Real-Time Hierarchical Map Segmentation for Coordinating Multirobot Exploration

TIANZE LUO[1], **ZICHEN CHEN**[1], **BUDHITAMA SUBAGDJA**[2], **AND AH-HWEE TAN**[2], (Senior Member, IEEE)

[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798
[2]School of Computing and Information Systems, Singapore Management University, Singapore 188065

Corresponding author: Tianze Luo (tianze001@e.ntu.edu.sg)

ABSTRACT Coordinating a team of autonomous agents to explore an environment can be done by partitioning the map of the environment into segments and allocating the segments as targets for the individual agents to visit. However, given an unknown environment, map segmentation must be conducted in a continuous and incremental manner. In this paper, we propose a novel real-time hierarchical map segmentation method for supporting multi-agent exploration of indoor environments, wherein clusters of regions of segments are formed hierarchically from randomly sampled points in the environment. Each cluster is then assigned with a cost-utility value based on the minimum cost possible for the agents to visit. In this way, map segmentation and target allocation can be performed continually in real-time to efficiently explore the environment. To evaluate our proposed model, we conduct extensive experiments on map segmentation and multi-agent exploration. The results show that the proposed method can produce more accurate and meaningful segments leading to a higher level of efficiency in exploring the environment. Furthermore, the robustness tests by adding noises to the environments were conducted to simulate the performance of our model in the real-world environment. The results demonstrate the robustness of our model in map segmentation and multi-agent environment exploration.

INDEX TERMS Autonomous agents, intelligent agents, multi-agent systems, agent-based modeling, image segmentation.

I. INTRODUCTION

Effective coordinated exploration of environment by a fleet of autonomous agents has remained a critical problem in application domains like search and rescue [1], cleaning [2], and sensor deployment [3]. As the environment can be partially or totally unknown, a model or map representation of the environment must be constructed incrementally in order to plan and assign the exploration tasks to the individual agents. What each agent should do or where it should go to visit next can only be determined after the map is constructed at least partially.

A typical approach to tackle this exploration problem is by identifying the so called frontier points (locations at the

The associate editor coordinating the review of this manuscript and approving it for publication was Emre Koyuncu.

boundary between the known and unknown areas) as the locations to visit by the different agents [4]. As an agent reaches a frontier point, new areas or regions are unfolded as the agent scans its surroundings so that new frontiers can be identified and allocated to the agents. This process may be repeated until all the areas in the environment are explored. For efficiency consideration, an agent is typically allocated to the frontier with the least-cost to visit (i.e., the nearest frontier location to the agent) [4]. Depending on the situation in hand, the basic frontier-based method may not be optimal especially when multiple agents are involved. Since agents are allocated to visit nearby points or locations, they may explore a relatively small area together instead of spreading more evenly to discover unknown places. One way to tackle this is by assigning every agent to a frontier location based on the utility or cost of visiting it given the number

of surrounding agents [5]. However, the effectiveness of this approach still depends on the condition of the environment to explore. The target frontiers to be allocated may also still be nearby to each other leading the agents to visit confined areas together at the same time. Another way to optimize this allocation is by firstly partitioning the scanned areas in the environment into regions or segments [6]. Based on the topological structure of the segments, the allocation to assign the agents to explore different locations can be done more efficiently. Wurm *et al.* employed the Voronoi algorithm to generate segments from a map of the environment and the Hungarian method to allocate the agents to the appropriate segments [6]. However, this approach is effective only when the initial information about the environment is available so that the partitions or segments for generating the topological structure are sufficient to make the efficient allocation.

More recent methods employ classical search and planning algorithms to handle the exploration and search allocation tasks in a decentralized manner [7]–[10]. However, the distributed versions of the allocation methods in [7] and [9] still need some offline mapping and evaluation of the environment beforehand. On the other hand, Best *et al.* [8] and Smith *et al.* [10] proposed decentralized control approaches to allow real-time and incremental allocation process, but they may require a great deal of computation and training time, especially to deal with large environments.

In this paper, we propose a novel incremental clustering method for real-time map segmentation and multi-agent exploration task allocation. The map segmentation method is designed to partition a given indoor environment map into meaningful segments (e.g., rooms, corridors, corners), in an incremental manner. Instead of requiring the entire map to be available upfront, the method is designed to work when the map is initially available only partially or totally unknown. The real-time incremental segmentation is conducted through a novel hierarchical clustering method inspired by Adaptive Resonance Theory (ART) neural network [11] that categorizes the point sampled or sensed by the agents in the environment.

Based on the segmented map, a task allocation method is applied to assign each agent to a segment based on a cost-utility function. While the cost measures the distance from the selected agent to a target location, the utility measures the potential of discovering a new target location. By applying the cost and utility function, we not only consider the cost of visiting the targeted location by an agent, but also consider the potential discovery of new areas during the exploration.

Overall, our proposed method offers a real-time incremental and continual exploration model for multiple agents in optimal and efficient manner while the information, knowledge, or map about the environment is initially lacking.

We compare our proposed hierarchical map segmentation method with the state-of-the-art segmentation methods based on a collected set of 30 real-world maps. The results show that our proposed method outperforms the state-of-the-art methods in producing more accurate and meaningful segments.

We further conduct comparative experiments under two settings: exploration based on known environments (wherein the map of the environment is available at the beginning) and exploration in unknown environments (wherein no map is available and one must be made from scratch). We compare the proposed method with the other state-of-the-art segmentation-based exploration methods, including Voronoi segmentation, distance-transform segmentation, feature-based segmentation, and morphological segmentation for task allocation in known environments. We also compare with the Frontier-based exploration method as the baseline for exploration in unknown environments. The evaluation results show that the proposed method outperforms all the methods mentioned above in terms of efficiency in reducing the distance travelled by the agents.

The remainder of this paper is organized as follows. Section II discusses the prior literature related to the multi-agent exploration problem. In Section III, we present the hierarchical adaptive clustering method for multi-agent map exploration, including segmenting a known map and incrementally segmenting an unknown map during exploration. Section IV describes the series of experiments conducted on multi-agent map exploration, with performance comparison with the frontier-based map exploration method and other segmentation based exploration method, including Voronoi segmentation, Morphological segmentation, distance transform segmentation and feature-based segmentation. Section V discusses the results of map segmentation and environment exploration. Finally, in Section VI, we conclude and discuss future work.

We summarize the contribution of our works as follows:

- We propose a new method called Hierarchical Adaptive Clustering to conduct map segmentation.
- Based on the Hierarchical Adaptive Clustering method, we further propose a multi-agent map exploration method, which leverages real-time map segmentation to coordinate multi-agent explorations.
- We conduct extensive experiments to compare our proposed methods with the state-of-the-art map segmentation and multi-agent exploration methods. The experiments show the superiority of our methods in both map segmentation and multi-agent exploration coordination.

