Towards coordinated multi-agent exploration problem via segmentation and reinforcement learning

Chen, Zichen

2019

Chen, Z. (2019). Towards coordinated multi-agent exploration problem via segmentation and reinforcement learning. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/137152>

<https://doi.org/10.32657/10356/137152>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 18:32:38 SGT

TOWARDS COORDINATED MULTI-AGENT
EXPLORATION PROBLEM VIA
SEGMENTATION AND REINFORCEMENT
LEARNING

A thesis submitted to
School of Computer Science and Engineering
Nanyang Technological University

by

CHEN ZICHEN

in partial fulfilment of the requirements
for the degree of Master of Engineering

August 2019

*To my beloved parents and grandparents,
Chen Guangqi, Shao Xiaowei, Chen Jizhao and Xu jinyu.
For their understanding, faith and endless love.*

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

23 August 2019

.....

Date

A handwritten signature in black ink, appearing to be '陈子晨' (Chen Zichen), written over a dotted line.

CHEN ZICHEN

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

23 August 2019
.....
Date

.....
Prof. Tan Ah-Hwee

Authorship Attribution Statement

This thesis contains material from 1 paper accepted at the conference and 1 paper under review in the following peer-reviewed journal in which I am listed as an author.

Chapter 3 is under review as Luo T, Chen Z, Subagdja B, Tan A H. Real-time Hierarchical Map Segmentation for Coordinating Multi-Robot Exploration[J]. IEEE Access, 2019.

The contributions of the co-authors are as follows:

- Prof. Tan suggested the initial project direction and edited the manuscript drafts.
- I wrote the drafts of the manuscript with Luo Tianze. The manuscript was revised together with Luo Tianze and Dr. Subagdja.
- I co-designed the study with Luo Tianze and implemented all the benchmark experiments, including Morphological, Distance-based, Voronoi, Feature-based segmentation methods and frontier-based exploration method. I also analyzed the results.
- Luo Tianze implemented the proposed methods.

Chapter 4 is accepted as Chen Z, Subagdja B, Tan A H. End-to-end Deep Reinforcement Learning for Multi-agent Collaborative Exploration[C]. 2019 IEEE International Conference on Agents (ICA). IEEE, 2019.

- Prof. Tan provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by Dr. Subagdja.
- I designed the study and implemented all the proposed and benchmark methods.

23 August 2019

.....

Date

.....
CHEN ZICHEN

Acknowledgement

This work would not have been possible without the inspiration and help from many people around me. I am especially indebted to my supervisor, Prof. Tan Ah-Hwee, who has been supportive of my career goals and worked actively to provide me with the protected academic time to pursue these goals. I am also indebted to Prof. Sinno Jialin Pan, for all his guidance and support throughout my research work. He will always be my inspiration. They taught me more than I could ever give them credit for here, and have shown me what a good scientist should be. I sincerely thank Dr. Budhitama Subagdja and Dr. Iain Philip Werry, for their helpful comments on my thesis. I am grateful for their kindness and encouragements during my research work here.

I am especially grateful to all of those with whom I have had a passionate and enjoyable time to work during related projects. My close lab mates have provided me with extensive professional and inspired suggestions and were always ready for a deep discussion about research work. Particular thanks to the Computational Intelligence Laboratory (CIL) in the School of Computer Science and Engineering (SCSE) at NTU. CIL is an amazing lab full of incredible students and staff. As a student here, I really feel proud and grateful for the unforgettable experiences.

Nobody has been more important to me in the pursuit of my dream than the members of my family. I would like to thank my parents, Chen Guangqi and Shao Xiaowei, grandparents, Chen Jizhao and Xu Jinyu, aunt, Chen Wenhua, whose understanding love and selfless support are with me in whatever I pursue. They gave me the life, patient guidance and strong backing.

As a business background student, it is not easy to obtain a research master of computer science and engineering. Thanks for myself for never giving up and chasing the dream bravely. I had difficulties, problems and tears during my research time, but I never lost the enthusiasm for the happiness and exploring science. Life is fantastic!

Contents

Statement of Originality	i
Supervisor Declaration Statement	ii
Authorship Attribution Statement	iii
Acknowledgement	iv
List of Abbreviations	viii
Lists of Figures	x
Lists of Tables	xi
Abstract	xii
1 Introduction	1
1.1 Background and Motivations	1
1.2 Issues and Challenges	2
1.3 Thesis Objective	4
1.4 Approach and Methodology	4
1.4.1 Segmentation-based Approach	4
1.4.2 Learning-based Approach	5
1.5 Contributions	6
1.6 Outline of Thesis	7
2 Literature Review	9
2.1 Multi-agent Exploration Problem	10
2.1.1 Segmentation-based Exploration	10
2.1.2 Topological-based Exploration	11
2.1.3 Learning-based Exploration	12
2.2 Indoor Room Segmentation	13

2.3	Representation Learning and Decision Making in Dynamic Environments	14
2.3.1	Convolutional Neural Network	14
2.3.2	Deep Reinforcement Learning	15
2.3.3	Curriculum Learning	19
2.3.4	Intrinsic-driven Learning	19
3	Hierarchical Adaptive Clustering for Indoor Room Segmentation¹	21
3.1	HAC Methodology and Model	24
3.1.1	HAC Architecture	24
3.1.2	Low-level ART network	24
3.1.3	Upper-level ART network	26
3.2	HAC Mechanism	27
3.2.1	Merge fragmentary segments Criteria	27
3.3	HAC Segmentation-based Exploration	28
3.3.1	Exploration in the Partially Known Environment	31
3.3.2	Exploration in the Unknown Environment	34
3.4	Experiments and Discussion	36
3.4.1	Exploration Settings	36
3.4.2	Evaluation in HAC Segmentation	38
3.4.3	Evaluation in HAC segmentation-based Exploration	40
3.5	Conclusion	44
4	End-to-end Deep Reinforcement Learning for Multi-agent Collaborative Exploration²	47
4.1	CMAPO Model	49
4.1.1	Representation Learning	49
4.1.2	Proximal Policy Optimization Based Learning System	50
4.1.3	Problem Setup	52
4.2	Enhancement of CMAPO	53
4.2.1	Curriculum Learning Based Sampling	53
4.2.2	Intrinsic-driven Exploration	55
4.3	Experiments and Results	56
4.3.1	Experiment Environment	56

---

¹The work is still under review as Luo T, Chen Z, Subagdja B, Tan A H. Real-time Hierarchical Map Segmentation for Coordinating Multi-Robot Exploration[J]. IEEE Access, 2019.

²The work is accepted as Chen Z, Subagdja B, Tan A H. End-to-end Deep Reinforcement Learning for Multi-agent Collaborative Exploration[C]. 2019 IEEE International Conference on Agents (ICA). IEEE, 2019.

4.3.2	Curriculum Experiment Environment	57
4.3.3	Evaluation of CMAPPO	57
4.4	Conclusion	60
5	Conclusion and Future Work	61
5.1	Conclusion	61
5.2	Future Directions	62
	Bibliography	64

List of Abbreviations

ABBREVIATIONS	FULL EXPRESSIONS
Adam	Adaptive Moment Estimation
ART	Adaptive Resonance Theory
CNN	Convolutional Neural Networks
KL	Kullback–Leibler Divergence
MDP	Markovdecision Process
NP	Non-deterministic Polynomial
PPO	Proximal Policy Optimization
SLAM	Simultaneous Localization and Mapping

List of Figures

2.1	The overall taxonomy of Literature Review	9
3.1	The HAC architecture consisting of two modified ART neural networks: low-level and upper-level ART. Each ART consists of F_1 layer (input) and F_2 layer (category) (F_1' and F_2').	25
3.2	The map segmentation procedure by HAC. (a) Randomly sampled 7000 points on the empty space of the map. (b) Groups of low-level clusters wherein each group represents a segmented region. (c) The output of upper-level clustering in which each blue point represents the segment center. A graph is constructed by connecting the segment centers. (d) The final segmentation output after the fragmented segments are merged.	28
3.3	The procedure of hierarchical clustering. (a) illustrates the procedure of sampling points. (b) shows the <i>lower layer</i> clusters while (c) shows <i>upper layer</i> clusters. (d) demonstrates the topological information output of final clusters. Figure (e) to (h) shows the procedure of incremental segmentation during exploration: (e) sample points on the newly explored area. (f) and (g) Hierarchical clustering, which are similar to (b)(c). (h) Compute new segment centres and merge them with the segment graph.	33
3.4	Examples of comparing HAC segmentation with four state-of-the-art segmentation methods. From left to right column: (1) the ground truth; (2) Morphological segmentation; (3) Distance-based segmentation; (4) Voronoi graph-based segmentation; (5) Feature-based room segmentation; (6) HAC segmentation.	39
3.5	Average travel distance gain for HAC segmentation-based multi-agent map exploration under partially known map with nodes' weight and without nodes' weight, compared with the frontier based multi-agent exploration method.	42

3.6	Average travel distance gain for HAC segmentation-based exploration method, compared with the Morphological, Feature, Voronoi, Distance based exploration methods.	43
3.7	Average travel distance gain for real-time HAC segmentation-based exploration under the unknown map, compared with the frontier-based multi-agent exploration.	44
3.8	Average travel distance gain for real-time clustering based exploration compared with frontier based exploration method on maps with different complexities.	45
4.1	The architecture of the CMAPPO. In decomposed representation, the information of obstacles is incrementally increased during the unknown environment exploration.	49
4.2	The architecture of curriculum learning	55
4.3	The sample maps of different stage levels. The top is the maze-like environment, the bottom is the office-like environment.	57
4.4	(a) and (b) are the example of randomly generated maze-like scene and office-like scene respectively.	58
4.5	The AMD of average two training samples (maze-like maps) with CMAPPO method in 4 stage levels compared to the model without curriculum learning and intrinsic reward (No C&I). The total training episodes are 35,000, averaged over 100 runs.	59

List of Tables

1.1	Pros and cons of proposed approaches.	7
3.1	Average Precision and Recall ($\pm$ Standard Deviation) of the Segmentation Methods on 30 Maps	38
4.1	Performance metrics of AMD evaluated for CMAPPO method, and Frontier-based method on the target unknown environment with different number of agents (the results are rounded off to the nearest integer).	60

Abstract

Exploring an unknown environment by multiple autonomous robots is a major challenge in the robotics domain. The robot or agent needs to incrementally construct a model or a map representation of the environment while performing its domain tasks like surveillance, search and rescue tasks, and cleaning. What the robot should do or where it should go to visit next can only be determined after the map is constructed at least partially.

The typical approach is by taking a frontier point which is located in the boundary between a known area and an unknown region as the target location to visit. This point is selected from other frontiers as revealed whenever the robots observe the environment. However, when multiple robots are involved, the task becomes more challenging as they have to explore the unknown environment as efficient and fast as possible while avoiding conflicts or interferences among the robots that can reduce the efficiency.

Although coordinating a team of autonomous robots to explore an unknown environment can be done in an efficient way, partitioning the map of the environment into separate regions or segments as the targets allocated to the robots to visit is an efficient approach. The partitioning must be performed continually and incrementally. There is a trade-off that generating many small segments can provide more details of the environment, but may lose the representation of larger areas that are useful and relevant to the exploration task at hand. A Hierarchical Adaptive Clustering (HAC) segmentation of the indoor environment is introduced in this thesis that can strike a balance between fine-grained clustering and generalized segmentation during the exploration. With the HAC approach, an effective multi-agent task allocation approach is developed, wherein the partitioning and allocation processes can be performed continually and incrementally in real-time. Experimental results on HAC-based exploration method shows that it is comparable with other state-of-the-art approaches including Frontier-based allocation and Voronoi-based exploration. The model outperforms the others in terms of meaningful topological clusters and efficient exploration.

However, non-learning based methods usually employ a fixed strategy to allo-

cate the robots or agents to explore selected locations that sometimes can not handle the unpredictable and dynamic situations well. These methods can be effective in a single robot case, but assigning multiple robots to explore different locations is challenging since individual robots may interfere with others, making the overall tasks less efficient. A learning-based approach is proposed to solve those issues in this thesis. The algorithm is called CNN-based Multi-agent Proximal Policy Optimization (CMAPPO), which is for allocating multiple robots to explore different environments while over time improving their strategies to allocate the tasks more efficiently and flexibly. This algorithm combines CNN to process multi-channel visual inputs from the observed environment, curriculum learning for improving learning efficiency, and PPO algorithm for motivation based reinforcement learning. Based on the evaluation, the CMAPPO can learn a more efficient strategy for multiple robots (the robot is named agent in the rest of this thesis) to explore the environment than the conventional frontier-based method.

This thesis introduces a novel indoor space segmentation-based exploration method which is based on topological clusters of an enclosed environment to perform multi-agent exploration. Considering the dynamic situations in the environment, this thesis further develops a new end-to-end deep reinforcement learning architecture for multi-agent exploration strategy by using Convolutional Neural Network (CNN) and Proximal Policy Optimization (PPO).

Keywords: Multi-agent exploration, Deep learning, Reinforcement learning.

Chapter 1

Introduction

1.1 Background and Motivations

Multi-agent exploration problem is one of the core tasks in the field of mobile robots, which often involves motion planning, sensing, and uncertainty management [1]. In recent years, advanced applications such as search and rescue, reconnaissance[2], mowing, and cleaning robots[3] are built upon exploration strategies. Such related applications play a key role in our daily life. However, exploring an enclosed environment by a team of autonomous agents has remained a challenging problem. Since the prior knowledge relevant to the task or the model of the environment can be partially or totally unavailable, the agents must map or construct the model of the environment incrementally. In this way, the appropriate exploration tasks can be assigned to the right agents. The agents can only plan the exploration tasks or decide where to go when the environment map is available at least partially.

Towards the exploration problem, the first step is to obtain knowledge about the environment and build the exploration strategy via limited information. It is very important to learn about the environment and make the exploration decision simultaneously. A typical approach is to find or select appropriate frontier points which are located at the boundary between known and unknown areas [4]. When one agent reaches the frontier points, it observes the surroundings and obtains the information about new known areas so that the new frontier points can be assigned to the agent. All the unexplored areas can be uncovered when the agent repeats this process. In order to improve the efficiency, Fox et al. introduce that an agent is allocated to the frontier point with least-cost to visit (the nearest frontier location to the agent).

Some ad-hoc strategies or heuristics can be applied to optimize the selection or allocation of the targets to the multi-agent system. First, partitioning the scanned areas into different regions or segments. Since the allocation to assign the agents to

explore different locations can be done more efficiently with the topological structure of the segments. Second, the exploration strategy can be considered not only with a single agent but also with the aspect of being a team. In a complex and highly dynamic environment, it is almost impossible to optimally solve the multi-agent exploration problem with basic frontier-based strategies. Therefore, depending on those considerations, the combination methodologies need to be developed for finding good exploration strategies.

This thesis focuses on constructing a good representation of the environment and developing the coordinated multi-agent strategies for further improving collaboration exploration. First, this thesis studies the indoor environment segmentation methods, in particular finding a proper way for real-time indoor exploration tasks. Second, the frontier-based approach is extended to multi-consideration model, which is based on a cost-utility measure. Third, this thesis studies how to optimally solve this problem with a deep reinforcement learning method and further improve the learning performance. Detailed problem descriptions and the contributions are as follows.

1.2 Issues and Challenges

Semantically segmenting a map of an indoor environment is an important process in exploration tasks. For example, applications like floor plan identification or robotic path planning require environment segmentation to complete the task efficiently. It is a core idea in multi-agent exploration problems that dispatching the agents to different areas improves exploration efficiency. Although it has been studied for decades resulting in numerous methods, making an accurate and meaningful indoor map segmentation remains a challenge. It is a trade-off that generating many small segments can cover more accurate detail of the environment, but may lose the representation of some larger areas that are useful and relevant to the particular domain at hand.

Previous work on environment segmentation for exploration tasks has some limitations. These segmentation methods often aim to segment the layout of rooms, which are not designed for agent exploration problem, and under-segmentation and over-segmentation may occur. Undersegmentation may occur in some cases wherein some large segmented regions include smaller areas that should intuitively be identified as separate segments (e.g connected rooms with a corridor). In this case, the segments are over-generalized to the extent that they do not correspond to any meaningful region in the environment. The size of rooms sometimes are different in the real world, thus, this can easily reduce the efficiency of exploration, especially when one agent searches a big room but others stop at the end of the last segmented re-

gion. Oversegmentation happens when the environment is segmented to many small regions (e.g. Voronoi based [5, 6, 7]) which is unnecessary and reduces computation efficiency. Another issue is these segmentation methods can not perform ideally in real-time exploration.

