972-2977.
[74] K. K. Oh and H. S. Ahn, Distance-based control of cycle-free persistent formations [C]. In Intelligent Control, 2011: 816–821.
[75] Oh K, Ahn H. Distance-based undirected formations of single-integrator and double-integrator modeled agents in n-dimensional space [J]. International Journal of Robust & Nonlinear Control, 2014, 24(12):1809-1820.
[76] Xia Y, Na X, Sun Z, et al. Formation control and collision avoidance for multi-agent systems based on position estimation [J]. ISA Transactions, 2016, 61:287-296.
[77] Wang J, Xin M. Multi-agent consensus algorithm with obstacle avoidance via optimal control approach [C]. American Control Conference. IEEE, 2011:2783-2788.
[78] Wang J, Xin M. Optimal consensus algorithm integrated with obstacle avoidance [J]. International Journal of Systems Science, 2013, 44(1):166-177.
[79] Seo J, Kim Y, Kim S, et al. Consensus-based reconfigurable controller design for unmanned aerial vehicle formation flight [J]. Proceedings of the Institution of Mechanical Engineers Part G Journal of Aerospace Engineering, 2012, 226(7):817-829.
[80] Davis D T, Chung T H, Clement M R, et al. Consensus-based data sharing for large-scale aerial swarm coordination in lossy communications environments [C]. IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2016:3801-3808.
[81] Liu X, Sun J, Dou L, et al. Leader-following consensus for discrete-time multi-agent systems with parameter uncertainties based on the event-triggered strategy [J]. Journal of Systems Science & Complexity, 2017, 30(1):30-45.
[82] Lu W, Atay F M, Jost J. Consensus and synchronization in discrete-time networks of multi-agents with stochastically switching topologies and time delays [J]. Networks & Heterogeneous Media, 2012, 6(2):329-349.
[83] Tang H, Li T. Convergence rates of discrete-time stochastic approximation consensus algorithms: Graph-related limit bounds [J]. Systems & Control Letters, 2018, 112:9-17.
[84] Yang R., Clement B., Mansour A., Li M., Wu N. Modeling of a complex-shaped underwater vehicle for robust control scheme [J]. Journal of Intelligent & Robotic Systems, 2015, 80(3-4): 491–506.

[85] Olfati-Saber R., Fax J.A., Murray R.M. Consensus and cooperation in networked multi-agent systems [C]. Proceedings of the IEEE, 2007: 215–233.
[86] Beard R.W., Stepanyan V.: Information consensus in distributed multiple vehicle coordinated control [C]. In: Decision and Control, 2003: 2029–2034.
[87] Jadbabaie A., Lin J., Morse A.S. Coordination of groups of mobile autonomous agents using nearest neighbor rules [J]. IEEE Transactions on automatic control, 2003 48(6), 988–1001.
[88] Olfati-Saber R.,Murray R.M. Consensus problems in networks of agents with switching topology and time-delays [J]. IEEE Transactions on automatic control, 2004, 49(9), 1520– 1533.
[89] Ren W., Beard R.W., Atkins E.M. A survey of consensus problems in multi-agent coordination [C]. In: American Control Conference, 2005: 1859–1864.
[90] Horn, R.A., Johnson, C.R. Matrix analysis [M]. Cambridge university press, 2012.
[91] Yan W, Wu W. Data structure (C-Language) [M]. Computer education press, 2012, 168(12):62-62.
[92] Fiedler M. Algebraic connectivity of graphs [J]. Czechoslovak Mathematical Journal, 1973, 23(23):298-305.
[93] Kim, Y., Mesbahi, M. On maximizing the second smallest eigenvalue of a state dependent graph laplacian [J]. IEEE transactions on Automatic Control, 2006, 51(1): 116–120.
[94] Nighot M.K.,Patil V.H., Mani G.S. Multi-robot hunting based on swarm intelligence [C]. In: Hybrid Intelligent Systems (HIS), 12th International Conference on, 2012: 203–206.
[95] Wang X., Zerr B., Thomas H., Clement B., Xie Z. Pattern formation for a fleet of AUVs based on optical sensor [C]. OCEANS'17, 2017:1-9.
[96] Chiang J Y, Chen Y C. Underwater image enhancement by wavelength compensation and dehazing [J]. IEEE Transactions on Image Processing, 2012, 21(4):1756-1769.
[97] S. S. Thakare A. M. Sahu. Underwater image de-noising by using adaptive wavelet transformation [J]. International Journal of Innovative Research in Computer and Communication Engineering, 2015, 3(4): 2865-2871.
[98] SAUC-E 2016 Mission and Rules1. http://sauc-europe.org/2016_Mission_Rules_1.0.pdf
[99] Subashini. Comparison of Filters used for Underwater Image Pre-processing[J]. International Journal of Computer Science & Network Security, 2010, 10(1): 58-65.
[100] Srividhya K, Ramya M M. Performance analysis of pre-processing filters for underwater images [C]. International Conference on Robotics, Automation, Control and Embedded Systems. IEEE, 2015:1-7.
[101] Voicu L I. Practical considerations on color image enhancement using homomorphic filtering [J]. Journal of Electronic Imaging, 1997, 6(1):108-113.
[102] Garcia R, Nicosevici T, Cufi X. On the way to solve lighting problems in underwater imaging [C]. Oceans. IEEE, 2003:1018-1024.
[103] Hou G J, Luan X, Song D L. A study on color model selection for underwater color image preprocessing [C]. IEEE International Conference on Cyber Technology in Automation, Control,