II. RELATED WORK

In this section, we present and discuss other methods and approaches related to our work which include methods of map segmentation and multi-agent exploration.

A. MAP SEGMENTATION

Semantic map segmentation have been developed and studied for decades. Among them, Morphological Segmentation, Distance Transform-based Segmentation, Voronoi Graph-based Segmentation, and Feature-based segmentation can be considered as representatives of the existing map segmentation methods [12].

The morphological method of segmentation [13] worked on a grid map. During the segmentation, boundaries grow iteratively one pixel at a time into separate regions through the difference between dilation and erosion functions. If a separated region has a certain size between a lower and higher threshold, all the cells in this region are classified as an individual segment. This dilation-erosion process repeats until all the accessible cells in the grid map are marked as inaccessible. Afterwards, the labeled areas are extended to occupy all the unlabeled accessible area with a wavefront propagation. This process results in a well-labeled segmented map.

Diosi *et al.* [14] proposed a semi-autonomous method to perform map segmentation based on a distance transform method. The main idea of the distance transform-based segmentation is to find the centre of each segment via the distance transform and label the accessible areas with the wavefront propagation, which is similar to the morphological segmentation. The distance transform of each *accessible* pixel is the distance of this pixel to the nearest *inaccessible* pixel, and the local maxima of the distance transform lies at the center of a room. The room centers are uniquely labeled and the wavefront propagation extends the labeled area to the entire map.

The feature-based segmentation [15] uses features from 360° laser scanners which are placed at every accessible cell. The features are classified by Adaboost classifier to get the room labels such as office or corridors. Neighbour pixels with the same labels are merged to get the segmented map. The feature-based method requires a pre-training process from a pre-existing dataset.

The Voronoi graph-based segmentation is the best performing method among the others mentioned above [12]. It firstly computes a Voronoi graph on the map, and gets the critical points which have exactly two closest obstacle cells from the graph. The critical lines, which are the lines connecting the critical points and the closest obstacles, are drawn to separate the map into Voronoi cells. Finally, the cells are merged to form a segmented map according to a set of pre-defined rules. Relative methods for computing Voronoi graph have been elaborated in [16]–[18]. Although the Voronoi method can achieve the best performance among the others [12], it requires a set of manually defined rules to guarantee good performance. Similarly, the morphological and the distance transform-based approaches of segmentation also require manually set parameters to secure the performance.

In recent years, deep learning based methods achieve a high level of performance in segmentation tasks [19], such as R-CNN based models [20]–[22], Dilated Convolutional Models [23]–[25], RNN-based models [26], [27], and attention-based models [28], [29]. However, most deep learning methods require abundant training data for learning to perform proper image segmentation. Since in our dataset, the data samples, i.e., maps of the indoor environment available, are inadequate for training deep neural networks, we are not able to conduct experiments and perform comparisons with deep learning methods.

B. MULTIAGENT EXPLORATION

Studies on multi-agent exploration have emerged since last decades. Different techniques and approaches have been used like the frontier-based method wherein each agent makes its own decision to select a target to visit though they share a global world representation [30]. Bautin *et al.* improved the frontier-based exploration by ranking the agents to allocate to a particular frontier location based on their travel distances to the frontier [31]. More recently, the frontier-based exploration has also still been demonstrated as effective to construct a topological map of the environment by assigning an agent to visit a location based on the cost-utility function taking into account geometric, topological, and semantic criteria [32]. Other recent variants of frontier-based exploration on multi-agent exploration include a distributed multi-robot model [33] and some others involving communication constraints and collaboration among the agents [34]–[36].

The complexity of assigning many frontier points to different agents was significantly reduced by allocating the agents to segmented areas instead of frontiers. Wurm *et al.* used Voronoi algorithm to segment the floor plan into several regions and applied Hungarian method to optimally allocate the agents to the proper segment [6]. Voronoi algorithm scans areas in the map to generate a graph structure forming the segments' boundaries. The Hungarian method searches for optimal configuration of assigning the agents to the segments based on the costs taken to visit the corresponding locations. Although it can provide a cost-optimal solution for the agents to explore the environment, the allocation must be made simultaneously for all the agents and no values regarding exploring unknown areas are taken into consideration. Other segmentation-based exploration methods like morphological method, distance transform-based method, and feature-based method have also been proposed [12] though they are more context or domain dependent in terms of optimality compared with the Voronoi-Hungarian approach. More recently, several works have considered making the allocation in a distributed manner allowing exploration to be done in real-time. Omidshafiei *et al.* [7] and Chopra *et al.* [9] have proposed allocation methods that, however, require offline pre-processing of the map with strict spatio-temporal constraints for exploration. Best *et al.* [8] and Smith *et al.* [10] made use of Monte-Carlo Tree Search to incrementally produce exploration tree to share among the agents. The later, however, demands much more computation and exploration time.

Another approach to efficiently allocate the task is by semantically labeling features perceived in the environment. Beetz *et al.* used the perception method to conceptually tag features based on natural language labeling as they are identified during exploration [37]. Besides the map, the environment is modeled as a graph connecting different classifications of features such as rooms, corridors,

and doorways [38], [39]. However, this approach requires domain-dependent labeling and conceptual structure as prior knowledge.

Recently, some deep reinforcement learning methods have been proposed to tackle the multi-agent exploration tasks [40]–[44]. The reinforcement learning methods aim to learn a policy that maximizes the accumulated reward from an environment, requiring extensive training procedures to obtain the appropriate policy in the specific environment. However, the adaptiveness of the multi-agent exploration policy to any brand new environments remains a problem of the reinforcement learning methods.

III. PROPOSED METHOD

In this paper, we propose a hierarchical adaptive clustering model for map segmentation and a cost-utility based task allocation method for multi-agent exploration. In this section, we introduce the segmentation model as well as the multi-agent exploration task allocation method including detailed algorithms and performance analysis.

A. HIERARCHICAL ADAPTIVE CLUSTERING

The model of the proposed segmentation method consists of multiple levels of clustering systems. Each level can be considered as a modified Adaptive Resonance Theory (ART). ART is a family of neural network models that categorizes and grows clusters from inputs in a self-organized manner [45]. The particular ART model employed in this paper is a modified version that clusters inputs in Euclidean space and learns the center points (centroids) of the clusters. We apply two levels of clustering 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. In contrast, the original ART, as in [11], may represent clusters as areas or regions in a multi-dimensional space.