Segmentation-based methods for multi-agent exploration problems can be divided into two parts: partition the environment and allocate tasks based on the segmented map. The segmentation method can generate segments from a map of the environment. For example, Wurm et al. employ the Voronoi algorithm to partition the environment from a map [7]. The segmentation clusters partition the environment to a representation which can be more appropriate for agents to build a cooperative strategy. Based on the segmented map, appropriate allocation tasks can be assigned to agents in order to improve the exploration efficiency. For example, Wurm et al. exploit Hungarian method to allocate the agents which can speed up the exploration process [7]. However, this approach is usually ineffective if agents can not obtain the whole environment information at the beginning. When considering the potential utility of visiting less explored areas, this approach may not be suitable for dynamic multi-agent exploration.

Although segmentation-based exploration method can handle many complex situations, some unpredictable situations require a more robust and adaptive approach to further improve the performance (e.g., temporary communication interruption). Specifically, when more agents are involved, the task becomes more challenging as they have to explore the unknown environment as efficient and fast as possible while avoiding conflicts or interferences among the agents that can reduce the efficiency. Learning-based approaches can be considered to solve the issues and improve exploration performance.

With deep learning methods, the problem can be divided into two parts: visual observation understanding and a reinforcement learning model for determining the target action. Visual observation usually exploits the features of CNN as the inputs to the training networks. For example, Zhu et al. introduce using CNN to obtain knowledge about the real-time explored environment [8] in a single agent system. The reinforcement learning model targets the robust and optimal actions for each agent. For example, a team of agents can learn how to explore the environment collaboratively even though one agent stops. However, previous works on agent exploration problem focus on a single agent system and are difficult to be used with a multi-agent system. The reason is that using a deep reinforcement learning approach for multi-agent exploration problem is challenging. First, visual observation is not easy for a machine to understand the task, and requires a long time to converge. Second, explo-

ration agents contain dynamic actions, and it is difficult to train the model with large exploration and action space.

1.3 Thesis Objective

In the first, this thesis focuses on developing a good representation of the environment which can provide effective and clear situation information in real-time to make exploration more efficient. This thesis first studies segmentation-based approaches, which are the foundation of multi-agent exploration strategy. A novel environment segmentation method is proposed in this thesis, which is used to have meaningful environment knowledge. Then, this new segmentation is used to build a multi-agent exploration model.

Another research objective of this thesis is using an end-to-end deep reinforcement learning approach to solve the multi-agent exploration problem. In this way, it can be used to handle the dynamic and complex environment.

1.4 Approach and Methodology

1.4.1 Segmentation-based Approach

To balance the high precision fine-grained partition and meaningful grouping of areas in the environment during the exploration, hierarchical-based segmentation can be considered as a new way to represent the environment situation. Hierarchical-based segmentation methods are flexible across various exploration environments, which contain rich information about the environment. Segments can be formed and constructed hierarchically within multiple layers of the neural network. This thesis aims to solve multi-agent exploration problems by a highly adaptive segmentation-based exploration approach. Building on this, a Hierarchical Adaptive Clustering (HAC) segmentation method is presented to produce a comprehensive map representation across hierarchical levels and consider regions' connectivities to generate the topological information for multi-agent exploration strategy. The segmentation method is based on a modified version of Adaptive Resonance Theory or ART neural network model [9], which is a bottom-up structure where small regions are partitioned at the first layer and the formed regions then group together to make larger common segments at the next layer.

HAC segmentation-based exploration approach is evaluated and compared with state-of-the-art methods that include frontier-based coordinated multi-agent map method,

as well as a group of segmentation based map exploration methods, including the Voronoi-based segmentation, distance transform segmentation, feature-based segmentation and morphological-based segmentation. The experiment results show that it can achieve state-of-the-art performance on a variety of indoor environments at various levels of complexity.

1.4.2 Learning-based Approach

Another research work in this thesis is to investigate how to obtain meaningful environment features via visual observation of each agent. Then, the work aims to use extracted features as the input to a reinforcement learning network to determine the target action for each agent.

Visual Observation Learning

Given an indoor environment map, we expect agents can explore the environment just based purely on situational observations. CNN can reflect the representation and extract the important features of the currently observed situation. For example, CNN is widely used in different computer vision tasks, like real-time object detection [10], face recognition [11] and image denoising [12]. A CNN model is used to identify the observed situation for each agent. Different from the single-agent problem, it is necessary to define the raw visual observations to multi-channel representations otherwise it may be impossible for each agent learning the meaningful situation knowledge. The observations are divided into four states: agents, known regions, unexplored areas, and obstacles. In this way, the reinforcement learning model can learn from the current situation without any previous knowledge.

Learning with Enhancement

Deep reinforcement learning is self-adaptive and suitable for solving large-scale complex problems, but using a deep reinforcement learning approach for multi-agent exploration problems is challenging. Proximal Policy Optimization (PPO) [13] can alternate between sampling data through interaction with the environment. It formulates the constraint as a penalty in the objective function and optimizes with stochastic gradient ascent, which is a good approach to get a balance between ease of implementation, sample complexity, and ease of tuning. Since its robustness, a PPO algorithm based deep reinforcement learning approach is developed to handle the multi-agent exploration problem with centralized learning and decentralized execution. Although PPO can provide an efficient exploration strategy with the visual inputs, it takes a

long time to converge, specifically, when the environment is complex. Therefore, some enhancements are applied to this work.

Curriculum Learning

To improve exploration, a curriculum learning model is added to the reinforcement learning model. Curriculum learning as an improvement function is used to deal with some complex and sparse scenarios. For example, it can make it possible to maximize learning efficiency [14]. With curriculum learning, it can provide a solid and natural learning process, and find the simple task policy to complete the hard tasks [15]. The learning model is conducted in a teacher-student fashion where the teacher controls the training sampling and student learn the knowledge from teacher's selection.

Intrinsic Reward

In our unknown environment exploration problem, typically, rewards are given by searching the unexplored regions. Thus, rewards become sparse when agents have explored the surrounding and fallen into a deadlock. To further improve the learning performance, the intrinsic reward can be a motivation to enable agents to explore more areas compared to the single extrinsic reward [16], which can be called as an intrinsic-driven exploration. The approach is based on how the difference of the situation between the two steps so that the agents are enabled to visit more unexplored areas.

This thesis uses Convolutional Neural Network (CNN) to understand the environment and adapt the Proximal Policy Optimization (PPO) to learn the CNN-based representation, then propose a CNN-based multi-agent centralized learning model for the multi-agent indoor exploration problem. To improve the exploration performance, the curriculum lessons are designed for deep reinforcement learning method. The proposed CNN-based Multi-agent Proximal Policy Optimization (CMAPPO) algorithm can learn a more efficient strategy for multiple agents to explore the environment than the conventional frontier-based method.

1.5 Contributions

In this thesis, a new representation and algorithms are designed for multi-agent exploration problem built on a segmentation-based approach and deep reinforcement learning-based approach. This thesis introduces a novel indoor map segmentation algorithm Hierarchical Adaptive Clustering (HAC) for multi-agent exploration that provides a hierarchical and adaptive way to segment the environment and propose a new coordinated exploration algorithm with the topological information from the segmentation method. Modifying the model of Adaptive Resonance Theory (ART)[9]

network, a hierarchical segmentation method is built to learn the indoor environment and split the environment to suitable individual tasks for agents.

Using observation of visualization, we want to learn the multi-agent exploration strategy by the visual observation inputs so we learn the environment situation information by CNN. A CNN-based Multi-agent Proximal Policy Optimization (CMAPPO) algorithm is developed to address the multi-agent coordinated indoor exploration problem. This algorithm combines CNN to process multi-channel visual inputs from the observed environment, curriculum-based learning for improving learning efficiency, and PPO algorithm for motivation based reinforcement learning.

The main pros and cons of each approach are summarized as:

	HAC	CMAPPO
Pros	1. Generates hierarchical and meaningful segments leading to a higher level of efficiency in exploring the environment. 2. Does not require training and is convenient when conducting. 3. Can be applied to any kinds of environment.	1. An end-to-end approach, is very convenient to be used. 2. Does not need to preprocess input observation images. 3. Provides efficient solutions when the environment is dynamic and unpredictable.
Cons	1. The performance can be further improved in some cases. 2. Can not be suitable to a dynamic situation.	1. Requires long time training.

Table 1.1: Pros and cons of proposed approaches.

The main contributions are summarized as:

- The HAC is formalized for indoor environment segmentation and can be applied to indoor multi-agent exploration problems.
- An end-to-end visual observation based multiple and parallel reinforcement learning algorithm for multi-agent exploration problem is proposed.
- A curriculum learning model is designed to address the complex exploration environment learning problem: additional lessons to improve the performance of learning efficiency, convergence and generalization of the multi-agent exploration.
- An intrinsic reward approach is introduced to improve the exploration ability in the unknown environment.

1.6 Outline of Thesis

In Chapter 2, the background of the multi-agent exploration problem and related approaches are introduced, especially the segmentation-based approaches, CNN and

deep reinforcement learning. Basic concepts and related works are described in this chapter.

In Chapter 3, the Hierarchical Adaptive Clustering (HAC) segmentation-based method for multi-agent exploration problem is introduced, which is a novel specific segmentation method designed for this problem. This hierarchical indoor environment segmentation method is designed with a modified ART model. Based on the segmentation results, it is enabled to build a multi-agent exploration strategy with topological information. The results show the proposed method is an efficient solution, and has a state-of-the-art performance on the benchmark of typical approaches.

In Chapter 4, this thesis focuses on solving the multi-agent exploration problem with the deep reinforcement learning method. With raw visual observation inputs, a CNN-based representation learning model is used to learn the situation map and the relationship among agents. Then, extracted features are used as the representation to Proximal Policy Optimization (PPO) network to build the collaborative exploration strategy. In this way, a CNN-based Multi-agent Proximal Policy Optimization (CMAPPO) algorithm is proposed. To improve exploration, some enhancement methods are added to the learning model. The enhancements include curriculum learning and intrinsic reward, which can encourage agents to visit more unexplored areas. CMAPPO shows a novel way to achieve success only with visual observation inputs and outperforms the state-of-the-art compared to the conventional frontier-based method.

Finally, in Chapter 5, this thesis concludes with discussions on promising future research directions. The models will be trained in realistic situations to solve challenging multi-agent exploration problem on various applications.

Chapter 2

Literature Review

In this chapter, the concepts of multi-agent exploration problem and several approaches that are related to this thesis are reviewed. First, this thesis discusses the multi-agent exploration problem and typical state-of-the-art approaches for designing efficient coordinated exploration systems on the environment map. Following the discussion, the main challenges and motivations in multi-agent exploration problem are described. Since some exploration situations can not be solved optimally with conventional approaches, the Convolutional Neural Network (CNN) and reinforcement learning approaches are introduced, wherein their connections to exploration problem are discussed.

In each subsequent section, this thesis reviews existing works and discuss their connections to this thesis. The flow of a literature review is presented in Fig. 2.1.

Figure 2.1: The overall taxonomy of Literature Review

2.1 Multi-agent Exploration Problem

The proper articles on multi-agent exploration have emerged at the beginning of the 1990s. In a multi-agent context, exploring an enclosed environment by a team of autonomous agents has remained a challenging problem. Different techniques and approaches have been used to solve the multi-agent exploration problem.

Yamauchi introduces the frontier-based method wherein each agent makes its own decision to allocate a target to visit though they share a global world representation [17]. Frontiers are defined as the areas on the boundary between the open environment and unexplored environment. During the exploration time, each agent can maintain its local map and make its decisions individually on which frontier pointers to navigate and also share the global map concurrently. The goal of this method is to search the unexplored areas as much as possible and obtain more information about the world.

Based on frontier-based method, Bautin et al. propose an improved method that each agent evaluates its relative rank among the other agents in terms of travel distance to each frontier [18]. This method is more effective because it explicitly allocates each agent to the lowest rank. Recently, the leading solutions to this problem are offered by segmentation-based exploration, perception-based exploration, and topological-based exploration.

2.1.1 Segmentation-based Exploration

The complexity of the space often makes exploration problem difficult. One way to reduce the complexity is to allocate different agents to segmented areas instead of frontier points [7]. Approaches based on the environment segmentation are widely used for many indoor environment exploration tasks. The segmentation-based approaches are more efficient than frontier-based approaches since they can reduce the spatial complexity of the environment by partitioning the environment to different individual areas and include more information about the connection between areas comparing to frontier points.

Voronoi diagram is a typical segmentation approach on the exploration problem, which specifies a set of obstacles points and segments the areas with equidistant from closest obstacles. Wurm et al. introduce a strategy to solve the coordinated multi-agent exploration problems, which is based on Voronoi segmentation to allocate the agents to different areas [7]. Voronoi segmentation is used to produce a set of one-dimensional connecting nodes, representing different areas (tasks) for agents. The nodes and edges of the Voronoi have a maximal distance of at least two points of the obstacle points in the map. It collapses leaf edges into the node points to prune to the

major skeleton and connect the critical points. However, with Voronoi segmentation, the potential issue is that trivial segmentation is generating during the exploration.

Segmentation-based exploration may not lead the shortest path but tends to perform maximal clearance of the unexplored space. Other similar works include Morphological operators, distance transform-based method and feature-based method [19]. Those works are mainly for specific targets, and their performances are more context or domain-dependent in terms of optimality compared with the Voronoi-based approach.

2.1.2 Topological-based Exploration

Topological-based approaches sometimes consist of segmentation and perception, which can be considered as an instance of the optimization problem. It is an NP-hard problem [20]. Burgard et al. [21] introduce this problem with the trade-off of exploration cost and area utility. This method further develops the classical frontier-based method. when exploring a new frontier point, the attributes of cost and utility will be updated. The approach solves the multi-agent exploration problem by adding explored and unexplored areas information via the trade-off of exploration cost and area utility, increasing the efficiency of the traditional frontier-based method.

For a complex exploration environment, the topological-based methods are a way to represent spatial information by establishing a topological map of cells. They regularly represent the exploration environment with hierarchical architectures. Zivkovic et al. utilize the graph partitioning to achieve a hierarchical path planning [22]. More specifically, they consider hierarchical exploration architecture as the navigation graph where can be simplified with graph cut criteria. Similar to hierarchical architecture, Brunskill et al. introduce the approach used with the incremental map, which is based on the online generation of topological maps and computes the graph partitioning with spectral clustering [23].

A more challenging problem than a pure multi-agent environment is regarded as human-agent collaborative problem [24]. Govindarajan et al. extend the topological-based method to human-agent exploration problem, which is likely to have noise and errors because of the dynamical process. With topological representation, when humans choose trajectories at their discretion, agents can adapt and choose complementary topological classes to maximize exploration efficiency. In this way, the topological-based approach is a proper way to make it robust towards a heterogeneous team of humans and agents.

2.1.3 Learning-based Exploration

Tackling multi-agent problems using deep reinforcement learning have been increasingly popular nowadays. Tampuu et al. [26] introduce multi-agent cooperation and competition problem using deep Q-Learning framework in a famous video game Pong. They can learn robust strategies to play against and cooperate with another adaptive agent to gain scores. However, the solution is only for two agents (or two adversarial agents) and may not be transferable to other situations. Compared to traditional optimization solutions, deep reinforcement learning provides a solution through adaptation and learning.

Few works have been done in applying deep reinforcement learning robot exploration problems. Recently, Zhu et al. [8] use CNN and LSTM for feature extraction and A3C to approximate the model. However, their method is only for a single agent with a known map and still requiring A* algorithm for planning the path to reach the nearest frontier point similar to the frontier-based method. To the best of our knowledge, the work presented in Chapter 4 is the first attempt to apply deep reinforcement learning for multi-agent exploration in an unknown environment with only visual observation as the inputs.

Perception-based Exploration

Recently, the development of deep learning inspired several perception-based approaches. Perception-based approaches are designed to be similar to human-being in relating the signs and things with their meaning. Thus, the goal of perception method is to identify the meanings of exploration environment and make strategies like human decision behaviours.

Beetz et al. introduce utilizing perception method to extract conceptual features based on natural language labelling during the exploration [27]. Zender et al. introduce a perception method based on environment labelling, and assign the different labels to agents [28]. The environment can be modelled as a graph connecting which classifies the space to different features, such as rooms, corridors or doorways. With the topological space partition, the agents can pass on the detected objects to the conceptual map, where they can satisfy some specific missions better. For example, the mission for agents exploration can be only searching for the corridors. Those perception-based approaches comprise high-level environment features with semantic mapping and model the human esthesia competency to relate the connections of visual observation, which aims to meet robust robot control and human-like conceptualization. Partitioning the environment to a semantic level not only is suitable for

reliable robot navigation but also is a representation to build a similar human conceptualization of spatial organization.

Compared to segmentation-based exploration methods, perception-based methods more depend on deep learning techniques, which requires a long time to learn the domain-dependent labelling and conceptual structure (as prior knowledge). In this way, although perception-based methods are similar to human decision-making logic, exploring the entire environment is still challenging.

2.2 Indoor Room Segmentation

Semantically segmenting indoor floor plans or maps into significant regions is an important task for many robotic related applications [19]. A meaningful space partition is significantly helpful and powerful to exploration problem that can be as the foundation to build a topological graph, and speed up the computation performance. The reasonable way to divide the environment is regarding the individual rooms. This may cause inefficient to multi-agent exploration problem, since the size of the rooms are different sometimes, where a large room needs more agents to explore.