and Intelligent Systems, 2015:1456-1461.
[104] C. Tomasi, and R. Manduchi. Bilateral filtering for gray and color images [C]. 1998 IEEE International Conference on Computer Vision, 1998:839-846.
[105] H. Dai, X. Gu. A New method based on bilateral-filtering for illumination invariant face recognition [C]. IEEE Third International Conference on Information and Computing, 2010:242-245.
[106] K. N. Chaudhury, D. Sage, and M. Unser. Fast O(1) bilateral filtering using trigonometric range kernels [J]. IEEE Transactions on Image Processing, 2011:3376-3382.
[107] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990, 12(7):629-639.
[108] Y. You, W. Xu, A. Tannenbaum, and M. Kaveh, Behavioral analysis of anisotropic diffusion in image processing [J]. IEEE Transactions on Image Processing, 1996, 5(11):1539-1553.
[109] S. Kim, S. Chung, S. Jung, and S. Cho. An improved illumination normalization based on anisotropic smoothing for face recognition [J]. International Scholarly and Scientific Research & Innovation, 2008, 2(1):81-87.
[110] S. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, et al. Adaptive histogram equalization and its Variations, Computer Vision [J]. Graphics and Image Processing, 1987, 39(3): 355–368.
[111] W. N. J. Hj W. Yussof, M. S. Hitam, E. A. Awalludin, and Z. Bachok. Performing contrast limited adaptive histogram equalization technique on combined color models for underwater image enhancement [J]. International Journal of Interactive Digital Media, 2013, 1(1):1-6.
[112] R. Beohar, P. Sahu. Performance analysis of underwater image enhancement with CLAHE 2D median filtering technique on the basis of SNR, RMS error, mean brightness [J]. International Journal of Engineering and Innovative Technology, 2013, 3(2):525-528.
[113] C. Srisailam, P. Sharma, and S. Suhane. Color image denoising using wavelet soft thresholding [J]. International Journal of Emerging Technology and Advanced Engineering, 2014, 4(7): 475-478.
[114] He K, Sun J, Tang X. Single image haze removal using dark channel prior [J]. IEEE Transactions Pattern Analysis and Machine Intelligence, 2011, 33(12):2341-2353.
[115] Goldstein E B. Sensation and perception [M]. ACADEMIC, 1984.
[116] Preetham A J, Shirley P, Smits B. A practical analytic model for daylight [C]. Conference on Computer Graphics and Interactive Techniques. ACM Press/Addison-Wesley Publishing Co. 1999:91-100.
[117] Shwartz S, Namer E, Schechner Y Y. Blind Haze separation [C]. Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. IEEE, 2006:1984-1991.
[118] J. T. Houghton, The physics of atmospheres [M], 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2001, ch. 2.
[119] S.Q.Duntley. Light in the sea [J]. Journal of the Optical Society of America,1963, 53(2): 214–233.

[120] L. A. Torres-Méndez and G. Dudek. Color correction of underwater images for aquatic robot inspection [C]. In Proc. EMMCVPR, 2005: 60–73.
[121] Tyler J E. Optical oceanography [J]. Atlas - News Supplement to Earth-Science Reviews, 1968, 4(4):A302-A302.
[122] Fujinaga N, Yamauchi Y, Kijima S, et al. Pattern formation by oblivious asynchronous mobile robots [C]. International Conference on Distributed Computing. 2012:312-325.
[123] Wold S., Esbensen K., Geladi P. Principal component analysis [J]. Chemometrics and intelligent laboratory systems, 1987, 2(1-3): 37–52.
[124] Liu, T. Application of Markov chains to analyze and predict the time series [J]. Modern Applied Science, 2010, 4(5): 162–166.
[125] Sinclair, A. Cs294 Markov chain monte carlo: Foundations & applications, 2009.
[126] Ren, W., Beard, R., McLain, T. Coordination variables and consensus building in multiple vehicle systems [J]. Cooperative control, 2005:439–442.
[127] Wang L. State consensus for multi-agent systems with switching topologies and time-varying delays [J]. International Journal of Control, 2006, 79(10):1277-1284.
[128] Feng Xiao, Long Wang. Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays [J]. IEEE Transactions on Automatic Control, 2008, 53(8):1804-1816.
[129] Ren, W., Beard, R., McLain, T. Coordination variables and consensus building in multiple vehicle systems [J]. Cooperative control, 2005:439–442.
[130] Ren W, Beard R W. Consensus seeking in multi-agent systems under dynamically changing interaction topologies [J]. IEEE Transactions on Automatic Control, 2005, 50(5):655-661.
[131] Moreau L. Stability of multi-agent systems with time-dependent communication links [J]. IEEE Transactions on Automatic Control, 2005, 50(2):169-182.
[132] Ren, W., Beard, R., & McLain, T. Coordination variables and consensus building in multiple vehicle systems. Cooperative control, 2005:439–442.
[133] Benozzi, L.: A detection strategy for line formation of a team of humanoid robots, Master dissertation. University of Florence, 2017.
[134] Aldebaran: Nao software 1.14.5 documentation/hardware/nao technical overview/joints. http://doc.aldebaran.com/1-14/family/robots/ joints_robot.html, 2013.
[135] Aldebaran: Nao software 1.14.5 documentation/reference/naoqi api/naoqi vision/ alvisual compass. http://doc.aldebaran.com/1-14/naoqi/ vision/alvisualcompass.html? highlight=compa- ss, 2013.