1) LOWER-LEVEL CLUSTERING NETWORK

Figure 1 shows the lower-level clustering network. Consider a location point $p = (x, y)$ in a $M \times N$ two-dimensional map of the environment. A normalized input $p = (p_1, p_2)$, where $p_1 = \frac{x}{M}$ and $p_2 = \frac{y}{N}$ is computed so that $p_1, p_2 \in [0, 1]$. A cluster node j in F2 layer is activated based on bottom-up Euclidean choice function

$$T_j = 1 - \|p - w_j\|_2$$

where w_j is a weight vector associating the input p in F1 with the node j in F2 and $\|p - w_j\|_2$ denotes the euclidean distance between p and w_j.

A node J is selected to be the category of p in F2 if it satisfies the resonance condition such that

$$T_J = \max\{T_j \mid \text{for all } j \text{ in F2}, T_j \geq \rho\}$$

where $\rho \in [0, 1]$ is the vigilance parameter. If no F2 node satisfies the resonance (for every j, $T_j < \rho$ always holds),
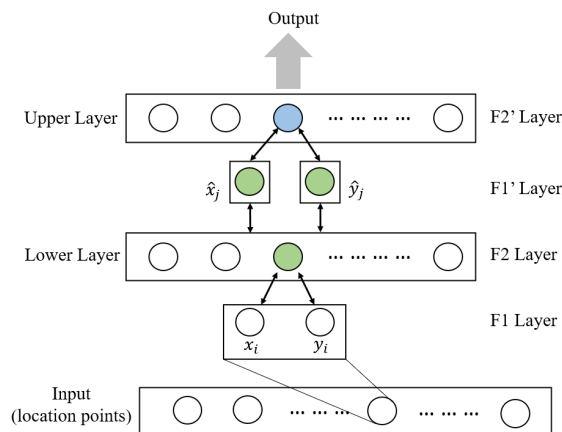

FIGURE 1. The model of hierarchical adaptive clustering, consisting of three layers, namely the *input layer* (F1 layer), the *lower layer* (F2 layer) for lower-level clustering, and the *upper layer* (F2' layer) for upper-level clustering. The model takes the normalized location points as inputs and clusters the input points through a two-level hierarchical clustering procedure. The output is the cluster center to which the input point belongs.

a new uncommitted node is recruited in F2 to represent the new input. Thus, the categories (clusters) are growing as the network encounters novel inputs.

Whenever a node J is selected, a weight update operation is conducted so that w_J is updated by

$$w_J^{(\text{new})} \leftarrow \frac{m^J . w_J + p}{m^J + 1}$$

where m^J is the number of input points categorized as J so far. Here, the weight w_J represents the center (centroid) of the cluster.

In the lower-level clustering, we consider a point p is inside a category j if its normalized position (p_x, p_y) has the closest Euclidean distance towards j (located at w_j) while no 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 w_j.

2) UPPER-LEVEL CLUSTERING NETWORK

Every center point $\hat{p}_j$ of every lower-level cluster node j in F2 is then presented consecutively as the input to F1' layer of the upper-level clustering network to generate the upper-level segments or clusters in F2' (Figure 1). This upper-level network works similarly as the lower-level one. However, it is 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 and the center point of the category being selected. This extra check is used so that any pair of upper-level cluster and low-level input cluster with a non-empty point (e.g., door, wall, partition) in between will be excluded from the cluster.
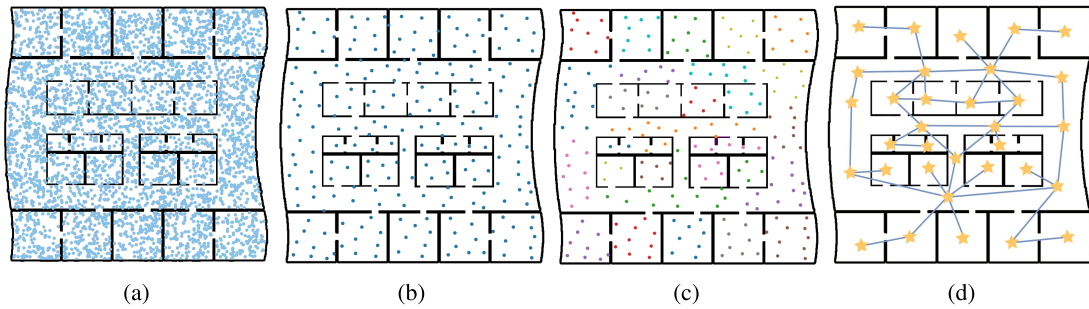
FIGURE 2. The process of hierarchical adaptive clustering: (a) Points sampling; (b) *Lower level* cluster centres; (c) *Upper level* clusters; and (d) The graph representation constructed form the upper level clusters.

FIGURE 3. The process of incremental segmentation during exploration: (a) Points sampling in the partially known area; (b) Lower level clustering and (c) Lower and Upper level clustering on the partially known area; and (d) Incremental graph construction based on the growing segment clusters.

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 Sample 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 p to present to lower level network in F1
5   Perform clustering with lower-level network to generate the set of cluster center points $\mathcal{S}_\mathcal{L}$ in F2
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 network in F1'
9   Perform clustering with upper-level network (excluding points with obstruction in between the cluster center) to generate the set of cluster center points $\mathcal{S}_\mathcal{U}$ in F2'
10 **end**
11 Construct graph G based on $\mathcal{S}_\mathcal{U}$

---

algorithm [48]. To simplify the matter, we set a threshold δ such that edges larger (longer) than δ can be excluded. In our experiment, we set δ to be the length of the shorter side of each 2-dimensional map. 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 2 (a) to (d).

C. COST-UTILITY BASED EXPLORATION TASK ALLOCATION

Based on the segments clustered from the map, an agent can be allocated to visit a segment based on a certain criteria. The criteria used in this paper is the least cost the agents need to spend to visit. Another criteria is the best opportunity or utility for a segment to be visited and explored.

In this part of the section, the values of the target allocation to select with the trade-off of utility and cost during the exploration are defined.

1) Cost

The cost of visiting a location by an agent is based on the traveling distance from the original location of the agent to the target. In this case, the shortest path distance based on A* path planning algorithm for 2D grid map is used.

B. MAP SEGMENTATION AND GRAPH CONSTRUCTION

Before the lower level segmentation starts, a pre-processing is conducted using the Shi-Tomasi approach to detect doors and door features within the area to be partitioned [46], [47]. The clustering is then conducted within a known or mapped area $\mathcal{A}$ in the environment. If the environment has been completely 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. In this case, unknown or unexplored areas in the environment are considered unsegmented.