Most agent exploration problem solved with indoor room segmentation methods are based on the following approaches: Voronoi graph based segmentation [5] [6] [7], morphological segmentation [29] [30], distance transform-based segmentation [31], feature-based segmentation [32] [33] [34], and the interpretation of architectural floor plans [35].

Meyer and Beucher introduce the morphological segmentation method [29] that each pixel is initially labelled as the accessible area. Morphological operator [30] is used to generate the segmentation in this approach, which sets up a fuzzy grid map to represent the certainty of cell occupancy by obstacles. Then, based on the grid map, it can split areas apart that only connected the narrow passages. With the morphological erosion operator, the walls of the map are grown iteratively by one pixel. Then using the connected component analysis to verify whether previously connected accessible areas have become separated by the erosion. Repeat this procedure until all accessible cells have been removed from the map.

Distance Transform-Based Segmentation is another commonly used algorithm [31]. This method is based on place labels at different locations that are located by agents. The distance transform is used to group map cells to rooms and connect them with the closest label. In this way, it can generate local maximum to move all the pixels to a common local maximum where is belong to one room. The local maximum of the distance transformation always lies in the center of a room and is smaller

than those inside a large room with narrow corridors or doors.

Voronoi graph-based segmentation is one of the major algorithms for segmentation [5]. The nodes and edges of Voronoi graphs have a maximal distance of at least two points of a finite set of obstacle points in the map. A Voronoi graph is generalized on the grid map and collapsing leave edges into the node points to prune to the major skeleton. Then connecting the critical points with two closest obstacle points to obtain critical lines. After completing all the critical lines, small Voronoi cells can be merged into larger regions under the merge laws.

Feature-based Segmentation is described in [32]. The basic data used for feature computation is collected by simulated rays. From this scan, a set of simple geometric features has been computed. The feature vector is classified by an AdaBoost classifier [36] into room labels. Finally, all neighbouring points with the same label are merged.

Other methods developed in recent year is based on the deep learning approach. Dodge et al. mention that fully convolutional networks are used to learn to detect the wall border [37]. The dimensions of map segments can be estimated by optical character recognition. Although those learning-based methods are more flexible than traditional methods and can prevent the influence from the noise, it is not very popular to be applied to real robotic applications. Since it requires a long time to train and is hard to transform the model into different scenarios.

2.3 Representation Learning and Decision Making in Dynamic Environments

2.3.1 Convolutional Neural Network

Convolutional Neural Network (CNN) is a classic approach to extract the features form images [38]. CNN is very helpful and is widely used for image understanding tasks. There have recently been significant advances based on CNN, such as face recognition [39], image classification [40], playing difficult games (e.g., Atari) [41]. In addition to image understanding tasks, CNN is also a popular model for speech recognition [42] and natural language understanding [43]. Here, we focus on the related problems of image understanding and visual observation based exploration.

LeCun et al. introduce the classic model of Convolutional Neural Network to learn complex, high-dimensional, nonlinear mappings from big data and make them suitable for image recognition or speech recognition tasks [38]. CNN changes the traditional way of pattern recognition, which is a subversion of hand-designed feature extraction. The traditional ways require human to label the features, then extract

relevant information and delete the irrelevant variabilities. With the convolutional architecture for recognition, each unit of a layer can receive a small neighbourhood feature of the previous layer. Those inner features can be combined by the higher layers, therefore, some degree of shift and distortion invariance is ensured and keep the topological information of the input. Inspired by biology, CNN provides a novel architecture for the machine to learn the raw image inputs. In this thesis, a CNN model is applied to extract the features of exploration situations.

The input plane receives images that are normalized and centred. A typical architecture is with a few layers, and each unit in the layer can receive the local neighbourhood features from the previous layer. With the receptive fields, it can extract essential features that will be combined by the higher layers. Then, the elementary feature detectors are used to have identical weight vectors. The outputs of such processes are called feature map. Different types of units have different feature map to compute different types of features, so that multiple features can be extracted at one location if the convolutional layer has many feature maps. CNN can be seen as synthesizing their own feature extractor, because all the weights can be learned with backpropagation.

Building on CNN, Krizhevsky et al. introduce the deep convolutional neural network for large high-resolution image data sets classification [40]. The method is effective and with high accuracy because they use non-saturating neurons, an efficient GPU implementation and "dropout" regularization. More recently, many papers have used CNN for complex image-related task understanding. Karpathy et al. extend the connectivity of CNN to classify the large-scale videos from YouTube [44]. He et al. introduce the advanced model of CNN called Mask R-CNN for object segmentation [45]. With a simple and flexible model, advanced CNN becomes a baseline for image understanding task.

Apart from normal image tasks, CNN is also used for robotics visual navigation task. Giusti et al. introduce a supervised learning method for robots navigation task, which is based on visual inputs [46]. However, it requires a complicated and time-consuming training period and only can be applied to the same environment.

2.3.2 Deep Reinforcement Learning

Increasingly, deep reinforcement learning has been a robust approach to solve challenging robotics problems. It has recently widely used and been increasing advances for many difficult tasks.

Gu et al. mention using off-policy training of deep Q-functions for autonomous robots to learn large repertoires of behavioural skills in physical world [47]. Zhu

et al. introduce a similar work of robotic control that proposes an actor-critic model with target-driven based policy update for visual navigation [48]. Deep reinforcement learning is also a popular approach to learn how to play hard games, and reach a high playing level. Mnih et al. introduce how to successfully learn control policies from high-dimensional sensory input with reinforcement learning in Atari game [49]. Mnih et al. introduce an end-to-end deep Q-network approach to learn successful policies in 50 different games [41]. Additionally, deep reinforcement learning is a novel way in the financial market. Deng et al. introduce using deep learning and reinforcement learning combined approach for financial asset trading [50]. The deep learning is for dynamic market condition features extraction, and reinforcement learning is for trading decision-making learning that accumulates the rewards in an unknown environment.

For the multi-agent exploration problem, reinforcement learning with a deep learning approach has become a new method. Tampuu et al. introduce multi-agent cooperation and competition problem using deep Q-Learning framework in a famous video game Pong [26]. They perform how to produce robust strategies play against and cooperate with another adaptive agent to get higher scores, instead of a hard-wired algorithm. However, the solution is only used in two agents (or two adversarial teams) situation and difficult to transfer to other situations. Compared to traditional optimization solutions, deep reinforcement learning can provide new thoughts to solve this multi-agent exploration problem.

Though people have developed many approaches to solving multi-agent exploration, there is no polynomial-time solution to obtain the optimal strategy on this problem. Through deep reinforcement learning approach, it is possible that multi-agent team can learn some models to perform coordinated exploration in the complex environment. Several well-performing models such as Deep Q Network [41] [51], Actor-Critic [52] [53], Deep Deterministic Policy Gradient [54], Proximal Policy Optimization [13] are good for a single agent and multi-agent environment.

Markov Decision Process

In single agent reinforcement learning, the environment of the agent is described by a Markov decision process [55]. The policy learning in the discrete-time Markov decision process (MDP) is defined by a 6-tuple $(S, A, r, p_0, p, \gamma)$, where $S \in \mathbb{R}^n$ is a n dimensional state space, $A \in \mathbb{R}^m$ is a m dimensional action space, $r(s_t, a_t) \in \mathbb{R}$ is the reward function, p_0 is the initial state distribution, $p(s_{t+1}|s_t, a_t)$ is the state transition distribution and $\gamma \in [0, 1)$ is the unnormalized discount factor. The objective of it to learn a stochastic policy, a probability distribution over actions $\pi : S \rightarrow \Delta A$, which is

parameterized by θ , to maximize the expected cumulative reward:

$$J(\theta) = \mathbb{E}_{s \sim p_\pi, a \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right] \quad (2.1)$$

where $p_\pi(s) = \sum_{t=1}^{\infty} \gamma^{t-1} \mathbb{P}(s_t = s)$ is the discounted state visitation distribution. The *value* function is expected total discounted reward over θ .

$$V^\pi(s_t) = \mathbb{E}_\pi \left[\sum_{t' \geq t} \gamma^{t'-t} r(s_{t'}, a_{t'}) | s_t, \pi \right] \quad (2.2)$$

In order to find a policy to maximize the return, we define state-action function

$$Q^\pi(s_t, a_t) = \mathbb{E}_\pi \left[\sum_{t' \geq t} \gamma^{t'-t} r(s_{t'}, a_{t'}) | s_t, a_t, \pi \right] \quad (2.3)$$

Policy Gradient Methods

Reinforcement learning problem can be defined as an optimization of an expectation when optimizing the stochastic policies. The formalism of stochastic computation process helps us to develop the intricate models that can involve a combination of stochastic and deterministic operations. Policy Gradient method is for reducing the variance, which is computing an estimator of the policy gradient and plugging it into a stochastic gradient ascent [13]. Many gradient estimator functions are proposed in recent years, and a proper choice of it is vital to reduce the variance. The most commonly used function is from

$$\hat{g} = \hat{\mathbb{E}}_t [\nabla_\theta \log \pi_\theta(a_t | s_t) \hat{A}_t] \quad (2.4)$$

where π_θ is a stochastic policy and $\hat{A}_t$ is an estimator of the advantage function at timestep t . The expectation $\hat{\mathbb{E}}_t$ is the empirical average, which is to alternate between sampling and optimization. With the implementations by constructing a policy gradient estimator based objective function, the estimator $\hat{g}$ is obtained by differentiating the objective.

$$L^{PG}(\theta) = \hat{\mathbb{E}}_t [\log \pi_\theta(a_t | s_t) \hat{A}_t] \quad (2.5)$$

The loss function is defined by $L^{PG}(\theta)$, which is an appearing method with multiple steps of optimization and the same trajectory in the experiments. However, the classic loss function is always with large policy updates, which leads to a worse learning result.

Proximal Policy Optimization

Proximal Policy Optimization (PPO) is a policy gradient method for reinforcement learning. The architecture of PPO is actually an Actor-Critic model, which means the PPO method can produce continuous actions and can be evaluated the current policy rules by the actor [13]. PPO is developed from the foundation of previous models and overcome the issues of convergence, which is sometimes better than existing state-of-the-art approaches and is much simpler to be implemented and tuned.

Policy Gradient is a helpful method in reinforcement learning but also challenging. For example, it is sensitive to the update step size. If the size is too small, the progress will be very slow. If the size is too large, the performance will be influenced by the noise or catastrophically drop. Another common issue with Policy Gradient is that the learning is inefficient since it has poor sampling ability [13].

The main idea of PPO is to constrain the size of a policy update that can eliminate these flaws. Although, methods like Trust Region Policy Optimization [56] and [57] provide a similar way to deal with those flaws. With the objective of learning from only images, PPO is better to solve the problems where visual inputs are significant. Since it is compatible with algorithms that share parameters between a policy and value function or auxiliary losses [13].

The novel point of PPO is that it can optimize a "surrogate" objective function using stochastic gradient ascent through the sampling data. Different from other methods, those sampling data can be reused for a stable and efficient update. One way to formulate the PPO is changing the constraint to the penalty:

$$\max_{\theta} \hat{\mathbb{E}} \left[\frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \hat{A}_t(s_t, a_t) \right] - \beta \hat{\mathbb{E}} [KL[\pi_{\theta_{old}}(\cdot|s_t), \pi_{\theta}(\cdot|s_t)]] \quad (2.6)$$

where $\hat{\mathbb{E}}$ denotes the empirical expectation over timesteps, $\hat{A}_t$ is the generalized advantage function, β controls the weight of the penalty. A Kullback–Leibler (KL) divergence here is for the conjugate gradient descent, which makes it stable training and maintains the reasonable update.

Another way is with clipped objective. It can be formulated by:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t [\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t)] \quad (2.7)$$

where $\hat{\mathbb{E}}$ denotes the empirical expectation over timesteps, θ is the policy parameter, r_t is the ratio of the probability under the new and old policies, $\hat{A}_t$ is the estimated advantage at time t , ϵ is the hyperparameter. With the function of the CLIP, we can evaluate a new policy with samples collected from the older one that controls the

difference between two policies.

2.3.3 Curriculum Learning

Bengio et al. introduce a boosting algorithm called curriculum learning, which trains with a uniform distribution over the training set [15]. The idea of curriculum learning is generated from human study strategies, which is learning from easy tasks to difficult tasks. The human can learn better when the examples are not randomly presented and the practices are gradually getting harder [15]. The corresponding training strategies at step λ is formulated by:

$$Q_\lambda(z) \propto W_\lambda(z)P(z) \quad \forall z \quad (2.8)$$

where z is a random variable representing an example for the learner, $P(z)$ is the target training distribution, $W_\lambda(z)$ is the weight applied to z at step λ in the curriculum sequence. The λ is starting from 0 and ending at 1. Such that $\int Q_\lambda(z)dz = 1$, then we have

$$Q_1(z) = P(z) \quad \forall z \quad (2.9)$$

Each curriculum can be seen as a sequence of training criteria. During the training, we associate weights on training examples that the weights can be reweighted. More specifically, the weights illustrate the sampling of which difficulty level of training examples. At the end of the sequence, the training process is conducted on the target training set or the target training distribution.

2.3.4 Intrinsic-driven Learning

As the learning signal, the primary rewards are provided to agents after every decision making. In the unknown environment exploration problem, typically, rewards are given by searching the unexplored regions. Thus, rewards become sparse when agents have explored the surrounding and fallen into a deadlock. To avoid such a situation and improve the exploration performance, it can be possible to provide the rewards not only from the environment or mission itself but also from the intrinsic.

The idea of the intrinsic reward is introduced by Pathak et al., they proposed that intrinsic reward can be a signal to enable the agents to explore the environment more than it might have and help agents learn to get more extrinsic rewards in the later [16]. It demonstrates and imitates an infant who does not pursue any specific target to learn about the world, but rather playing and exploring for the novelty of the experience. The approach is to reward agents based on how surprised the exploration environment

is, which drives agents to learn more strategies to explore the unexplored environment and find more about the surprising states. Specifically, agents tend to discover the new area (unexplored regions) and extrinsic reward.

To formulate the approach, the agent in state s_t is with the executing action a_t sampled which is from its current policy π and ends up in state s_{t+1} . With the intrinsic reward, the networks are trained by a forward and an inverse model. In the inverse model, the policy π is trained to optimize the sum of the extrinsic reward and the intrinsic reward and encodes the states s_t, s_{t+1} into features $\phi(s_t), \phi(s_{t+1})$ that are trained to predict action a_t . In the forward model, $\phi(s_t)$ and a_t are the inputs to predict the feature representation $\hat{\phi}(s_{t+1})$ of s_{t+1} . The prediction error in the feature space is used as the intrinsic reward, which means bigger difference enables the agent to obtain the bigger intrinsic reward.

The overall optimization problem that is solved for intrinsic-driven learning can be represented as

$$\min_{\theta_\pi, \theta_I, \theta_F} [-\lambda \mathbb{E}_{\pi(s_t; \theta_\pi)} [\Sigma_t r_t] + (1 - \beta)L_I + \beta L_F], \quad (2.10)$$

where θ_π is optimized to maximize the expected sum of rewards, θ_I is the likelihood estimation that is maximum by minimizing L_I amounts under a multinomial distribution, L_I is the loss function that measures the discrepancy between the predicted and actual actions, θ_F are optimized by minimizing the loss function L_F , $0 \leq \beta \leq 1$ is used to weigh the inverse model loss against the forward model loss, $\lambda > 0$ is used to weigh the importance of the policy gradient loss against the importance of learning the intrinsic reward signal. The loss function L_F is defined as:

$$L_F(\phi(s_t), \hat{\phi}(s_{t+1})) = \frac{1}{2} \left\| \hat{\phi}(s_{t+1}) - \phi(s_{t+1}) \right\|_2^2. \quad (2.11)$$

The intrinsic reward signal is computed as $\eta/2 \left\| \hat{\phi}(s_{t+1}) - \phi(s_{t+1}) \right\|_2^2$, where η is a scaling factor.

Chapter 3

Hierarchical Adaptive Clustering for Indoor Room Segmentation¹

Semantically partitioning indoor maps, such as the interior of a office containing several rooms into meaningful regions, can be used by a mobile agent or a team of agents to perform an exploration task in a search and rescue mission [58], planing efficient floor cleaning strategies [59], or to assist the mobile agents teleoperator [60]. Those applications are involved in our daily life and play an important role, such as cleaning [3], search and rescue, reconnaissance [2], mowing and autonomous deployment (e.g., WSN deployment [61]).

Autonomous indoor map segmentation has been studied for decades. Numerous approaches to construct suitable map partitions have been developed [19], including morphological segmentation [30, 62], distance transform-based partition [31], feature-based segmentation [32, 33, 34, 63, 64], and Voronoi graph-based segmentation [5, 6, 7], among which the Voronoi method is the most known and commonly used [19].