Publication and Conference

1. Wang Xiaomin*, Zerr Benoit, Thomas Helene, et al. Pattern formation of multi-AUV systems with the optical sensor based on displacement-based formation control [J]. International Journal of Systems Science, 2020, 51(2): 348-367. (SCI, IF: 2.149)
2. Wang Xiaomin*, Benozzi Lorenzo, Zerr Benoit, Xie Zexiao, et al. Formation building and collision avoidance for a fleet of NAOs based on optical sensor with local positions and minimum communication [J]. Science China Information Sciences, 2019, 62: 052205:1 – 052205:23. (SCI, IF: 2.731)
3. Xiaomin WANG*, Benoit ZERR, Helene THOMAS, Benoit CLEMENT, Zexiao XIE. Pattern formation for a fleet of AUVs based on optical sensor[C]. Oceans. 2017:1-9.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

ENSTA Bretagne

Titre : Algorithmes de perception autonomes pour une meute de robots sous-marins : stratégie de coordination à partir des images acquises par caméras

Mots clés : Système multi-AUV, formation en meutes, coordination de robots, traitement d'images sous - marines;

Résumé : Au cours des dernières années, les systèmes multi - AUV ont été de plus en plus étudiés dans le domaine de l'exploration des océans en raison de leurs avantages, leur efficacité, leur faible performance et leur haute tolérance aux capteurs à faible coût. Le contrôle de formation est devenu un nouvel axe de recherche. Contrairement aux systèmes multi-agents terrestres ou aériens, les difficultés de communication sous - marine constituent un défi majeur pour ces systèmes. Les communications acoustiques étant majoritairement limitées par la bande passante et le bruit, les communications optiques peuvent constituer une alternative. Leur courte portée peut être compensée par l'association de plusieurs robots. L'objectif de cette thèse s'est donc focalisé sur la formation en meute à l'aide des caméras fixées sur les robots.
Premièrement, on a analysé la faisabilité et l'efficacité d'un traitement d'information basé sur les images acquises par les robots et on a étudié les mécanismes de construction de la formation basés sur des positions locales obtenues par traitement d'images. Ensuite, une stratégie de coordination comprenant deux volets a été proposée: la conception d'une formation pyramidale plane robuste et l'étude de deux méthodes de contrôle de la formation, une méthode de contrôle reposant sur des positions locales et une autre faisant intervenir un algorithme de consensus asynchrone discret. Les méthodes proposées sont évaluées en simulation sous Blender et Matlab ainsi qu'à travers des expérimentations en environnements intérieurs avec canaux de communication limités avec des robots NAO. Les simulations et les expériences répétées ont permis de vérifier la faisabilité et la stabilité de la stratégie de coordination proposée.

Title : Autonomous Perception Algorithms for a Team of Underwater Robots : Coordination strategy based on onboard cameras

Keywords : Multi-AUV System, Formation Building, Consensus Algorithm, Coordination of robots, underwater image processing

In recent years, multi-AUV systems are getting increasing attentions in the ocean exploration because of the potential advantages, such as high efficiency and high tolerance to the lowperformance and low-cost sensors. Thus, formation control becomes a new research hotspot. Different to the land or aerial multi-agent systems, communication difficulty in the water is a big challenge to the multi-AUV system. Since the main acoustic communication is limited by bandwidth and noises, scholars begin paying attentions on the underwater optical communication. It can reduce the difficulty of acoustic communication. Its short working range can be extended by associating with multi-AUV systems, but the relevant studies are few. Therefore, the purpose of this work is to achieve the formation building based on the on-board camera for the multi-AUV system.
First, we analyze the feasibility and effectiveness of underwater camera sensing information and study the formation building mechanism based on the local positions got from cameras. Then a coordination strategy based on the on-board cameras is proposed, including two parts: designing a robust planar pyramid formation and developing two kinds of formation control methods: local position-based control method and asynchronous discrete consensus-based formation control method with local information. Finally, the proposed new coordination strategy is tested in the software (Blender and Matlab) with multi-AUV systems and in the built limited-communication indoor environment (similar to the underwater environment) with NAO robots. After repeating simulations and experiments, the feasibility and stability of coordination strategy is verified.