At the end of the upper-level segmentation, a graph representation structure G can be formed by connecting the center points in the upper level cluster centres $\mathcal{S}_\mathcal{U}$ 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

With segmentation, the A*-based path distance from agent k to segment i is denoted as $d_{A_{k,i}^*}$. 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,

$$C_{I,K} = \left\{ d_{A_{k,i}^*} \mid \forall k, i \; k \in [1, m], i \in [1, n] \right\}. \tag{1}$$

2) Utility

When an agent is allocated to a segment location, it is reasonable not to allocate another agent to the same location especially when there is still another segment unexplored. The utility value can be considered as a measurement to predict whether a segment is worthwhile to visit given other segments and other agents in the area. 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,

$$U_i \leftarrow \min_i \frac{c_{i,k}}{\beta}, \; \forall i, k, \; c_{i,k} < \beta, \tag{2}$$

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. In our experiment, we set β equals to the shorter side of each 2-dimensional map.

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,

$$s_k \leftarrow \arg\max_i (U_i - \gamma.c_{i,k} + \sigma.w_i), \tag{3}$$

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 can be defined as

$$w_i = 1 - \frac{e_i}{\max_j(e_j)}, \tag{4}$$

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 preemptively comparing to a centre regions, as centre regions are easier to be visited by any agents in subsequent exploration. Based on the graph representation G obtained by segmentation in **Algorithm 1**, a Google Power Iteration [49] can be applied.

The Google Power Iteration, presented by Bryan *et al.* [49], is a well-known page ranking algorithm to calculate the importance value for each webpage, where each webpage is a node in the network or graph. This method outputs the centrality score for each node. We adopt the centrality score as the importance score $(e_1, \ldots, e_n)$ for the nodes in graph G in Eq (4). The importance score of a node is high if it connects to many other nodes, and vice versa. Thus, by conducting power iteration, nodes located at the center may have high importance scores, while those around the border of the environment tend to have lower scores.

Besides the significance of centrality in the environment, it is also possible to assign w_i in relation to another factor related to the environment or task conditions.

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
9   **end**
10   allocate agent k to segment s_k, where
         $s_k \leftarrow \arg\max_i(U_i - \gamma.c_{i,k} + \sigma.w_i)$
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   **end**
16 **end**

---

3) INCREMENTAL SEGMENTATION AND EXPLORATION

Every time an agent moves to its designated area, the cost and utility corresponding to the agent and its target change according to equations (1) and (2). Similarly, the best target segment to allocate may change as well according to equation (3). 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 newly explored area $\mathcal{A}$ and subjected to the segmentation process. In this case, 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 on the map or separately in a decentralized manner when multiple agents are involved. The incremental segmentation process can be depicted in Figure 3.

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 whenever an area $\mathcal{A}$ is unfolded by an agent. **Algorithm 2** shows the overall steps taken by the multi-agent exploration system, which ensures a full coverage of the entire map exploration. Specifically, the condition in Line 2 is satisfied if and only if there is no more frontier point on the map. Since frontier points mark the boundaries between the explored area and unexplored area, the void of frontier points indicates that there is no unexplored area in the map.

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
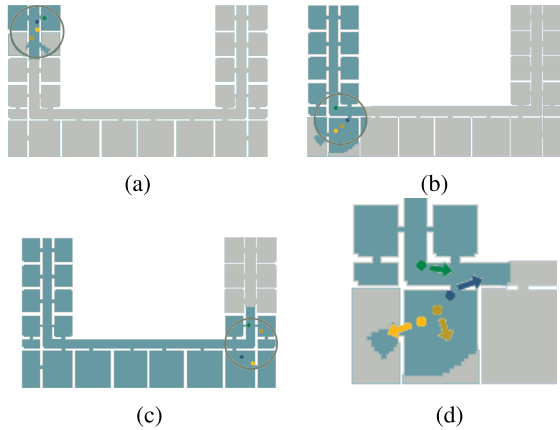
FIGURE 4. Illustration of hierarchical adaptive clustering based exploration by four agents in an unknown environment. The dark blue region indicates unexplored area, while the gray region indicates explored area. (a), (b) and (c) exemplify different exploration stages, while (d) shows the motion directions of each agent in more detail at the stage shown in (b).

may occur when the number of identified segments is less 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. A frontier is a location in the boundary between known and unknown area. Line 12 Algorithm 2 finds all frontiers (V) within the segment or area where the agent is currently resided in. A frontier v is then selected and assigned 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 [5], frontier allocation s'_k for agent k can be defined as follows,

$$s'_k = \arg\max_v(U'_v - \gamma \cdot c_{v,k}) \qquad (5)$$

where U'_v is the utility value of frontier point v, $c_{v,k}$ is the cost for agent k to visit v, and γ is the cost significance parameter. The utility U'_v is calculated in a similar way as calculating U_i.

Figure 4 illustrates the process of exploration as described in Algorithm 2 when it starts with an unknown environment. Figure 4(a) shows the starting condition after the agents scan the environment with most parts of the map still unknown. As shown in Figure 4(d), when a group of agents encounters several clusters, they tend to separate and spread out to explore different clusters farther away from each other according to the cost-utility based allocation as described in Algorithm 2. Since the allocation is based on existing frontier points available either within or outside segments, it is also guaranteed that all possible places or regions accessible by the agents will be visited or explored. In this case, the unavailability of frontier points becomes the termination criteria of the exploration.

IV. EXPERIMENTS

In this section, we evaluate and compare our hierarchical adaptive clustering based multi-agent exploration method with the existing state-of-the-art methods. To put the model in a practical sense, the evaluation is based on experiments with

a simulation of search and rescue tasks in an enclosed environment. The scenario of the simulation is based on exploration tasks by multiple robots or agents to identify victims trapped in an indoor environment or a ruined building in the aftermath of a disaster like the earthquake, fire, or flooding.

In this paper, the main task of the agents is scoped to visit potential enclosed areas and to send some information back to the base to construct the complete map and model of the environment in an efficient way possible despite the lack of prior map or initial information of the aftermath. Based on the constructed map, as the base receiving information from the agents, the segmentation process is conducted incrementally wherein the agents can be further assigned to unexplored areas.

A. MAP SEGMENTATION

We conduct the map segmentation experiments based on 30 floor plans extracted from the publicly available ROS-based room dataset [12], CVC-FP dataset [50] and R-FP dataset [51]. The floor plans are defined as grid-based map, where each cell in the map represents either an empty space or an obstacle.

We list the parameter settings of the segmentation method as follows.

- Number of sampled points n: 7000
- Lower-level vigilance ρ: 0.05
- Upper-level vigilance ρ^u: 0.2

To compare with our proposed hierarchical segmentation method, we use the implementation of the morphological, Voronoi, distance-based and feature-based methods, with default parameters as applied in [12].

The results are compared with the human-labeled ground truths, which are shown in the left-most column in Figure 5. We utilize the ground truths from [12] for some of the maps, and we label the ground truths for the rest.