Despite the extensive studies, autonomously segmenting regions from an indoor map in an accurate, efficient, but meaningful manner remains a great challenge, especially, partition the meaningful regions for agents exploration problem. For example, *undersegmentation* may occur in some cases using the methods above wherein some large segmented regions include smaller areas that should intuitively be identified as separate segments (e.g connected rooms with a corridor). In this case, the segments are over-generalized to the extent that they do not correspond to any meaningful region in the environment. On the other hand, *oversegmentation* may also occur in some other more accurate approaches (e.g Voronoi based [5, 6, 7]) wherein too many

---

¹The work is still under review as Luo T, Chen Z, Subagdja B, Tan A H. Real-time Hierarchical Map Segmentation for Coordinating Multi-Robot Exploration[J]. IEEE Access, 2019.

small regions are segmented unnecessarily and cause the exploration redundancy. In this case, the segmentation process may also miss some large regions which can be intuitively meaningful (e.g a long corridor instead of a series of segments). The excessive number of segments produced can also make further processing in the system computationally less efficient. Since segmentation underlies exploration strategies, irrelevant and imprecise partitions of regions can greatly affect the performance in achieving the domain tasks.

Different from existing segmentation-based exploration methods that applied static indoor room segmentation methods and tackle the trade-off in segmentation, a novel segmentation method for multi-agent exploration problem is introduced. Based on a modified version of Adaptive Resonance Theory or ART neural network model [9], segments can be formed and constructed hierarchically within multiple layers of the neural network. The segmentation process is conducted in a bottom-up fashion in which small areas are partitioned at the first level and the areas formed are then group together to make larger common segments at the next level. Since there is a chance that multiple agents as a team could interfere with each other and have an inefficient exploration strategy when exploring an unknown terrain or enclosed environment. A proper segmentation method can improve the exploration performance which is the foundation of building the efficient allocation strategy where exploration interference or conflicts are minimized. Thus, the exploration with meaningful regions information may overcome the above issues and can maximally cover all the regions in a short time.

Additionally, in an unknown environment, targets can only be determined after the environment representation partially constructed. The typical approach for unknown environment exploration is to find and select a number of frontier points (the locations between known and unknown areas) and allocate the nearest agent or the one with least-cost to explore [4]. With the development of segmentation-based exploration, the exploration have been more efficient. It allows agents to explore separate partitions according to the map segmentation basis. Wurm et al. [7] introduce the Voronoi segmentation-based multi-agent exploration method, which utilizes a pre-segmented map done by Voronoi method and allocates agents with the Hungarian method that achieving faster map exploration compared to frontier-based method. Since Voronoi method does not target for exploration segmentation, sometimes unreasonable allocation still happens.

A novel hierarchical adaptive clustering (HAC) segmentation method to partition the exploration area of the map and assign agents to explore different regions, solving the problem of multi-agent collaborative exploration in an unknown or partially

known environment. This chapter aims to build a suitable segmentation model to increase the efficiency of multi-agent exploration and benefit many robotic applications or products. The proposed segmentation method is for coordinated multiple agents scanning a partially known environment and with the area segmentation and the unknown environment with real-time exploration. The real-time segmentation can partition the explored area based on the HAC segmentation so that agents can be assigned into suitable regions to perform an efficient exploration.

Building on the HAC segmentation based exploration approach, this chapter produces a detailed environment representation across multiple levels, which can reduce the search overlap to improve efficiency. Furthermore, the segments' connectivity is utilized as the topological information to improve the team exploration performance, which is better than state-of-the-art algorithms. Solving agents exploration problem in this way is a fundamental method, since after Yamauchi [17] and Burgard et al. [21], most exploration approaches are based on frontier or simple space connectivity approach, but the performance has not been improved significantly. However, HAC segmentation partitions the environment into different segments (e.g. rooms, corridors, corners) with importance degree which can provide meaningful representation for agents. It can not only partition a whole map but also work when the map is initially available only partially or totally unknown. In this case, a team of agents can be assigned to the segments based on a cost-utility measure. The cost measures the distance from the selected agent to a target location, and the utility measures the potential of a target location to be explored by any agent. By applying the cost and utility function, the cost of visiting the targeted location and the potential discovery of new areas are all be considered.

This thesis evaluates and compares HAC segmentation method both quantitatively and qualitatively with state of arts that include morphological, distance-transform, feature-based, and Voronoi graph-based segmentation as mentioned above using thirty maps obtained from some real-world floor plans[19, 37, 65]. Meanwhile, the HAC segmentation-based multi-agent exploration method is evaluated on various complexity level exploration environment and is compared with the frontier-based method, and the Voronoi-based exploration methods.

In this chapter, the experimental results show that the HAC segmentation is better than the others in terms of accuracy and relevance, and HAC-based multi-agent exploration method provides a promising approach to collaborate with the agents' team and search the environment with an optimal solution. The new algorithm indicates that it can process multi-agent indoor room exploration with the exceptional performance but low-redundancy, in various settings.

3.1 HAC Methodology and Model

To solve the problem in coordinated autonomous agents and indoor map segmentation, we propose the HAC segmentation-based exploration method. The essential point of the proposed segmentation method is that close environment is clustered together into individual regions in a bottom-up fashion. In the following section, we introduce the architecture of HAC and coordinated multi-agent exploration with topological information, then formulate the algorithms and their performance analysis.

3.1.1 HAC Architecture

HAC architecture consists of two different modified Adaptive Resonance Theory (ART) [9] neural networks. ART is a neural network model that categorizes or grows clusters from inputs in a self-organized manner [9]. The particular ART employed in this chapter is a modified version that clusters inputs in euclidean space and finds the center points (centroids) of the clusters. The two levels of clustering were applied in this architecture: the lower-level clustering that groups points sampled randomly from the map or directly from input sensors; and the upper-level clustering which groups the lower level cluster centers into larger segments.

3.1.2 Low-level ART network

Figure 3.1 shows the low-level ART neural network. A point position $p = (x, y)$ in an $M \times N$ two-dimensional map can be used as an input to the ART network by normalizing p into input vector $\mathbf{p} = (p_1, p_2)$ where $p_1 = \frac{x}{M}$ and $p_2 = \frac{y}{N}$ so that $p_1, p_2 \in [0, 1]$. Let $\mathbf{P}$ be the set of all vectors as normalized points corresponding to the ones in the map ($\mathbf{p} \in \mathbf{P}$) and let $\mathcal{P} \subseteq \mathbf{P}$ be the set of the vectors corresponding to empty areas on the map (non-empty space like obstacles or walls are excluded).

Following the particular ART-2A-E [66], a cluster node j in F_2 layer is activated based on a bottom-up euclidean *choice function*.

$$T_j = 1 - \sqrt{\frac{|\mathbf{p} - \mathbf{w}_j|}{d}} \quad (3.1)$$

where $|\mathbf{x}| \equiv \sum_{i=1}^d x_i$ if $\mathbf{x} = (x_1, \dots, x_d)$ and d is the dimension of the input vector (in this case $d = 2$). $\mathbf{w}_j$ is a weight vector associating the input $\mathbf{p}$ in F_1 with the node j in F_2 .

A node J is selected to be the category of $\mathbf{p}$ in F_2 if it satisfies the *resonance condition* such that

$$T_J = \max\{T_j \mid \text{for all } j \text{ in } F_2, T_j \geq \rho\} \quad (3.2)$$

Figure 3.1: The HAC architecture consisting of two modified ART neural networks: low-level and upper-level ART. Each ART consists of F₁ layer (input) and F₂ layer (category) (F₁' and F₂').

where $\rho \in [0, 1]$ is the *vigilance* parameter. If no F₂ node satisfies the *resonance* (for every j , $T_j < \rho$ always holds), a new *uncommitted* node is recruited in F₂ to represent the new input. Thus, the categories are growing as the network is receiving novel inputs.

A particular weight update operation is applied whenever a node J is selected so that $\mathbf{w}_J$ is updated as follows,

$$\mathbf{w}_J^{(\text{new})} \leftarrow \frac{m^J \cdot \mathbf{w}_J + \mathbf{p}}{m^J + 1}, \quad (3.3)$$

where m^J is the number of input points categorized as J so far. This contrasts with the conventional update function used in other ART networks [9, 66]. Here, the weight $\mathbf{w}_J$ represents the center (centroid) of the cluster rather than the node's matching template.

In this approach, a point p is considered to be in a category j if its normalized position ($\mathbf{p}$) has the closest Euclidean distance towards j (located at $\mathbf{w}_j$) but cannot be greater than ρ as the *resonance* criteria. From empty areas on the map, n points are randomly sampled to make the set of sampled points $P = \{p_i\}_{i=1}^n$. Each $p_i \in P$ is presented consecutively as an input to be clustered or grouped as a node J . Thus, each node j in F₂ represents a segment in the map with a center point $\hat{p}^j = (\hat{x}^j, \hat{y}^j)$ as the inverse normalization of its weight vector $\mathbf{w}_j$.

3.1.3 Upper-level ART network

The center point $\hat{p}^j$ of each low-level segment node j in F_2 is then presented one at a time as the input to the F_1' layer of the upper-level ART to generate the upper-level segments or clusters in F_2' (Figure 3.1). This upper-level ART works similarly as the low-level one. However, it has also been modified so that the *resonance* criteria to select the matching category include an additional check if no obstruction (e.g walls, obstacles) exists between the input point with the center point of the category being selected. This extra check is used to avoid a lower-level segment nearby but located in a separate room to be included in a larger cluster in the upper-level network. The upper-level choice function is as follows,

$$T_j^{(\text{Upper})} = \begin{cases} 0 & \text{if there exists } \mathbf{p}' \in \mathcal{P}^c \text{ so that} \\ & \|\mathbf{p}' - \mathbf{w}_j\|_2 + \|\mathbf{p} - \mathbf{p}'\|_2 = \|\mathbf{p} - \mathbf{w}_j\|_2 \\ 1 - \|\mathbf{p} - \mathbf{w}_j\|_2 & \text{otherwise.} \end{cases} \quad (3.4)$$

In this case, $T_j = 0$ if there is a point $\mathbf{p}'$ in the set of occupied points $\mathcal{P}^c$ (the complement of empty areas $\mathcal{P}$) located along the straight line between $\mathbf{p}$ and $\mathbf{w}_j$. Based on the selected cluster and the input, its central point can be obtained by applying the centroid-based weight update function.

Before the segmentation starts with processing the input in the lower level ART, a pre-processing stage is conducted using Shi-Tomasi approach to detect doors and door features within the area to be partitioned [67, 68].

The clustering is then conducted within a known or mapped area $\mathcal{A}$ in the environment. If the environment has been known or mapped prior to the segmentation, $\mathcal{A}$ is considered covering the entire area of the environment. **Algorithm 1** shows the steps included in the segmentation process by the hierarchical clustering method.

After the upper-level segmentation, a graph representation structure G can be formed by connecting the center points in $\mathcal{S}_{\mathcal{U}}$ to each other. Each segment center corresponds to a node (vertex) with the edge connecting to another associated with their path distance, where the path is calculated by A* path planning algorithm [69]. To simplify the matter, edges larger (longer) than threshold δ can be excluded. This process is performed in the last step of **Algorithm 1**. When the map of the environment is known in advance, the segmentation process and the construction of G are illustrated in Figure 3.3 (a) to (d).

3.2 HAC Mechanism

This thesis introduces the basic architecture of HAC approach. Specifically, the entire hierarchical segmentation procedures are summarized as follows.

Step 1. Map feature engineering. Resize the map to proper size (can be ignored). With the Shi-Tomasi Corner Detector [67, 68] and door features, HAC is used to detect and identify small opening areas and narrow corners to be excluded from empty areas ($\mathcal{P}$) or to be considered as occupied regions or obstructions.

Step 2. Randomly sample n points from all empty areas (except walls, furniture, obstacles) on the map as set P .

Step 3. Generate the low-level segments $\mathcal{S}_{\mathcal{L}}$ by presenting every point p_i in P as the input to the low-level ART and consecutively cluster them into category nodes in the ART upper-layer based on low-level *vigilance* ρ as the closeness criteria. $\mathcal{S}_{\mathcal{L}}$ corresponds to these upper nodes.

Step 4. For every segment j in $\mathcal{S}_{\mathcal{L}}$, its center point $\hat{p}^j$ is used as the input to upper-level ART in which the same process of *resonance* search is applied to make larger segments based on upper-level *vigilance* ρ^u as the closeness criteria. Here, a constraint is added on the matching: there should be no walls or doors in between $\hat{p}^j$ and corresponding upper-level cluster center. By presenting $\hat{p}^j$ of every segment j in $\mathcal{S}_{\mathcal{L}}$, the set of upper-level segments $\mathcal{S}_{\mathcal{U}}$ can be generated.

Step 5. For every segment k in $\mathcal{S}_{\mathcal{U}}$, total points $P_{\mathcal{U}}^k$ of the segment k include all points in low-level segments that belong to segment k so that $P_{\mathcal{U}}^k = \bigcup_{j=1}^r \hat{P}^j$ in which segments 1 to r in low-level ART are included in k . The center points of low-level segments and the total points for each upper-level segment are illustrated in Figure 3.3 (b) and (c) respectively.

3.2.1 Merge fragmentary segments Criteria

The internal clusters in a common area such as large rooms or corridors sometimes can still be considered to be segmented separately. To increase the robustness, a graph G is constructed over the map where each graph node or vertex correspond to a segment center with edges connecting to each segment. Each edge in G is associated with a value indicating the Euclidean distance [70] between the connected nodes.

With the edge-cutting process on graph G , the merging and separating of segments can be conducted. A fully-connected G can be separated if the edge value is larger or longer than a threshold δ . The edge can also be removed if there is an obstruction (e.g., walls or doors) in between the nodes that are connected. Thus, we can cut or remove the edge and obtain the completed topological information of exploration

Figure 3.2: The map segmentation procedure by HAC. (a) Randomly sampled 7000 points on the empty space of the map. (b) Groups of low-level clusters wherein each group represents a segmented region. (c) The output of upper-level clustering in which each blue point represents the segment center. A graph is constructed by connecting the segment centers. (d) The final segmentation output after the fragmented segments are merged.

environment.

The pruned graph may then be decomposed into several sub-graphs. Each segment within a sub-graph can be merged to form a new larger segment cluster wherein the new segment center is based on its geometric centroid. We introduce the detailed process of graph-based segment merging and partition in Figure 3.3 (d).

The overall algorithm of HAC is illustrated in **Algorithm 1**.

3.3 HAC Segmentation-based Exploration

The proposed HAC segmentation method is designed as the foundation for multi-agent indoor environment exploration. To test the performance, this thesis evaluates it with hierarchical centralized multi-agent exploration system. A hierarchical centralized system is comprising a commander and a group of the individual agents wherein the commander can have the global information to decide on the exploration strategy

---

Algorithm 1: Hierarchical Clustering Segmentation

Input : area $\mathcal{A}$ in the map
Output: The list of segments with the corresponding centroids
 1 Pre-process $\mathcal{A}$ for door and door features with Shi-Tomasi approach
 2 Randomly drop n points $P = \{(x_i, y_i)\}_{i=1}^n$ on $\mathcal{A}$
 3 **for** each p_i in P **do**
 4   Normalize p_i into input vector $\mathbf{p}$ with Equation 3.1 to present to lower level ART in F1
 5   Perform clustering with lower-level ART to generate the set of cluster center points $\mathcal{S}_{\mathcal{L}}$ in F2 according to Equation 3.2
 6 **end**
 7 **for** each $\hat{p}^j$ in $\mathcal{S}_{\mathcal{L}}$ **do**
 8   put (normalized) $\hat{p}^j$ as the input to upper level ART in F1'
 9   Perform clustering with upper-level ART (excluding points with obstruction in between the cluster center) to generate the set of cluster center points $\mathcal{S}_{\mathcal{U}}$ in F2', following the Equation 3.4
 10 **end**
 11 Construct graph G based on $\mathcal{S}_{\mathcal{U}}$

---

for the agent team, acting as a central brain, and the agent team can conduct the local area exploration task. Specifically, in the commander level, commander allocates the agents to different regions, while in the agent level, the agents explore the environment within its assigned regions. The overall goal of this problem is to complete the environment with minimum travel distance. The Cost, Utility and Weight functions are defined for allocating each agent with the optimal segments selection and balancing the trade-offs. The Cost, Utility and Weight functions are introduced as follows.

Cost

We use Cost to measure the distance between the current agent location to possible target segments, which is a proper way to influence the exploration strategy. During the exploration time, each agent may have many potential targets, as the interesting regions. The cost is used for avoiding the travelling redundancy. If the commander sends the agent to the unexplored environment directly, the agents are more likely to perform repeated exploration somewhere, which will decrease the performance quality. For example, the agents are assigned to the same targets if the strategy is only based on the nearest distance. Moreover, there are many unexpected situations in exploration we need consider to address multi-agent exploration problem with interaction during the exploration. For example, one single agent may be broken during the exploration which is a dynamic situation. Therefore, making exploration decision with global situation considerations are particularly relevant to achieve successful multi-agent exploration.