B. DOMAIN CONFIGURATION AND SETTINGS FOR MULTIAGENT EXPLORATION

We use the grid world simulation to evaluate the performance of our proposed multi-agent exploration model. The test cases for multi-agent exploration are ten complex office maps from the ROS indoor room map collection [12], which are widely used in similar studies. Each map contains more than 20 separate rooms.

In our experiments, each map is captured in a grid-based representation as a $n \times m$ matrix, where n is the height and m is the width. Each cell in the grid is assigned to one of the following states:

- **Explored**: This cell is explored or visited by at least one agent.
- **Unexplored**: This cell has not been explored or visited by any agent.
- **Obstacle**: This cell is occupied by an obstacle that an agent cannot pass through.

To make the results comparable among different maps, we first define the map's width $m = 200$ cells, and its height is determined according to the aspect ratio.

Each agent can move in the environment one point or cell step at a time towards eight possible directions in a two-dimensional map. All agents are homogeneous or have the same kinematic or dynamic properties. The simulation does not allow an agent to move through an obstacle like a wall and similarly through each other. In addition, we assume each agent is equipped with a 360° LiDAR (Light Detection and Ranging) with a radius of $r = 15$ cells. During the exploration, in each step, each agent can move to any accessible neighbouring cell while scanning the local area within a radius r.

The experiments use the following parameter settings for segmentation:

Hierarchical clustering model
- Vigilance ρ in lower level clustering: 0.05.
- Vigilance ρ in upper level clustering: 0.2.

We set the vigilance ρ in upper layer according to the sensor range of the agents, such that an agent can explore the entire segment when it moves to the centre of the segment.

Multi-agent map exploration
- Cost parameter γ: 0.01.
- weight parameter σ: 0.35.

The average travel distance of every agent in an exploration trial is used as the performance measure to compare. This kind of measure has also been used commonly in other experiments on exploration [5], [6]. Another measure for comparing different methods in the experiment is the Average Travel Distance Reduction defined as follows,

$$Reduction = \frac{(d_b - d_p)}{d_b} \times 100\% \qquad (6)$$

where d_b represents the average travel distance of the baseline method, and d_p represent the average travel distance by the proposed or the evaluated method.

C. EXPERIMENT WITH KNOWN MAP
The first experiment is conducted to evaluate the efficiency of the task allocation based on the segmentation method. In this experiment, performances with different configurations of the number of agents (from 1 to 10 agents) are compared. We test different numbers of agents over four different starting points (top left/right corner and bottom left/right corner).

With the known map, the proposed task allocation method with hierarchical clustering segmentation method is compared with the basic frontier-based task allocation method in terms of travel distance reduction. The proposed method is also compared with the existing state-of-the-art methods for the average travel distance as evaluated as well in [5] including Voronoi-based method, Morphological method, Distance transform method, and Feature-based method. The best segmentation method according to [6] and [12] is the Voronoi-based method with Hungarian algorithm to optimize the task allocation.

D. COMPARISON IN UNKNOWN ENVIRONMENT
The second experiment is conducted to demonstrate the real-time incremental exploration task allocation based on hierarchical clustering for segmentation. The experiment setting regarding the number of agents and their starting points are the same as the experiment with the known map exploration.

In this experiment, the proposed method is compared with the Frontier-based method for travel distance reduction against the varying number of agents.

1) FRONTIER BASED METHOD
The frontier based method has been widely used for multi-agent exploration [5]. To perform the exploration, each agent is assigned a target location selected from a list of all frontier points. The selection is based on the distance (the cost) between the agent and the frontier point. It is also based on the utility of the target frontier point similar to the utility of a segment in the proposed method which depends on the number of the surrounding agents.

During the exploration, the selection criterion of how the agent k chooses a proper target frontier s_k^* can be described as:

$$s_k^* = \arg\max_v (U_v' - \gamma \cdot c_{v,k}) \qquad (7)$$

where U_v' is the computed utility of target v, $c_{k,v}$ is the distance from agent k to target v. γ is the relative importance of utility and distance, which generally is set to 1.

V. RESULTS AND DISCUSSION
Our proposed method has been implemented and evaluated in simulations and experiments. For generating the simulation results, we used the Python Matplotlib Animation Toolkit. All the experiments assumed that the agents share the global grid map that is produced by the sensor readings of all agents. The experiments are designed to test if the proposed method can significantly decrease the average travel distance compared to baselines and the state-of-the-art methods in both unknown and known environments.
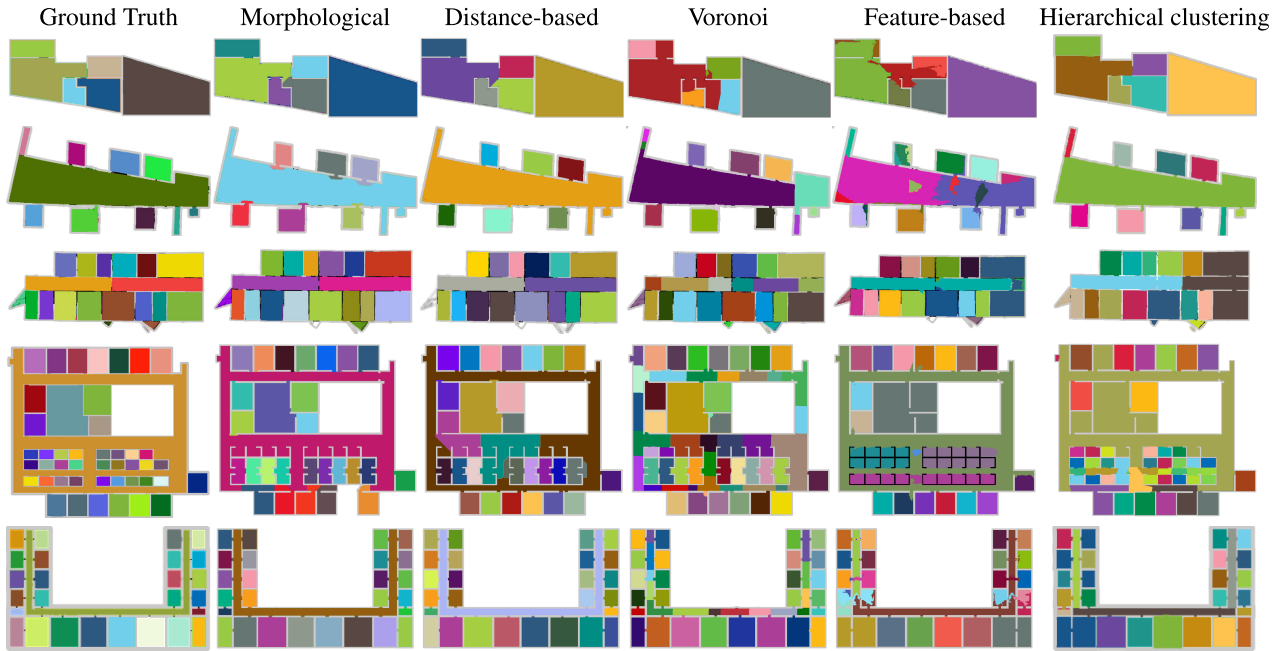
A. MAP SEGMENTATION
Metrics. Maps or floor plans with human-labeled segments from [12] are used as the ground truth to measure the accuracy and relevance of the outputs of the different segmentation methods. *Precision*, *Recall* and *F-measures* are used to quantitatively evaluate the methods.

$Recall = \frac{tp}{tp+fp}$, represents the containment of the generated segments in all areas of the ground truth. 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.

On the other hand, $Precision = \frac{tp}{tp+fn}$ can be defined as the containment of the human-labeled segments as the ground truth in all areas of the generated segments. In this case, fn is the total false negative or the areas of the human-labeled segments (ground truth) not coinciding with the areas of

TABLE 1. Average precision and recall (± standard deviation) of different segmentation methods over 30 maps.

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

FIGURE 5. Examples of segments generated by different segmentation methods compared with their corresponding ground truth are shown from left to right: The ground truth; Morphological segmentation; Distance-based segmentation; Voronoi graph-based segmentation; Feature-based room segmentation; and Hierarchical adaptive clustering segmentation.

the generated segments. F-measure $= \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$ is the harmonic mean of precision and recall.

The segmentation results are shown in Table 1. The quantitative comparison with respect to the ground truth shows that the segmentation accuracy of the proposed hierarchical adaptive clustering method is higher than other state-of-the-art methods. Through decomposing a map into small areas and hierarchically organising them to form semantic segments, the hierarchical adaptive clustering method can maintain fast and accurate segmentation. The cutting-edge techniques in the hierarchical adaptive clustering method can maintain a certain level of precision, which means neither under-segmentation nor over-segmentation happens.

Figure 5 shows some examples of the qualitative comparison of the hierarchical adaptive clustering method with other state-of-the-art methods. From the examples we can interpret that the morphological, distance transform and feature-based segmentation methods tend to oversegment the map. Voronoi graph-based segmentation, however, as a robust and accurate method, may still fail to segment the corridors correctly. The hierarchical adaptive clustering method can generally maintain high accuracy and robustness.

B. RESULTS OF KNOWN ENVIRONMENT EXPLORATION

Figure 6 summarizes the average travel distance and reduction rate by the proposed method in exploring the
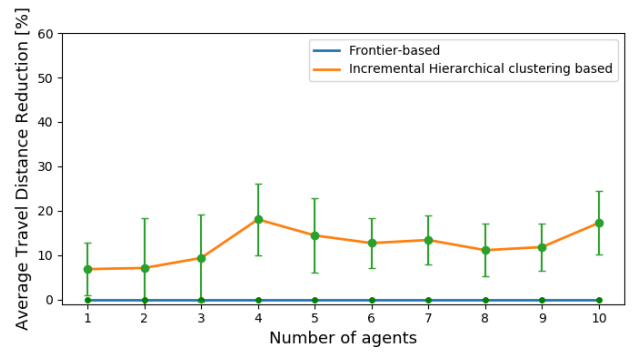
environment with and without node weighting. The results show that the proposed method can reduce the overall distance traveled to explore the environment compared to the baseline Frontier-based method. Moreover, comparing to the frontier-based method, more reduction can be observed as the number of agents is increased. In particular, our proposed model achieves more than 30% reduction (with around 20% standard error) from the baseline frontier-based approach for three agents on-wards. The hierarchical map segmentation provides a more effective task allocation strategy for a team of agents compared to the frontier-based task allocation methods.

We also compare the hierarchical clustering based exploration with and without the nodes' weights. Figure 6 includes the results when different weights for segments are used. In this case, a larger distance reduction can be obtained when fewer agents are involved. However, no significant reduction can be observed for larger numbers of agents. The reason is because when many agents are involved in exploration, the redundancy of the exploration secures all the regions to be easily accessed by any agent, no matter the regions are at the border or the center of the map. Although adding weightage to the nodes does not improve the performance when the number of agents is large, it still contributes to the task allocation when the number of agents is less than or equal three.
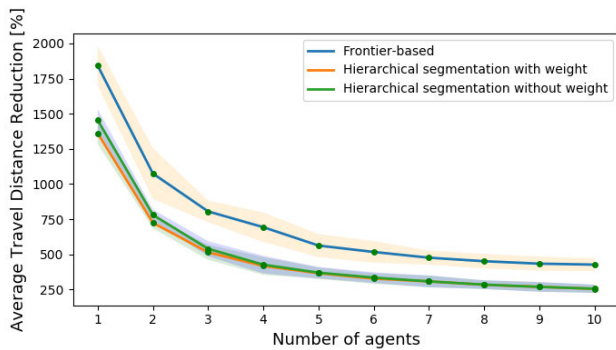
Figure 7 shows that the proposed hierarchical clustering based method consistently outperforms the other methods
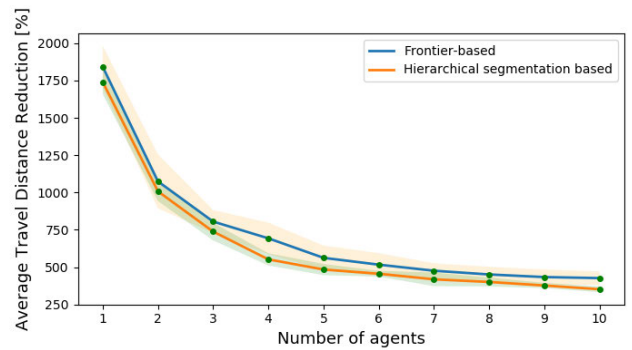
(a)
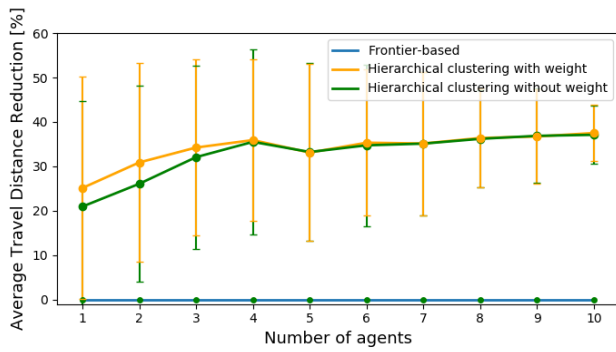
(b)

FIGURE 6. (a) Average travel distance reduction achieved by the hierarchical adaptive clustering based multi-agent map exploration under known maps with and without node weighting compared with the frontier based method. (b) The corresponding average travel distance of the segmentation-based multi-agent exploration (with and without weights) and the frontier-based method.