One important issue in the multi-agent exploration problem is how to assign the

proper target regions to individual agents. Given the fact that multi-agent exploration is a complex system, it is necessary to balance the trade-offs to address the problem reasonably and avoid several agents moving to the same areas. To find the solution and solve the issues mentioned before, we consider the distance or cost as one of exploration strategy influence factors. The cost of current segment cluster is computed by the optimal path from agents' location to the centroid of the segments. Typically, the optimal path can be computed by the A* search algorithm [69], an efficient way to find the path and its distance on a 2D grid map. With segmentation, the A*-based path distance from agent k to segment i is denoted as $d_{A^*,i}^k$. The location of i is then determined by the segment center point. Let $C_{I,K}$ be the set of possible cost of an agent to visit a segment i so that,

$$C_{t_i,r_k} = \left\{ d_{A^*,i}^k \mid i \in \{1, \dots, n\} \right\}. \quad (3.5)$$

Utility

It is difficult to predict each individual agent's decision at each step during the exploration. Besides, the decisions are easily affected by the surrounding environment or agents [7]. Thus, another essential consideration of multi-agent exploring problem is how to send the agents to different segments. The overall goal is to reduce the possibility of exploring the same targets.

Specifically, given a possible situation of that one agent has moved to a particular region, other agents should not be assigned to this region where may not be explored completely yet. We consider utility as a measurement to influence the allocation. Intuitively, if a cluster is near to an agent, the utility for other agents to explore this cluster should be lower, hence, other agents will have less probability to search this cluster, which increases the efficiency of completing the whole map. Furthermore, when an agent moves to a cluster, the adjacent utilities should decrease to prevent other agents to come to this area, performing the collaborative strategy. Let U_i be the utility of segment i . U_i can be defined as the minimum cost of an agent within a range surrounding i to visit the segment as follows,

$$U_i \leftarrow \min_i \frac{c_{i,k}}{\beta}, \forall i, k, c_{i,k} < \beta, \quad (3.6)$$

where $c_{i,k} \in C_{I,K}$ and $c_{i,k}$ is the cost for agent k to travel to segment i . β is the maximum limit of the cost for an agent to be considered in the utility.

Given the utilities of the segments and the costs to visit them by the agents, the

allocation for agent k can be obtained as follows,

$$s_k \leftarrow \arg \max_i (U_i - \gamma \cdot c_{i,k} + \sigma \cdot w_i), \quad (3.7)$$

where γ is the cost significance parameter, $c_{i,k}$ is the cost of agent k to visit i , σ is the weight significance parameter, and w_i the importance weight of segment i . The weight w_i can be used to determine if particular segments are more important than the others. The weight of segment i will be introduced in the next subsection.

Weight

During the multi-agent map exploration, intuitively, the agents should be sent to border regions and explore them firstly comparing to a centre region, especially, when they have the same cost and utility. Since the central area is more likely to be explored by other agents which causes the potential redundancy. With the consideration of weights, the agents can perform search strategy more collaboratively since agents obtain more details about the environment than before. The weight parameter is considered for each cluster node, which can represent global targets structure for the agents. The weights are calculated within the Google power iteration idea [71]. Through the power iteration, the sequence of eigenvectors $\{x_k\}_{k \in \mathbb{N}_0}$ converges to the importance score $(e_1, \dots, e_n)$ of each node. Furthermore, the weight of each node is defined as the following equation:

$$w_i = 1 - \frac{e_i}{\max_i(e_i)}, \quad (3.8)$$

where e_i is the importance score of segment i , indicating the centrality of segment i . Intuitively, after the map is segmented, the border region should be explored pre-emptively comparing to a centre regions, as centre regions are easier to be visited by any agents in subsequent exploration. Therefore, the map into graph is segmented G by **Algorithm 1**, as applying the Google Power Iteration [71] on graph G , which directly results in an importance score $(e_1, \dots, e_n)$ for each node of the graph G .

3.3.1 Exploration in the Partially Known Environment

Partially known environment is that the layout map or the floor plan of the environment is known or available as the prior knowledge, but the detail condition of a place or room in the environment (e.g furniture, rubles, people, victims) is unknown. One of the important reasons this thesis concerns this kind of situation is that the floor plan of the shopping malls or office rooms can be obtained when environment exploration

task is required such as search and rescue in a disaster or a dangerous situation, and the indoor map can be used as the partial information that makes it helpful to obtain the basis of the environment at first and exploring the environment faster.

The goal of exploration is to search the whole environment in a minimal steps that requires us to search the unexplored area with a reasonable multi-agent dispatching strategy. With the partially information of the environment, the segmentation graph can be generated to represent the environment firstly. The environment uses occupancy grid maps where each cell is corresponding to the unexplored area, explored area and the obstacle. This thesis focuses on unexplored segments during the exploration. If the agents are directly assigned to the unexplored segments or areas, it may get the optimal solution at current step, but it will cause repeated path in the further exploration. For example, two agents are assigned to the same segment, but one reaches the region firstly then another agent should go to the other segments, which may go back to the starting region. The main target is to avoid several agents move to the same segments or areas. To achieve the target mission and select the appropriate target segments for individual agents, the trade-off is considered among the cost of moving to a segment, the utility of that segment and the weight of the segment correlation to the global environment. The determination procedure is conduct by an iterative approach that target segments are selected by balance the trade-off. Then the utilities of all segments are recomputed. The Cost and Weight function make it possible to generate a more optimal strategy for the team of agents and perform the coordinated exploration. This process is repeated for the remaining agents to cover unexplored segments under the balance of target distance and target importance.

Each step an agent moves to its designated area, the cost and utility corresponding to the agent and its target change according to (3.5) and (3.6). Similarly, the best target segment to allocate may change as well according to (3.7). When the agent arrives at its allocated target, it scans its surroundings and the information from the sensor readings updates the current map of the environment in the corresponding area surrounding the agent location. This uncovered area resulting from the scan readings is considered as a new explored area $\mathcal{A}$ and subjected to the segmentation process.

In this paper, it is assumed that the segmentation and task allocation are conducted in every cycle of the agent execution. The cycles can be either synchronously or asynchronously performed among the agents. The segmentation process is initiated everytime an area $\mathcal{A}$ is unfolded by an agent. **Algorithm 2** shows the overall steps taken by the multi-agent exploration system.

Figure 3.3 illustrates an example of how the HAC segmentation-based method works and how it provides meaningful space information for the multi-agent explo-

Figure 3.3: The procedure of hierarchical clustering. (a) illustrates the procedure of sampling points. (b) shows the *lower layer* clusters while (c) shows *upper layer* clusters. (d) demonstrates the topological information output of final clusters. Figure (e) to (h) shows the procedure of incremental segmentation during exploration: (e) sample points on the newly explored area. (f) and (g) Hierarchical clustering, which are similar to (b)(c). (h) Compute new segment centres and merge them with the segment graph.

Algorithm 2: Exploration Task Allocation based on Cost-Utility Measure

---

```

1 perform segmentation with Algorithm 1 if known area  $\mathcal{A}$  is initially available
2 while exploration is unfinished do
3   for each agent  $k$  do
4     if new area  $\mathcal{A}$  is uncovered by agent  $k$  then
5       perform segmentation in  $\mathcal{A}$  with Algorithm 1
6     end
7     for each segment  $i$  do
8       update costs  $C_{I,K}$  and utility  $U_i$ , following Equation 3.5 and 3.6
9     end
10    allocate agent  $k$  to segment  $s_k$ , following 3.7
11    if no unallocated  $s_k$  is available (all have been allocated to other agents) then
12      find the set of frontier points  $V$  within the segment where agent  $k$  is currently in
13      allocate a frontier  $v \in V$  to agent  $k$  to visit
14    end
15    update position of agent  $k$  to allocate to the location
16  end
17 end
    
```

---

ration problem. It is essential to mention that this thesis generates the environment topological information at the end of the HAC segmentation method, which offers space connectivity that helps us identify which area may be explored frequently. Regarding this, the algorithm can increase the exploration efficiency.

3.3.2 Exploration in the Unknown Environment

Unknown environment means no floor plan or layout map are known (including the detail conditions). This is a normal situation that happens in our daily life and is a fundamental issue in many related applications. To choose the optimal targets for individual agents, we consider to balance the trade-off of different aspects. The exploration decisions are jointly influenced by the cost to the possible targets, the utility and the weights of targets. Additionally, multi-agent exploration in the unknown environment can be considered in two situations: $n_s \geq n_a$ and $n_s < n_r$, where n_r represents the number of agents and n_s represents the number of segments. The solution is designed dynamically to achieve a good performance.

This thesis considers two exploration situations in the unknown environment. One situation is when the number of unexplored segments is greater than the number of agents. After the agent reaches the target segment, it will perform the frontier-based exploration method within its area and the agents will be assigned to the next segment cluster when it complete the searching task within the current area.

Another exploration situation is the number of unexplored segments is less than the number of agents, exceptionally, one consideration is regarding other agents' actions as an important influence factor when the individual agents make decisions. This thesis first allocates the agents to occupy all the unexplored segments, while the rest agents will perform frontier-based exploration. Whenever new clusters are spotted, we assign unallocated agents to these clusters. In this way, the dynamic strategies lead the exploration becoming more robust and suitable for a lot of environment difference.

With the **Algorithm 2**, the task allocation process can be conducted continually and incrementally as each agent moves and scans its surrounding. It can also be performed in parallel for each unfolding area in the map or separately in decentralized manner when multiple agents are involved. The incremental segmentation process can be depicted as in Figure 3.3 (e) to (h).

The conditional statement in line 11 Algorithm 2 checks if no segment is available to be allocated to the agent since all segments have been allocated to others. This condition may occur when the number of identified segments is lesser than the number of agents. This also means that two or more agents may occupy the same segment. In that case, a frontier-based allocation is applied to the agent. Line 12 Algorithm 2 finds all frontiers (t) within the segment or area where the agent is currently resided in. A frontier t is then selected to assign to the agent to visit. The selection criteria for the frontier-based allocation can also be based on a cost-utility measure. Similar to the approach described in [21], the detailed selection criteria of how the agent i

---

Algorithm 3: Real-time Hierarchical Clustering

- 1 Agent k explores an unsegmented area $\mathcal{A}$
  - 2 Randomly sample m points $\{P_i, i = 1, \dots, m\}$ within $\mathcal{A}$
  - 3 $\hat{p}^j \leftarrow$ calculate first level clustering centroid of $\{p\}_{i=1}^m$
  - 4 Merge $\hat{p}^j$ with nearest second level cluster $\mathcal{S}_{\mathcal{U}}$ which satisfies criteria $\mathcal{C}_2$
  - 5 If no $\{\mathcal{S}_{\mathcal{U}}\}$ satisfies $\mathcal{C}_2$, create a $\mathcal{S}_{U+1}$ and set centroid as $\hat{p}^j$
-

chooses a target t can be described as:

$$(i, t) = \arg \max_{(i', t')} (U_{i'} - \beta \cdot V_{i'}^{t'}) \quad (3.9)$$

where $U_{i'}$ is the computed utility of target t , $V_{i'}^{t'}$ is the distance to target t . β is the relative importance of utility and distance, which generally is set to 1. To determine appropriate targets for agents, it should be iteratively computed to have the best overall evaluation $U_t - \beta \cdot V_t^i$.

Real-time Clustering

When exploring the unknown environment, the real-time clustering algorithm is performed as the segmentation method. As mentioned before, proper segmentation for exploration environment is crucial to complete the searching task efficiently. Specifically, the incremental segmentation for the scanned area can produce the clustering of the unexplored environment and can be a target signal for individual agents. The team of agents may not preform economic action selection strategies if only with frontier points. With the unreasonable targets allocation, agents may easily explore very close frontier points, which causes waste of resources.

To perform a better segmentation for incremental searching and provide a better exploration strategy, this thesis integrates the target segments selection method, the HAC segmentation-based method, and the frontier-based local exploration method into a complete multi-agent exploration method. This method can be used in both partially known environment and unknown environment, which is the real-time clustering based exploration. As summarized in **Algorithm 3**, the essential goal of this method is to lead individual agents perform real-time clustering, especially during exploring the unknown environment. Furthermore, this method is tested and evaluated in simulation, which will be explained in detail in the **Section 3.4**. The real-time clustering based exploration outperforms than the frontier-based exploration.

3.4 Experiments and Discussion

3.4.1 Exploration Settings

In this section, the thesis first evaluates and compares the HAC segmentation method. The proposed model is compared with four other state-of-the-art methods: Morphological, Voronoi, distance-based and feature-based methods, as presented in [19].

The indoor room segmentation experiments are conducted based on 30 different publicly available floor plans, where 20 maps are from the IPA room segmentation dataset [19], and 10 maps are from the R-FP dataset [37]. The floor plans from IPA room segmentation dataset, collected by Bormann et al. [19], contains 20 office floor plans, where the maps are taken from the Radish Repository [72] and the Oscar Martinez Mozos dataset. The R-FP dataset contains the scanned maps generated from real Japanese floor plans [37]. The maps are pre-processed to be grid-based, where each cell in the grid represents either an empty space or obstacle.

The experiments use the following parameter settings for HAC:

- Number of sampled points n : 7000.
- Low-level vigilance ρ : 0.95.
- Upper-level vigilance ρ^u : 0.8.

The implementation parameters of the Morphological, Voronoi, distance-based and feature-based methods are based on their default configurations as mentioned in Bormann et al. [19].

Performance Metric:

The performance measurement is designed with *Precision*, *Recall* and *F-measures* which can quantitatively evaluate the performance. The evaluation is checking against the human-labeled ground truth. The ground truth is based on the pre-labeled maps from IPA data set [19] and manually labeled the segments from R-FP dataset [37].

$Recall = \frac{tp}{tp+fp}$ denotes the containment of the generated segments in all areas of the ground truth, where tp is the total *true positive* or the coinciding generated segments areas with the ground truth, fp is the total *false positive* or the areas of the generated segments not coinciding with the ground truth. $Precision = \frac{tp}{tp+fn}$ denotes the containment of the human-labeled segments as the ground truth in all areas of the generated segments, where fn is the total *false negative* or the areas of the human-labeled segments (ground truth) not coinciding with the areas of the generated segments. $F-measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$ is the *harmonic mean* of precision and recall.

Additionally, to further test proposed method, this thesis evaluates HAC segmentation method in multi-agent exploration problem with other state-of-the-art map exploration methods. The cascaded models are tested on ten complex office maps from ROS indoor room map collection dataset [19], which are widely used in similar research. Each map contains more than 20 separate rooms that is suitable to test the collaborative ability of the various multi-agent map exploration algorithms. The maps are captured in a grid-based (pixel level) representation, stored in the form of a $n \times m$ matrix, where n is the height and m is the length. Each cell in the grid is assigned to one of the following states:

- **Explored:** This cell is explored by at least one agent.
- **Unexplored:** This cell haven't been explored by any agents.
- **Obstacle:** This cell is occupied by obstacles that agents cannot pass.

To make the results comparable among different maps, this thesis first defines the width of the map to be $m = 200$ cells, and the height of the map is determined according to the aspect ratio. In addition, each agent is equipped with a 360° sensor with a radius of $r = 15$ cells. During the exploration, each agent can explore the local area within a radius r in one step.

The experiments test more than 100 parameters combination, and adopt the best possible parameters for the ten experiment maps. The HAC segmentation-based exploration experiments are conducted with the following parameter settings:

Multi-agent map exploraiton

- Cost parameter γ : 0.01
- weight parameter σ : 0.35
- Utility update parameter β : 0.6

The vigilance ρ in the upper layer is appropriate for the sensor range used by the agents, so that when an agent is at the centre of a segment, it is able to explore the entire segment. Each agent can update the segmentation information and possible target parameters to the commander as the global information during the experiments.

Performance Metric:

The performance measurement is designed with average travel distance of each agent within a single trial to assess the exploration. Explicitly, the average of travel distance among the agents as the indicator for the assessing the performance after

	Precision	Recall	F-measure
Morphological	0.96±0.04	0.84±0.13	0.90
Distance Transform	0.96±0.04	0.89±0.10	0.93
Voronoi	0.94±0.04	0.94±0.07	0.94
Feature-based	0.91±0.10	0.81±0.21	0.85
HAC	0.97±0.03	0.95±0.04	0.96

Table 3.1: Average Precision and Recall ($\pm$ Standard Deviation) of the Segmentation Methods on 30 Maps

completing the whole map exploration. As commonly done in the exploration literature [7, 21], the average travel distance gain is the additional evaluation indicator. Those evaluations are chosen since the experiments are conducted entirely in simulation. Time based measures can be suitable for physical agents or actual robots in which all the operations are conducted in real-time. Real-time measure could be biased by processing overhead in the simulation steps which could be run in irregular intervals. To avoid the software processing speed influence and other confusing factors, this thesis considers a fair way. The travel distance gain is defined as

$$Gain = \frac{d_b - d_p}{d_b} \times 100\% \quad (3.10)$$

where d_b represent the travel distance of the baseline method, and d_p represent the travel distance by the proposed method.