FIGURE 7. Average travel distance for hierarchical adaptive clustering based exploration compared with morphological, feature, voronoi, and distance-transform based exploration method.

including morphological, voronoi, distance transform and feature based exploration methods in terms of average travel distance in exploring the environment.

C. RESULTS OF UNKNOWN ENVIRONMENT EXPLORATION

Figure 8 shows the average travel distance and reduction rate when the environment is initially unknown. The results show when no map information is provided, the proposed exploration method achieves more than 10% reduction in travel distance compared to the baseline frontier-based method.

(a)

(b)

FIGURE 8. (a) Average travel distance reduction achieved by the hierarchical adaptive clustering based exploration with unknown maps from the frontier-based method. (b) The corresponding average travel distances.
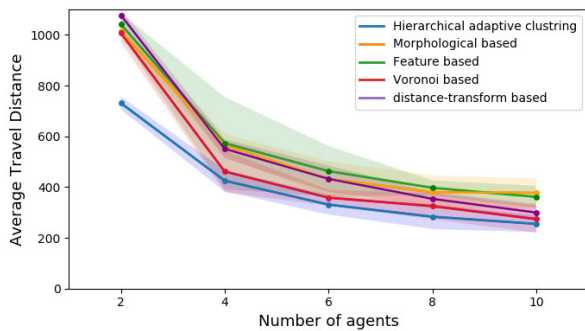
Furthermore, to test the performance of the method under different environments, we categorize the experiment maps into two types:

- maps with simple environment,
- maps with complex environment.

As the number of rooms grows, the map becomes more complex, and a better coordination strategy is needed to overcome the increase of complexity. We categorize maps with more than 30 rooms as the complex environment, whereas maps with 30 rooms or less belong to the simple environment. For example, in Figure 5, the map in the fourth row is a complex environment, while others are categorized as simple environments. We show the performance comparison between the proposed multi-agent exploration and the frontier-based method under both types of the map. The results are shown in Figure 9. The results show that with more complex maps, the proposed method can still allocate the exploration tasks better than the Frontier-based method.

We also test the robustness of the approach in more realistic settings by introducing 5% random noises to the environment as false detections of empty space, such that the sensor or the point sampling may get false detection of either empty spaces or due to the noises. Obstacles in the environment or those perceived by false detection may indeed affect the map segmentation and path planning procedure. For example, a single room may be wrongly segmented into two parts, and
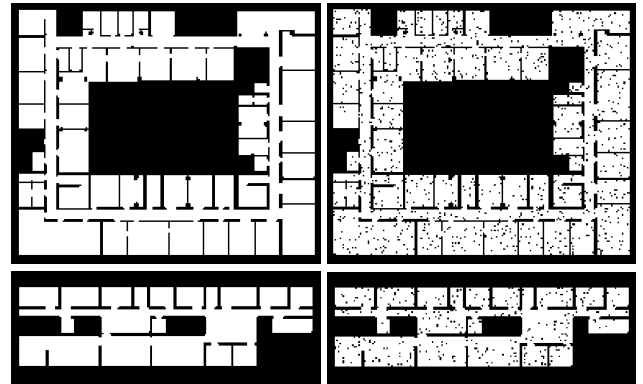
(a)

(b)

FIGURE 10. (a) Original maps. (b) Maps with 5% random noise as obstacles.
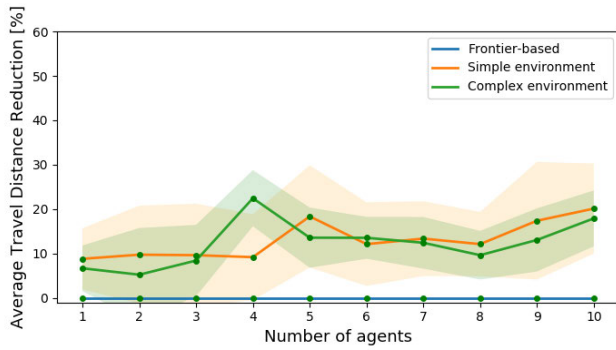
(a)
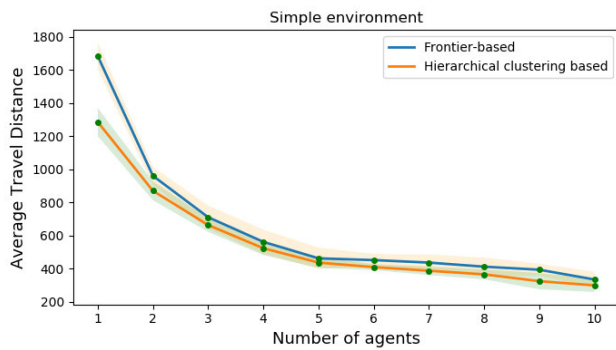
(b)
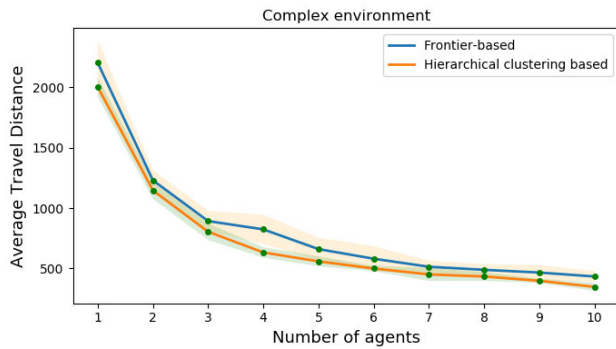
(c)

FIGURE 9. (a) Average travel distance reduction achieved by the hierarchical adaptive clustering based exploration compared with frontier based exploration method on maps with different complexities. (b) Average travel distances in the simple environment; (c) Average travel distance in the complex environment.