3.4.2 Evaluation in HAC Segmentation

Table 3.1 shows the quantitative comparison of the results with respect to the ground truth. The results show that the segmentation accuracy of HSOAS outperforms the other state-of-the-art methods. The accuracy is even higher than the Voronoi method which is known to be the most robust and accurate method compared with the others.

By decomposing the map into small areas and hierarchically organize them to form semantic segments, the HAC can maintain fast and accurate segmentation. The edge cut techniques in HAC provide larger and meaningful chunks of segmented regions and yet maintain a certain level of precision. This method can strike a balance between *under-segmentation* and *over-segmentation*.

Fig 3.4 shows some sample outputs produced by HAC and other state-of-the-art methods. The examples show that the morphological, distance transform and feature-based segmentation methods tend to lump some regions into larger segments. On the other hand, the Voronoi method, which is known to be more robust and accurate than

Figure 3.4: Examples of comparing HAC segmentation with four state-of-the-art segmentation methods. From left to right column: (1) the ground truth; (2) Morphological segmentation; (3) Distance-based segmentation; (4) Voronoi graph-based segmentation; (5) Feature-based room segmentation; (6) HAC segmentation.

the others, cannot segment a larger chunk of region such as corridors. Although it has employed a pruning mechanism to simplify the generated topological graph [5], it is only limited to group segments with two adjacent neighbours. In that case, it may fail to recognize a group of segments when there are many doors or opening rooms implying that most segments along a corridor may have more than two adjacent neighbours.

In contrast, HAC can produce larger segments through the edge cutting process wherein the grouping is only constrained by the distance between two central points of the segments and possible obstructions between them. Therefore, segments can still be grouped into a larger chunk of regions even though some of them have more than two adjacent neighbours. This can be observed in Figure 3.4 , which shows that with HAC, large segments can be formed to represent meaningful and intuitive geometrical shape.

3.4.3 Evaluation in HAC segmentation-based Exploration

Evaluation in Partially Known Environment

The experiments are designed for evaluating and demonstrating the ability of the HAC segmentation-based multi-agent task allocation approach within the the partially known map. Agents can share the segmentation information, updating utility and the targets weight. In this experiment, the team sizes of agents are from 1 to 10 with four different starting points (top left/right and bottom left/right), which are used to balance the variance in each size experiment. Combining those features in the experiment, the overall system architecture is assumed with partially knowing the whole map at the beginning, while agents can update the explored areas to the system. As mentioned before, the environment map is segmented by HAC with topological connectivity information firstly before assigning agents to perform coordinated exploration.

To illustrate how the HAC segmentation-based exploration approach performs in the partially known environment, this thesis compares with the frontier-based [21], Voronoi based method [7], Morphological method, Distance Transform method and Feature-based method [19]. The detail of Voronoi based exploration method is described as follows.

Voronoi Based Method

Voronoi based method provides appropriate and reasonable approach to solve the multi-agent exploration problem [7]. It distributes the agents to structure regions which are partitioned into segments. The generation of Voronoi segments can be described that the nodes and edges of Voronoi graphs have a maximal distance of at least two points of the obstacle points in the map, and the Voronoi graph collapses leave edges into the node points to prune to the major skeleton and connect the critical points. The goal of Voronoi diagram function is to segment unexplored environment to some meaningful partitions. However, it is can not be used in an incremental exploration environment, lacking flexibility sometimes. The Voronoi based method is combined with frontier points and region segments as its target selection signal to perform the exploration task. The method is described as below:

1. Determine the segments S of a map, and a set of frontier points of each segment.
2. Compute for each agent the distance C for reaching the each S .
3. Assign each agent using the Hungarian Method [73].

Voronoi based exploration method shows that the multi-agent exploration problem can be solved by partition the unexplored environment to small regions. Since

it lacks the ability of doing incremental segmentation and merge small segments to a reasonable one, it generate many small piece segments during the exploration that may not provide a fast and efficient performance. Bormann et al. [19] introduce four state-of-the-art room segmentation methods, which are popular for agents exploration. Additionally, the thesis compares the proposed method with [19] to evaluate the performance. Other map segmentation methods are Morphological, Distance Transform and Feature-based method. As for the exploration method, it is the same as the Voronoi based exploration. The results show that proposed approaches are superior to others since they have a better understanding of explored and unexplored environment during the exploration and approaches are hierarchical to perform a flexible scenes for agents.

Morphological Exploration Method: Meyer and Beucher [29] introduce the Morphological Segmentation method, which works on a grid map where the pixels are initially labeled accessible only at mapped accessible areas. With the morphological erosion operator, the walls of the map are grown iteratively by one pixel. Then, connected component analysis is used to verify whether connected accessible areas have become separated. Repeat those steps until all the accessible cells have been removed.

Distance Transform Exploration Method: The transform is the distance of an accessible pixel to the closest border pixel. The local maximum always lies in the center of a room, and it is smaller than those inside a large room with corridors or doors. Segmentation completes until the room found center is labeled uniquely and extended into remaining unlabeled space via wavefront propagation.

Feature-based Exploration Method: Martinez Mozos et al. [32] mention that the data for feature computation is collected by simulated rays. From the scan, a set of simple geometric features has been computed. The feature vector is classified by an AdaBoost classifier [36] into room labels.

Results and Discussion

The thesis implements and evaluates the approaches on a series of simulation experiments to perceive the interaction effect between meaningful environment partitions and multi-agent decision strategies. For generating the simulation results, the Python Matplotlib Animation Toolkit is used. All the experiments assumed that the agents share the global map which is produced by the sensor readings of all agents. The experiments have been designed to evaluate if the HAC segmentation-based exploration method manages to significantly decrease the average travel distance compared to the state-of-the-art methods.

To demonstrate the effectiveness of HAC segmentation-based exploration, Figure

3.5 is used to summarize the average gain of four methods on each map to reach the goal. It presents that allocating agents based on a comprehensive understanding of the map through the topological representation is significantly more efficient than the frontier based method. Notably, when the number of agents increases, the HAC segmentation-based exploration can be even more efficient. To find the influence of weight factor, this thesis compares the HAC segmentation-based exploration with and without the nodes' weight. It is worth to mention that one of the novelties is that the space connectivity information is considered during the exploration. In Figure 3.5, with consideration of nodes' weight (i.e., allocate agents to border nodes in precedence), it outperforms in the small number of agents. When the number of agents increases, the agents can separate to explore both centre and border nodes. However, it reaches the maximally optimal plan when the number of agents increases. Therefore, it does not significantly improve the performance with a large number of agents. Conversely, for small agent team size, considering the weights of nodes can achieve improved efficiency in map exploration.

Figure 3.5: Average travel distance gain for HAC segmentation-based multi-agent map exploration under partially known map with nodes' weight and without nodes' weight, compared with the frontier based multi-agent exploration method.

Fig 3.6 shows the contrast to Morphological, Feature, Voronoi and Distance Transform based exploration method, the HAC segmentation-based method consistently outperforms than other state-of-the-art techniques. It is important to notice that the best result reduces up to 36% average travel distance of those methods. For those methods in comparison, although they dispatch the tasks based on the map segmentation, their solutions are not optimal, leading to poorer performance.

Figure 3.6: Average travel distance gain for HAC segmentation-based exploration method, compared with the Morphological, Feature, Voronoi, Distance based exploration methods.

Evaluation in Unknown Environment

The experiments are designed to assess the advantage of the proposed real-time hierarchical clustering based multi-agent exploration method within the unknown environment. During the exploration, the agents share the global segmentation information and make decision dynamically. The team size and starting points are the same as the setting of the experiments in partially known environment. This thesis assumes that the map or layout is unknown when the agents begin exploration task.

This thesis compares the real-time hierarchical clustering based exploration with the frontier-based exploration method to evaluate the performance. The frontier-based exploration is widely used in many applications that is an essential method to solve this problem and as the baseline approach.

Results and Discussion

The implementation setting in the unknown environment is similar to the partially known environment. The experiments are designed to evaluate whether the proposed real-time hierarchical clustering based exploration method can conduct with exceptional average travel distance and average travel distance gain compared to the state-of-the-art methods.

Figure 3.7 illustrates the results on average travel distance gain showing that real-time HAC segmentation-based method can help agents plan better exploration strategies even the agents do not have any information about the environment at the beginning. Since based on real-time clustering, the agent team can obtain better un-

understandings of the environment, like having knowledge of rooms. Therefore, the proposed real-time method can achieve better efficiency than the frontier-based exploration by assigning agents to separate partitions in the unknown environment.

Figure 3.7: Average travel distance gain for real-time HAC segmentation-based exploration under the unknown map, compared with the frontier-based multi-agent exploration.

Furthermore, to test the performance of real-time HAC segmentation-based method under different environment, the experiment maps are categorized into two types:

- maps with the simple environment (less than 30 rooms),
- maps with the complex environment (more than 30 rooms).

As the number of rooms grows, the map becomes more complex, and better coordination strategy is needed to overcome the complexity. The performance shows the comparison between the real-time hierarchical clustering based multi-agent exploration and frontier-based method under different types of the map. The results are shown in Figure 3.8. As the environment complexity and the number of agents increasing, it shows that the proposed real-time exploration method becomes more efficient compared to the frontier-based method due to the highly coordinated agent task allocation and the better representation of the environment.

3.5 Conclusion

In this chapter, we have presented the HAC segmentation method for semantic map segmentation and further develop this method to multi-agent exploration problem.

Figure 3.8: Average travel distance gain for real-time clustering based exploration compared with frontier based exploration method on maps with different complexities.

The basic principle of the HAC segmentation method is to hierarchically cluster points and small regions of the map into larger segments. Using a modified ART neural network architecture, HAC is designed to handle partitions and regions in Euclidean space where segments are formed into multiple levels of clusters in a bottom-up fashion. This topological structure of clusters allows fast and yet precise segmentation. The bottom-up formation of segments inherently maintains the central points or centroids of the clusters which are useful to generate a graph structure that links different segments in a meaningful way. The graph can also be useful for further processing, such as robotic path planning.

Additionally, this thesis defines the HAC segmentation-based exploration method that partitions the environment to meaningful regions and provides a hybrid decision-making strategy with cost, utility and topological weights. Having an appropriate environment segments and target selection, the efficiency and allocation problems can be solved during the multi-agent exploration. The inappropriate segmentation or simple location points to represent exploration environment would lead multi-agent to make the incorrect allocation decision and overlap path. The typical methods such as frontier-based method or Voronoi segmentation-based method work as the state-of-the-art methods to multi-agent exploration. However, they are possible to miss some areas which can result in an overlap with the travelling path and increase the whole distance. The HAC segmentation method provides a good environment representation of rooms, corridors and the topological relationship to help solving those issues and reducing the overall exploration distance.

Based on the experiments, it is shown that HAC segmentation method outperforms other state-of-art-methods in segmentation, both quantitatively and qualitatively, in terms of accuracy and precision. Besides, the expanding application of this method provides a remarkably efficient multi-agent indoor exploration method compared to the typical state-of-the-art methods.

Despite the promising results, the HAC segmentation method has only been tested in deterministic grid-based maps, wherein each cell can only be either an obstacle or empty space. However, in real-world cases, the environment can be non-deterministic as well wherein a cell may express the uncertainty or probability of the existence of empty space or obstruction. The segmentation problem on a non-deterministic map is a challenge and requires further investigation.

Chapter 4

End-to-end Deep Reinforcement Learning for Multi-agent Collaborative Exploration ¹

This chapter is about multi-agent exploration problem with deep reinforcement learning. While the segmentation-based method can solve this problem with an efficient way, the deep learning-based approach can consistently outperform the optimization-based methods in the unknown environment and provide strategies to handle unpredictable or dynamic situations. This chapter emphasizes learning-based approaches for multi-agent exploration problem.

Agent exploration as a fundamental function in the robotics area can be applied to many important tasks. One fundamental challenge in robotics domain is exploring an environment when no a priori knowledge or only partial information about the environment is available. The robot or agent needs to map the environment while performing its domain tasks like surveillance [74], search and rescue tasks [75], cleaning [76], or collecting objects [77].

The common method of robots exploration is by taking a frontier point which is located in the boundary between a known area and an unknown region as the target location to visit. This point is selected from other frontiers as revealed whenever the robots observe the environment. Some ad-hoc strategies or heuristics can be applied to optimize the selection or allocation of the targets to the agents [17] [7]. However, when multiple robots are involved, the task becomes more challenging as they have to explore the unknown environment as efficient and fast as possible while avoiding

---

¹The work is accepted as Chen Z, Subagdja B, Tan A H. End-to-end Deep Reinforcement Learning for Multi-agent Collaborative Exploration[C]. 2019 IEEE International Conference on Agents (ICA). IEEE, 2019.

conflicts or interferences among the agents that can reduce the efficiency.

This chapter presents a learning approach for multi-agent exploration task rather than a fixed strategy to allocate the agents to their targets. This thesis develops a deep reinforcement learning model that takes raw visual observations of the environment and represents the state of affair in terms of multi-channel maps of agents, known regions, unexplored areas, and obstacles. The state representation is used as the input to a CNN (Convolutional Neural Network) model to process and extract important features of the environment relevant to the exploration task in hand. The extracted features are then used as the input to a PPO (Proximal Policy Optimization) network to determine the target action for every agent to take.

To further improve the learning performance, a curriculum model and intrinsic rewards are applied to direct the exploration. In this way, a new better strategy can be discovered allowing efficient exploration applicable to different environments and conditions.

The approach is evaluated by comparing it with the state-of-the-art Frontier-based multi-agent exploration algorithm [17] [7]. The experimental results show that our learning approach can discover better strategies for the multi-agent indoor exploration problem. The model exhibit reasonable generalization ability across teams of agents of different sizes and in various environments.

For the remaining parts of this chapter, the formulation is firstly presented, and training process of CMAPPO is introduced in Chapter 4.2. An enhancement with curriculum method and intrinsic reward method are introduced in Chapter 4.3. Lastly, the experiments and a conclusion are subsequently presented in Chapter 4.4 and Chapter 4.5.

The main contribution of this chapter research is summarized as follows.

- A CNN-based observation understanding method for multi-agent exploration problem is proposed that can transform the purely visual image to highly structured state representation.
- A multiple and parallel PPO-based reinforcement learning model is proposed to address the coordinated multi-agent exploration problem.
- The learning curriculum is designed to improve the learning efficiency of multi-agent exploration into a complex environment.
- An intrinsic reward approach is introduced to solve the unknown environment behaviour problems and improve the exploration ability.

Figure 4.1: The architecture of the CMAPPO. In decomposed representation, the information of obstacles is incrementally increased during the unknown environment exploration.

4.1 CMAPPO Model

In this section, the proposed CMAPPO reinforcement learning method is introduced to solve the coordinated multi-agent exploration. The thesis presents it in two parts, one is the CNN-based observation model for understanding the exploration environment situation, the other is learning-based approach for performing multi-agent exploration.

4.1.1 Representation Learning

To represent the current observed situation and identify the important features, a CNN model is used. The CNN reflects the exploration situation and features in the environment that are relevant to the agents' overall exploration tasks.

In order to enable the agents to learn the coordinated exploration strategy, the features discovered from the CNN model are used as the input state to a reinforcement learner. In this case, the exploration strategy can be learned from scratch without any prior knowledge through two stages of learning: feature extraction by CNN and actions learning by a reinforcement learner.

The architecture of CMAPPO can be depicted in Fig. 4.1. The CNN part of the architecture contains six layers - three convolutional and three fully-connected networks to extract features from the raw visual input of the environment.

The first convolutional layer filters the 84x84x3 input pattern where the observation is decomposed to agents channel, obstacles channel, unexplored regions channel and explored regions channel. Each of the input channels are connected with 16 filters with kernel size of 8x8 and a stride of 4. The second convolutional layer takes the input from the output of the first convolutional layer and produces the output with 32 filters, each with kernel size of 4x4 and a stride of 2. The third convolutional layer

is same as the second one and takes the output of the second layer as the input. Each hidden layer is followed by the ReLUs non-linearity with 256 neurons.

ReLU Nonlinearity

Following Nair and Hinton, we construct the network with Rectified Linear Units (ReLUs) neurons nonlinearity. Nair and Hinton introduce this significant function and present it is able to train faster than their equivalents with tanh units [78]. ReLUs can prevent the vanishing gradient problem which is a main issue in deep neural network.

The way to model a neuron's output f as ReLUs function of its input x is with

$$f(x) = \max(0, x). \quad (4.1)$$

4.1.2 Proximal Policy Optimization Based Learning System

Proximal Policy Optimization (PPO) [13] is a new family of policy gradient methods for deep reinforcement learning, which alternate between sampling data through interaction with the environment. It formulates the constraint as a penalty in the objective function and optimizes it with stochastic gradient ascent.

This thesis extends the PPO to handle the multi-agent issues by developing centralized learning and decentralized execution.

In particular, the agents share and contribute to the same PPO network. This thesis considers that the system involves n agents and each agent i can obtain its state $s_{i,t}$ at time t . With the given state s for agent i , the thesis computes the value and updates the policy with experiences collected by all agents simultaneously. Each agent utilizes the same policy to generate an exploration strategy.

The training process alternates between actions by executing the policy in parallel and updating the policy with the all selected actions. The constraints and advantage estimation are used to optimize the KL divergence, which can be described as follows,

$$L^{KL PEN}(\theta) = \mathbb{E}[\frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} A_t(s_t, a_t) - \beta KL[\pi_{\theta_{old}}(\cdot|s_t), \pi_{\theta}(\cdot|s_t)]], \quad (4.2)$$

where π_{θ} is the policy, a is the selected action, A_t is the generalized advantage function, and β is a parameter used for next policy update. In each iteration, the selected actions are used to build the surrogate loss $L^{KL PEN}(\theta)$ that is optimized with the Adam optimizer [79] under the KL divergence constraint for K epochs.

Adam is an optimizer method that computes adaptive learning rates for each parameter. The advantage of Adam method is that Adam can perform an exponentially

decaying average of past gradients as well as keep flat minima in the error surface. The decaying averages of past and past squared gradients m_t , v_t is represented as following:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t, \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2. \end{aligned} \quad (4.3)$$

where β_1 and β_2 are hyperparameters. After initialization, it will generate the biases. Thus, these biases are counteracted with

$$\begin{aligned} \hat{m}_t &= \frac{m_t}{1 - \beta_1^t}, \\ \hat{v}_t &= \frac{v_t}{1 - \beta_2^t}. \end{aligned} \quad (4.4)$$

Overall, the Adam update rule can be written as:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \varepsilon} \hat{m}_t, \quad (4.5)$$

where g is the gradient, η is the learning rate. This thesis uses $\eta = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 1e - 08$ in the experiments.

In PPO, the value function V is used to estimate the advantage $\hat{A}_i^t$ that does not look beyond time-step T . After running the policy for T timesteps, the selected actions are used for an update. The loss L^V for value V_ϕ is built with squared-error loss and Adam optimizer, which can be defined as follows,

$$L^V(\phi) = - \sum_{i=1}^N \sum_{t=1}^{T_i} \left(\sum_{t' > t} \gamma^{t'-t} r_i^{t'} - V_\phi(s_i^t) \right)^2 \quad (4.6)$$

A truncated version of generalized advantage estimation is used, which can be formulated as follows

$$\begin{aligned} \hat{A}_i^t &= \sum_{l=0}^{T_i} (\gamma \lambda)^{l-t+1} \delta_i^{l-1}, \\ \delta_i^t &= r_i^t + \gamma V(s_i^{t+1}) - V(s_i^t), \end{aligned} \quad (4.7)$$

where i is the agent, t specifies the time index in $[0, T]$, γ is the discount parameter for future rewards, and λ is a parameter which provides a bias-variance trade off. Policy and value function are updated independently and their parameters are not shared during the training. The multiple PPO can be scaled to the multi-agent system in a decentralized fashion where each agent performs the action individually. The com-

pleted procedure is presented in the Algorithm 4. In the next section, the thesis will introduce some enhancement strategies to improve learning efficiency in exploring an unknown environment.

4.1.3 Problem Setup

Reinforcement learning for the multi-agent coordinated exploration problem can be formulated as a Markov Decision Process (MDP). Formally, an MDP can be described as a 6-tuple (S, A, r, p, γ, o) , where S is the state space, A is the action space, r is the reward function, p is the state transition distribution, γ is the observation space and o is the observation probability distribution. In the multiple agents setting, the training network is centralized for policy learning and decentralized for action execution. Each agent has its own local observations. The detailed information about our state space, action space and reward function are described as follows.

State Space

As mentioned before, the observation o_i^t consists of current self agent situation map o_a^t , other agent situation map $o_{a'}^t$, obstacles (walls) map o_w^t , unexplored regions map (goals) map o_g^t and explored regions map o_e^t . The self-agent situation map o_a^t shows current self location while other agent situation map $o_{a'}^t$ indicates the locations of other agents. The obstacles map o_w^t represents the obstacles where an agent can not pass through. The unexplored regions map o_g^t represents a region that has not been explored or visited yet, and the explored regions map o_e^t represents empty regions that have been explored by the agents. The observations are processed by CNN to approximate the policy network. Specifically, those inputs include the measurements of the current frames from a 360-degree scanner which has a maximum range of 1. It provides the 84x84 matrix of the distance values as the input for each channel (i.e. $o_i^t \in \mathbb{R}^{3 \times 84 \times 84}$).

Action Space

The action space consists of directional actions in discrete space. Each agent has four actions: move up, move down, move right and move left. For each step the agent only can move one cell and the agent only can scan its surrounding within range of a single cell. The output size is $4i$, where i is the number of agents.

Reward Function

The rewards for each agent can be formulated as follows,

$$r_i^t = (g_r)_i^t + (p_r)_i^t + (f_r)_i^t. \quad (4.8)$$

where r_i^t is the reward received by agent i at timestep t which is the sum of three components, g_r , p_r and terminal reward f_r . In particular, each agent is rewarded by $(g_r)_i^t$ for arriving at an unexplored region

$$(g_r)_i^t = \begin{cases} r_{unexplored} & \text{if } \|P_i^t - P_g^t\| \leq 1.0 \\ 0 & \text{otherwise} \end{cases}, \quad (4.9)$$

where the P_i^t and P_g^t are the centre location of the agent and the unexplored region respectively.

The agent will be given a small punishment $(p_r)_i^t$ at each timestep. $(f_r)_i^t$ is a final reward and will be given when all the areas are completely explored. This thesis sets $r_{unexplored} = 0.2$, $p_r = 0.001$ and $f_r = 1.0$ in the training.

4.2 Enhancement of CMAPPO

4.2.1 Curriculum Learning Based Sampling

Although CMAPPO may be able to learn some efficient strategy based on the raw input image of the map, it can take too long to finish given the complexity of the environment and limited observation of each agent. In this case, curriculum learning is applied to direct the learning process to acquire knowledge more accurately. The learning is conducted in a teacher-student fashion that the teacher monitors the training process and different curriculum levels, while the student learns the given tasks.

Based on [15], the curriculum can be defined as training distribution Q_λ and formulate the curriculum learning as follows

$$Q_\lambda(z) \propto W_\lambda(z)P(z) \quad \forall z. \quad (4.10)$$

denote z as the random variable representing an example, $P(z)$ is the target training distribution, $W_\lambda(z)$ is the weight applied to example z at stage level λ in the curriculum sequence, and $0 \leq \lambda \leq 1$. The entropy of Q_λ distribution is increasing with

---

Algorithm 4: CMAPPO Algorithm

Input : Environment situation observation images
Output: Motion action for each agents

```

1 Initialize the policy network  $\pi_\theta$  and value function  $V(s_t)$ .
2 while True do
3   for agent  $i = 1, 2, \dots, N$  do
4     Run policy  $\pi_{\theta_{old}}$  in environment for  $T$  time-steps.
5     Process the observation through the CNN model, obtaining  $o_i^t$ .
6     Obtain observation, reward and action  $\{o_i^t, r_i^t, a_i^t\}$ .
7     Compute advantage estimates  $\hat{A}_1, \dots, \hat{A}_T$  using Equation 4.7.
8     If  $T_i > T_{max}$ , Break (where  $T_{max}$  is the maximum step)
9   end
10   $\pi_{\theta_{old}} \leftarrow \pi_\theta$ 
11  for  $j = 1, \dots, E_\pi$  do
12     $L^{KL PEN}(\theta) = \hat{\mathbb{E}}_i^t[\frac{\pi_\theta(a_i^t|o_i^t)}{\pi_{\theta_{old}}(a_i^t|o_i^t)}\hat{A}_i^t - \beta KL[\pi_{\theta_{old}}|\pi_\theta]]$  according to Equation 4.2
13    if KL divergence is not satisfied then
14      break
15    end
16    Update  $\theta$  by Adam optimizer according to Equation 4.5
17  end
18  for  $k = 1, \dots, E_V$  do
19     $L^V(\phi) = -\sum_{i=1}^N \sum_{t=1}^{T_i} (\sum_{t'=t}^{T_i} \gamma^{t'-t} r_i^{t'} - V_\phi(s_i^t))^2$ 
20    Update  $\phi$  by Adam optimizer according to Equation 4.5
21  end
22  if  $KL[\pi_{\theta_{old}}|\pi_\theta] > \beta_{high} KL_{target}$  then
23     $\beta \leftarrow \alpha\beta$ 
24  end
25  else
26     $KL[\pi_{\theta_{old}}|\pi_\theta] < \beta_{high} KL_{target}$  end
27     $\beta \leftarrow \beta/\alpha$ 
28  end

```

$W_\lambda(z)$ and should meet the following requirements

$$H(Q_\lambda) < H(Q_{\lambda+\epsilon}) \quad \forall \epsilon > 0, \quad (4.11)$$

In this case, $W_\lambda(z)$ is monotonically increasing whenever new example (ϵ) is added.

Figure 4.2: The architecture of curriculum learning

The thesis illustrates the curriculum learning model with reinforcement learning and the teacher-student curriculum interaction in Fig. 4.2. At each time-step, the teacher can choose the difficulty level of task and aim that the student can success on the target task.

4.2.2 Intrinsic-driven Exploration

To improve the exploration, it is possible to provide the rewards not only from the environment or the domain itself but also from a condition intrinsic to the agent. The idea of intrinsic reward is introduced by Pathak et al. [16]. They propose that intrinsic reward can be a signal to enable the agents to explore the environment more rather than just getting extrinsic rewards. The approach is to reward the agents according to how much surprising the environment is, which drives the agents to learn more strategies to explore the unvisited area and find more about the surprising states. Specifically, agents tend to discover the new area (unexplored regions) more often.

With the intrinsic reward, the networks are trained by a forward and an inverse model. In the inverse model, the policy π is trained to optimize the sum of the extrinsic reward and the intrinsic reward and encodes the states s_t and s_{t+1} into features $\phi(s_t)$ and $\phi(s_{t+1})$ that are trained to predict action a_t . In the forward model, $\phi(s_t)$ and a_t are the inputs to predict the feature representation $\hat{\phi}(s_{t+1})$ of s_{t+1} . The prediction error in the feature space is used as the intrinsic reward, which means a bigger difference of observations enables the agent to obtain a higher intrinsic reward.

$\hat{a}_t = (s_t, s_{t+1}; \theta_I)$ is used to represent the latent space and L_I (intrinsic loss) to calculate the difference between reality and prediction. The later state can be predicted by action and current state feature, denoted as $\hat{\phi}(s_{t+1}) = f(\phi(s_t), a_t; \theta_F)$. L_F (PPO loss) is used to calculate the difference.

For optimizing the CMAPPO, the loss of final model can be formulated as follows

$$\min_{\theta_\pi, \theta_I, \theta_F} \left[-\lambda \mathbb{E}_{\pi(s_t; \theta_\pi)} [\Sigma_t r_t] + (1 - \beta) L_I + \beta L_F \right], \quad (4.12)$$

where θ_π is optimized to maximize the expected sum of rewards, θ_I is the likelihood estimation that is maximum by minimizing L_I amounts under a multinomial distribution, L_I is the loss function that measures the discrepancy between the predicted and actual actions, θ_F are optimized by minimizing the loss function L_F , $0 \leq \beta \leq 1$ is used to weigh the inverse model loss against the forward model loss, $\lambda > 0$ is used to weigh the importance of the policy gradient loss against the importance of learning the intrinsic reward signal.

The intrinsic reward $^i r$ is computed as $\eta L_F(\hat{\phi}(s_{t+1}), \phi(s_t))$, where η is a scaling factor. The completed rewards can be formulated as follows

$$r_i^t = (^g r)_i^t + (^p r)_i^t + (^f r)_i^t + \zeta (^i r)_i^t, \quad (4.13)$$

This thesis sets $\eta = 1/128$ and $\zeta = 0.01$ in the experiments.

4.3 Experiments and Results

4.3.1 Experiment Environment

The algorithm of CMAPPO is implemented with TensorFlow [80] and the group of agents with scanning camera and the environment are simulated in the Unity simulation framework. This thesis trains the multi-agent coordinated exploration on a computer with a Ryzen Threadripper 1900X CPU and an Nvidia GTX 1080 Ti GPU.

The environment is presented as a 20 x 20 cells space. Each unit in the environment is assigned to one object (Figure 4.1): obstacle (red), explored region (green), unexplored region (black) and agent (self-perspective agent is blue, other agents are grey).

In the training experiment, each agent can only move one cell within one cell scanning range at a time in the environment. The exploration scene will be reinitialized when the agents are reaching the maximal steps or completing all the tasks. To demonstrate the robustness of the model, this thesis randomly places the agents and doors, generating random office environments for the agents at each re-initialization. The Average Moving Distance (AMD) is used as the measurement, which is the average moving cells of 100 times experiments. As for the evaluation, Table 4.1 shows the results from the experiments as averaged over 100 runs for each different environ-

Figure 4.3: The sample maps of different stage levels. The top is the maze-like environment, the bottom is the office-like environment.

ment.

4.3.2 Curriculum Experiment Environment

The environments with curriculum learning have different lesson levels. The first lesson places the agents in one room to explore, so that the agents only learn to reach unexplored regions within a small area. The main objective here is to assist the agents to learn useful exploration knowledge and develop collaboration strategies first. This thesis creates a simple curriculum with 4 stages, which can be illustrated as four different uncovered regions as in Fig. 4.3.

The maximal steps are 1000 for each individual agent. The strength of the entropy regularization β is 0.001, discount factor γ is 0.7, and update value estimate λ is weighted by 0.95, with the constraint parameter ε 0.2.

4.3.3 Evaluation of CMAPPO

In order to evaluate the performance of our proposed CMAPPO model (with curriculum learning and intrinsic reward), new maps are randomly generated as our testing scenes (examples are illustrated in Fig. 4.4, the doors are randomly positioned). The CMAPPO method is applied in a varying difficulty level of the environment.

In all trials with CMAPPO and curriculum learning, the agents can completely explore the entire areas in the environment. Without the curriculum stages and the intrinsic reward, it fails to complete the exploration tasks as shown in Fig. 4.5 as no CI configuration. The figure shows the average steps taken to explore the entire

Figure 4.4: (a) and (b) are the example of randomly generated maze-like scene and office-like scene respectively.

environment in every consecutive stage of training with the curriculum-based method. It is shown that, features learned from a simpler environment in one stage enables faster more stable learning on another more complex one in the next subsequent stage of learning. By applying the curriculum-based learning, the agents can also generalize their learned knowledge to be applicable in a variety of conditions and environments.

CMAPO is able to learn the strategies for coordinated exploration in the complex unknown environment. The results are shown in Table 4.1, which compares CMAPO with the conventional Frontier-based method applied with an ad hoc strategy to efficiently allocate the exploration tasks to different agents. The comparison is done for different unknown environments.

The Frontier-based method is a coordinated multi-agent exploration algorithm [17] [7]. Each agent selects its target based on the distance to the possible target frontier points and the utility of those points. During the exploration, the detailed selection criteria of how the agent i chooses a target t can be described as follows

$$(i, t) = \arg \max_{(i', t')} (U_{t'} - \beta \cdot V_{t'}^{i'}) \quad (4.14)$$

where $U_{t'}$ is the computed utility of target t , utility is defined as the minimum distance of an agent within scanning range to visit, and $V_{t'}^{i'}$ is the distance to target t . β is the relative importance of utility and distance, which generally is set to 1.

It can be seen that CMAPO takes less steps than Frontier-based method. In Table 4.1, it shows that CMAPO can make the agents explore the environment efficiently. When the number of agent is small, the CMAPO agents can significantly outperform the frontier-based ones in terms of AMD. With more agents involved, the difference of AMD is reduced, but CMAPO agents can still explore the environment more efficiently.

Table 4.1 shows that the number of steps taken using the strategy learned by CMAPO is consistently smaller for all configurations of environment and the number of agents than the one using the Frontier-based strategy which also considers other

(a) One Agent

(b) Two Agents

(c) Three Agents

(d) Four Agents

Figure 4.5: The AMD of average two training samples (maze-like maps) with CMAPPO method in 4 stage levels compared to the model without curriculum learning and intrinsic reward (No C&I). The total training episodes are 35,000, averaged over 100 runs.

Agents/Method	CMAPPO		Frontier-based Method	
	Maze	Office	Maze	Office
One agent	501	501	758	933
Two Agents	387	386	676	605
Three Agents	348	311	440	410
Four Agents	226	254	334	371

Table 4.1: Performance metrics of AMD evaluated for CMAPPO method, and Frontier-based method on the target unknown environment with different number of agents (the results are rounded off to the nearest integer).

agents situations. This indicates that the deep reinforcement learning of CMAPPO can acquire an effective coordination strategy directly from raw input images from the map of the environment without explicitly taking the utility or cost information of other agents into consideration.

4.4 Conclusion

In this work, the multi-agent exploration method is developed wherein the agents learn their coordinating strategy to allocate their tasks from experiences. This thesis combines the concept of curriculum learning and intrinsic reward to a deep reinforcement learning system called CMAPPO to solve multi-agent exploration in unknown environment problems which only takes raw visual observation as the inputs. The thesis demonstrates the learning capabilities of the proposed method against the state-of-the-art Frontier-based exploration method in various environment and conditions. It has been demonstrated that CMAPPO can learn better exploration coordination strategies that consistently outperform the Frontier-based method with a fixed mechanism for multi-agent task allocation based on cost and utility.

In the future, the study can be extended to include bigger more complex environment evaluated in actual robotic systems to evaluate the practicality of the approach. The proposed model can also be extended further by including other kinds of intrinsic motivation such as exploration strategies based on novelty.

Chapter 5

Conclusion and Future Work

In the preceding chapters, the need for multi-agent exploration and the existing work in this field are discussed. Artificial intelligence is a popular trend in human development and will involve more in our future life. Robotics as a fundamental area of artificial intelligence helps to provide solutions to some difficulties where are hard or impossible to be achieved by the human. Agent exploration problem as a significant issue in the robotics area is very close to our life, especially, the multiple agents' exploration. It can provide solutions to many applications, such as search and rescue in a disaster. As artificial intelligence becoming more and more important, how to make agents behave more sophisticated is the main research target for us. The existing exploration methods are more focus on the optimization approach which may not offer an efficient and adaptive exploration strategy when it applied to a dynamic and complex environment. This thesis presents two approaches that are effective and novel in multi-agent exploration problem.

5.1 Conclusion

In this thesis, new algorithms for multi-agent exploration with the segmentation-based method and learning-based method are developed. The segmentation-based method is based on map partitions to allocate the agents to different areas. Although the segmentation-based method can already handle many complex exploration environment tasks, the exploration strategy can be further improved to tackle some unpredictable situations. For example, a team of agents are usually as a group during the exploration, which means if some errors occur to one agent, the whole group will be influenced. Although reinforcement learning can solve many tasks, it is still far away from fully understanding the multi-agent exploration problem. Along with the success and progress of deep learning technologies, building a robust and efficient

multi-agent exploration system is possible. This thesis investigates using deep reinforcement learning approach to deepen the understanding of multi-agent exploration and develop a new algorithm to handle the exploration problem more robust.

This thesis introduces a new indoor room segmentation method called HAC, which is designed for multi-agent exploration on partially known and unknown environment. The HAC is a hierarchical fashion that provides robust and inflexible performance in different size of rooms or irregular rooms. With the adaptive segmentation method, the team of agents can understand the environment better and provide an exceptional collaborative exploration strategy. Moreover, a new technique is developed for indoor map segmentation that can be used in real-time exploration or before the exploration. Based on the HAC, the exploration approach is further developed on both known map exploration and unknown environment exploration. Also, the proposed segmentation method and segmentation-based exploration method are evaluated with typical state-of-the-art methods, which can achieve the optimal results.

Subsequently, this thesis introduces a new end-to-end CNN-based multi-agent exploration with the deep reinforcement learning method. A raw visual observation situation understanding technique is developed which can be used as the input to our reinforcement learning model. This reinforcement learning model is further developed to a multiple agents learning model which is based on the single-agent PPO model. Exploration of the unknown environment remains challenges, but our model can capture and understand the dynamic and various environment situation to generate a reasonable multi-agent exploration strategy. Inspiring from human behaviour, the various difficulty levels curriculum and intrinsic reward are designed, leading the agents quickly learning a stable and solid knowledge. CMAPPO is tested and evaluated on the target environment, and the results present our model can learn a more efficient strategy for multiple agents to explore the environment than the conventional frontier-based method.

5.2 Future Directions

The ultimate goal of multi-agent exploration is to develop strategies that can handle different dynamic and complex situations. There are two main difficulties in developing exploration strategies. First, the dynamic of the situation is hard to predict. For example, the agents can meet some unpredictable situations, such as temporary communication interruption, or even the malfunction. Such dynamics make it difficult to allocate a good target for each agent without the knowledge of the whole environment. Second, most existing learning methods are only applied with a single agent

and simple environment. However, in reality, it is often more complex and requires multiple agents to complete a task collaboratively.

Deep learning provides opportunities to develop improved exploration strategies in the future. Here this thesis discusses several possible future directions to develop multi-agent exploration strategies. 1) solving multi-agent exploration with deep learning methods: since multi-agent learning-based methods have a larger training space and actions than a single agent system, it is hard to be trained in the past. With the development of computation machines, it can be further researched and improved nowadays. 2) a good representation for multi-agent exploration: the proposed CMAPPO approach uses CNN to analyze the environment situations, however, it is sometimes hard to learn and requires a long training time. Understanding the environment with visual observation images is difficult, since the machine needs a long time to learn a little change of the images. Thus, if an approach can successfully be developed to provide a better representation for agents, it may improve the exploration performance. 3) handling the dynamics during the exploration: besides the situations like an interruption, a common situation is that agents may explore one target at the same time because learning and predicting others action are difficult. If dynamic situations can be considered, it may be able to develop a more robust model.

The future directions are summarised as following:

- Build a dynamic environment situation understanding model for agent exploration, which should be an effective and efficient visual observation analysis model and is specific for the exploration task.
- Extend the proposed multi-agent reinforcement learning approach to a more complex task. Each agent can have different tasks and cooperatively complete a big mission.
- Increase the number of agents to solve more realistic and intricate problems.
- Apply CMAPPO cooperative multi-agent framework to different applications, providing robust and intelligent solutions.

Bibliography

- [1] Camillo J Taylor and David J Kriegman. Exploration strategies for mobile robots. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 248–253. IEEE, 1993.
- [2] Brian Yamauchi. Autonomous urban reconnaissance using man-portable ugvs. In *Unmanned Systems Technology VIII*, volume 6230, page 62300S. International Society for Optics and Photonics, 2006.
- [3] Hoang Huu Viet, Viet-Hung Dang, Md Nasir Uddin Laskar, and TaeChoong Chung. Ba*: an online complete coverage algorithm for cleaning robots. *Applied intelligence*, 39(2):217–235, 2013.
- [4] Dieter Fox, Jonathan Ko, Kurt Konolige, Benson Limketkai, Dirk Schulz, and Benjamin Stewart. Distributed multirobot exploration and mapping. *Proceedings of the IEEE*, 94(7):1325–1339, 2006.
- [5] Sebastian Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.
- [6] Patrick Beeson, Nicholas K Jong, and Benjamin Kuipers. Towards autonomous topological place detection using the extended voronoi graph. In *Proceedings of 2005 IEEE International Conference on Robotics and Automation (ICRA 2005)*, volume 4, pages 4373–4379, 2005.
- [7] Kai M Wurm, Cyrill Stachniss, and Wolfram Burgard. Coordinated multi-robot exploration using a segmentation of the environment. In *Proceedings of 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008)*, pages 1160–1165. IEEE, 2008.
- [8] DeLong Zhu, Tingguang Li, Danny Ho, Chaoqun Wang, and Max Q-H Meng. Deep reinforcement learning supervised autonomous exploration in office environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7548–7555. IEEE, 2018.

- [9] Gail A Carpenter and Stephen Grossberg. *Adaptive resonance theory*, pages 1–17. Springer, Boston, MA, 2016. ISBN 978-1-4899-7502-7.
- [10] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
- [11] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. Deep face recognition. In *bmvc*, volume 1, page 6, 2015.
- [12] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.
- [13] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [14] Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1311–1320. JMLR. org, 2017.
- [15] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.
- [16] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2017.
- [17] Brian Yamauchi. Frontier-based exploration using multiple robots. In *Proceedings of the second international conference on Autonomous agents*, pages 47–53. ACM, 1998.
- [18] Antoine Bautin, Olivier Simonin, and François Charpillet. Minpos: A novel frontier allocation algorithm for multi-robot exploration. In *International conference on intelligent robotics and applications*, pages 496–508. Springer, 2012.
- [19] Richard Bormann, Florian Jordan, Wenzhe Li, Joshua Hampp, and Martin Hägele. Room segmentation: Survey, implementation, and analysis. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 1019–1026. IEEE, 2016.

- [20] Brian P Gerkey and Maja J Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, 23(9):939–954, 2004.
- [21] Wolfram Burgard, Mark Moors, Cyrill Stachniss, and Frank E Schneider. Coordinated multi-robot exploration. *IEEE Transactions on robotics*, 21(3):376–386, 2005.
- [22] Zoran Zivkovic, Bram Bakker, and Ben Krose. Hierarchical map building and planning based on graph partitioning. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 803–809. IEEE, 2006.
- [23] Emma Brunskill, Thomas Kollar, and Nicholas Roy. Topological mapping using spectral clustering and classification. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3491–3496. IEEE, 2007.
- [24] Nolan Bard, Jakob N Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, et al. The hanabi challenge: A new frontier for ai research. *arXiv preprint arXiv:1902.00506*, 2019.
- [25] Vijay Govindarajan, Subhrajit Bhattacharya, and Vijay Kumar. Human-robot collaborative topological exploration for search and rescue applications. In *Distributed Autonomous Robotic Systems*, pages 17–32. Springer, 2016.
- [26] Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente. Multiagent cooperation and competition with deep reinforcement learning. *PloS one*, 12(4):e0172395, 2017.
- [27] Michael Beetz, Moritz Tenorth, Dominik Jain, and Jan Bandouch. Towards automated models of activities of daily life. *Technology and disability*, 22(1, 2): 27–40, 2010.
- [28] Hendrik Zender, O Martínez Mozos, Patric Jensfelt, G-JM Kruijff, and Wolfram Burgard. Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems*, 56(6):493–502, 2008.
- [29] Fernand Meyer and Serge Beucher. Morphological segmentation. *Journal of visual communication and image representation*, 1(1):21–46, 1990.

- [30] Elisabetta Fabrizi and Alessandro Saffiotti. Augmenting topology-based maps with geometric information. *Robotics and Autonomous Systems*, 40(2-3):91–97, 2002.
- [31] Albert Diosi, Geoffrey Taylor, and Lindsay Kleeman. Interactive slam using laser and advanced sonar. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1103–1108. IEEE, 2005.
- [32] Oscar Martinez Mozos, Axel Rottmann, Rudolph Triebel, Patric Jensfelt, Wolfram Burgard, et al. Semantic labeling of places using information extracted from laser and vision sensor data. 2006.
- [33] Staffan Ekvall, Danica Kragic, and Patric Jensfelt. Object detection and mapping for service robot tasks. *Robotica*, 25(2):175–187, 2007.
- [34] Kristoffer Sjöö. Semantic map segmentation using function-based energy maximization. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4066–4073. IEEE, 2012.
- [35] Lluís-Pere de las Heras, Sheraz Ahmed, Marcus Liwicki, Ernest Valveny, and Gemma Sánchez. Statistical segmentation and structural recognition for floor plan interpretation. *International Journal on Document Analysis and Recognition (IJDAR)*, 17(3):221–237, 2014.
- [36] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.
- [37] Samuel Dodge, Jiu Xu, and Björn Stenger. Parsing floor plan images. In *Machine Vision Applications (MVA), 2017 Fifteenth IAPR International Conference on*, pages 358–361. IEEE, 2017.
- [38] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.
- [39] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113, 1997.

- [40] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [41] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.
- [42] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10): 1533–1545, 2014.
- [43] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.
- [44] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.
- [45] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.
- [46] Alessandro Giusti, Jérôme Guzzi, Dan C Ciresan, Fang-Lin He, Juan P Rodríguez, Flavio Fontana, Matthias Faessler, Christian Forster, Jürgen Schmidhuber, Gianni Di Caro, et al. A machine learning approach to visual perception of forest trails for mobile robots.
- [47] Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 3389–3396. IEEE, 2017.
- [48] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 3357–3364. IEEE, 2017.

- [49] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
- [50] Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai. Deep direct reinforcement learning for financial signal representation and trading. *IEEE transactions on neural networks and learning systems*, 28(3):653–664, 2017.
- [51] Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2137–2145, 2016.
- [52] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, pages 6379–6390, 2017.
- [53] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.
- [54] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
- [55] Martin L Puterman. *Markov Decision Processes.: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.
- [56] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization. *CoRR*, abs/1502.05477, 2015. URL <http://arxiv.org/abs/1502.05477>.
- [57] Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Rémi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample efficient actor-critic with experience replay. *CoRR*, abs/1611.01224, 2016. URL <http://arxiv.org/abs/1611.01224>.
- [58] Carlos Sampedro, Alejandro Rodriguez-Ramos, Hriday Bavle, Adrian Carrio, Paloma de la Puente, and Pascual Campoy. A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based

- techniques. *Journal of Intelligent & Robotic Systems*, 2018. doi: <https://doi.org/10.1007/s10846-018-0898-1>.
- [59] Richard Bormann, Joshua Hampp, and Martin Hägele. New brooms sweep clean-an autonomous robotic cleaning assistant for professional office cleaning. In *Proceedings of 2015 IEEE International Conference on Robotics and Automation (ICRA 2015)*, pages 4470–4477, 2015.
- [60] Ming Gao and J Marius Zöllner. Local contextual trajectory estimation with demonstration for assisting mobile robot teleoperation. In *Proceedings of 2017 European Conference on Mobile Robots (ECMR 2017)*, pages 1–8, 2017.
- [61] Gurkan Tuna, V Cagri Gungor, and Kayhan Gulez. An autonomous wireless sensor network deployment system using mobile robots for human existence detection in case of disasters. *Ad Hoc Networks*, 13:54–68, 2014.
- [62] Pär Buschka and Alessandro Saffiotti. A virtual sensor for room detection. In *Proceedings of 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*, pages 637–642, 2002.
- [63] Stephen Friedman, Hanna Pasula, and Dieter Fox. Voronoi random fields: Extracting topological structure of indoor environments via place labeling. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 2109–2114, 2007.
- [64] Jan Oberlander, Klaus Uhl, J Marius Zollner, and Rudiger Dillmann. A region-based slam algorithm capturing metric, topological, and semantic properties. In *Proceedings of 2008 IEEE International Conference on Robotics and Automation (ICRA 2008)*, pages 1886–1891, 2008.
- [65] Lluís-Pere de las Heras, Oriol Ramos Terrades, Sergi Robles, and Gemma Sanchez. Cvc-fp and sgt: a new database for structural floor plan analysis and its groundtruthing tool. *International Journal on Document Analysis and Recognition*, 18(1):15–30, 2015.
- [66] Thomas Frank, Karl-Friedrich Kraiss, and Torsten Kuhlen. Comparative analysis of fuzzy ART and ART-2A network clustering performance. *IEEE Transactions on Neural Networks*, 9(3):544–559, 1998.
- [67] Jianbo Shi and Carlo Tomasi. Good features to track. In *Proceedings of 1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 1994)*, pages 593–600. IEEE, 1994.

- [68] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. *International Journal of Computer Vision*, 9(3), 1991.
- [69] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- [70] Per-Erik Danielsson. Euclidean distance mapping. *Computer Graphics and image processing*, 14(3):227–248, 1980.
- [71] Kurt Bryan and Tanya Leise. The \$25,000,000,000 eigenvector: The linear algebra behind google. *SIAM Rev.*, 48(3):569–581, March 2006. ISSN 0036-1445. doi: 10.1137/050623280. URL <http://dx.doi.org/10.1137/050623280>.
- [72] Andrew Howard and Nicholas Roy. The robotics data set repository (radish), 2003. URL <http://radish.sourceforge.net/>.
- [73] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.
- [74] Ben Grocholsky, James Keller, Vijay Kumar, and George Pappas. Cooperative air and ground surveillance. *IEEE Robotics & Automation Magazine*, 13(3): 16–25, 2006.
- [75] Zoltán Beck, Luke Teacy, Alex Rogers, and Nicholas R Jennings. Online planning for collaborative search and rescue by heterogeneous robot teams. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 1024–1033. International Foundation for Autonomous Agents and Multiagent Systems, 2016.
- [76] Julia Fink, Valérie Bauwens, Frédéric Kaplan, and Pierre Dillenbourg. Living with a vacuum cleaning robot. *International Journal of Social Robotics*, 5(3): 389–408, 2013.
- [77] Zhang Libin, Yang Qinghua, Bao Guanjun, Wang Yan, Qi Liyong, Gao Feng, and Xu Fang. Overview of research on agricultural robot in china. *International Journal of Agricultural and Biological Engineering*, 1(1):12–21, 2008.
- [78] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

- [79] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [80] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.