(a)

(b)

FIGURE 11. (a) Average travel distance reduction achieved by the hierarchical adaptive clustering based exploration from the frontier based exploration method in noisy settings. (The error bars indicate the standard error.) (b) The corresponding average travel distances.
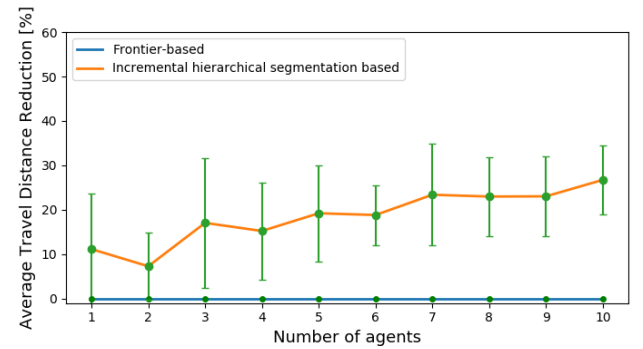
the agents will have to bypass the obstacles. Therefore, obstacles or noises in the environment reduces the efficiency of the exploration task. Figure 10 shows the samples of original maps and their noisy settings. The exploration results under unknown environment are shown in Figure 11. Nevertheless, even under the noisy environment, the experiment still shows a high level of robustness in our proposed exploration method. Specifically, despite the error in identifying some empty areas, our proposed method can still conduct more efficient exploration than the baseline frontier-based method. Overall, obstacles have a mild effect on our incremental segmentation and exploration procedure. Even if redundant
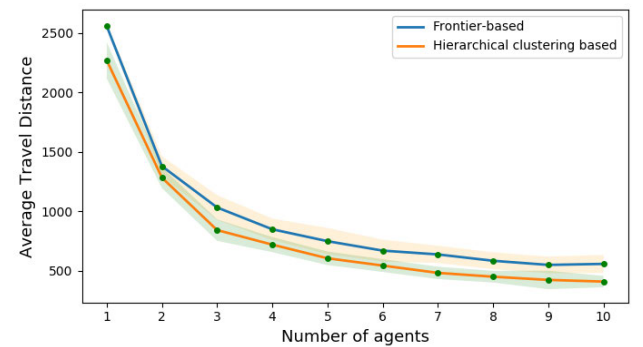
segmentation occurs due to noisy detection by the sensors, resulting in a whole room being segmented into parts, the task allocation method provided in Algorithm 2 can still rectify this mistake, as based on the cost-utility function, the nearest agent is highly likely to be assigned to explore both segments of the same room. Therefore, our proposed real-time segmentation and task allocation method can mitigate the dependency on the accuracy in the map segmentation. Similar to the noisy environment, in the case when the doorway is not accurately detected and the segmentation is not

well performed, our proposed method is still able to allocate agents to suitable positions based on our allocation algorithm and conduct incremental segmentation to further rectify the allocations.

In addition, the collaborative exploration is also scalable in terms of the number of agents. The results under noisy environment also indicate that despite the inaccuracy affecting individual costs and utilities, the overall outcomes of the allocation efficiency are not so much different from those in perfectly accurate configurations since they are based on local collective measures of crowd density obtained continuously while the agents move and new segments are identified.

VI. CONCLUSION

We have presented a novel method combining hierarchical map segmentation and cost-utility based task allocation for multi-agent exploration. The method allows the segmentation process and the task allocation to be conducted iteratively and incrementally in real-time, making it suitable for exploration in either known or unknown environment.

Compared with other state-of-the-art segmentation-based exploration methods, our approach performed better in terms of the average distance traveled by the agents. We also showed that the approach can reduce the overall travel distance compared with the standard frontier-based method to deal with unknown environment. In addition, it is demonstrated that the segmentation method together with the cost-utility-based task allocation criteria robustly supports the exploration process in reducing the overall travel distance even in the noisy or cluttered environment. Thus, our approach offers a solution towards addressing one of the hard problems in multi-agent exploration wherein the environment is unknown and/or noisy.

Going forward, we will scale up the investigation to look at how the segmentation and allocation methods can be extended to deal with more practical and realistic tasks or operations beyond exploration, such as search and rescue, wherein prior knowledge or information about the situation is lacking. Some physical aspects of behavior and constraints can be included like collision handling and diversity of physical properties to study their influences on the cost and utility of exploration. Similarly, additional factors in relation to heuristics or strategies of exploration like energy sources or number of potential people to find in a region as the weightage to speed up the search and rescue tasks. Furthermore, to deal with the variations in different domains and environments beyond floor plans, an integration with reinforcement learning techniques is necessary to enable continual adaptation and improvement of the task performance over a long period of time.

REFERENCES

[1] B. Yamauchi, "Autonomous urban reconnaissance using man-portable UGVs," *Proc. SPIE*, vol. 6230, May 2006, Art. no. 62300S, doi: 10.1117/12.660435.

[2] H. H. Viet, V.-H. Dang, M. N. U. Laskar, and T. Chung, "Ba*: An online complete coverage algorithm for cleaning robots," *Int. J. Speech Technol.*, vol. 39, no. 2, pp. 217–235, 2013.

[3] G. Tuna, V. C. Gungor, and K. Gulez, "An autonomous wireless sensor network deployment system using mobile robots for human existence detection in case of disasters," *Ad Hoc Netw.*, vol. 13, pp. 54–68, Feb. 2014.

[4] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart, "Distributed multirobot exploration and mapping," *Proc. IEEE*, vol. 94, no. 7, pp. 1325–1339, Jul. 2006.

[5] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 376–386, Jun. 2005.

[6] K. M. Wurm, C. Stachniss, and W. Burgard, "Coordinated multi-robot exploration using a segmentation of the environment," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2008, pp. 1160–1165.

[7] S. Omidshafiei, A.-A. Agha-Mohammadi, C. Amato, and J. P. How, "Decentralized control of partially observable Markov decision processes using belief space macro-actions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 5962–5969.

[8] G. Best, O. M. Cliff, T. Patten, R. R. Mettu, and R. Fitch, "Dec-MCTS: Decentralized planning for multi-robot active perception," *Int. J. Robot. Res.*, vol. 38, nos. 2–3, pp. 316–337, 2019.

[9] S. Chopra, G. Notarstefano, M. Rice, and M. Egerstedt, "A distributed version of the Hungarian method for multirobot assignment," *IEEE Trans. Robot.*, vol. 33, no. 4, pp. 932–947, Aug. 2017.

[10] A. J. Smith, G. Best, J. Yu, and G. A. Hollinger, "Real-time distributed non-myopic task selection for heterogeneous robotic teams," *Auto. Robots*, vol. 43, no. 3, pp. 789–811, Mar. 2019.

[11] G. A. Carpenter and S. Grossberg, "Adaptive resonance theory," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds. Boston, MA, USA: Springer, 2010, pp. 22–35, doi: 10.1007/978-0-387-30164-8_11.

[12] R. Bormann, F. Jordan, W. Li, J. Hampp, and M. Hagele, "Room segmentation: Survey, implementation, and analysis," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1019–1026.

[13] F. Meyer and S. Beucher, "Morphological segmentation," *J. Vis. Commun. Image Represent.*, vol. 1, no. 1, pp. 21–46, 1990.

[14] A. Diosi, G. Taylor, and L. Kleeman, "Interactive SLAM using laser and advanced sonar," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2005, pp. 1103–1108.

[15] O. M. Mozos, A. Rottmann, R. Triebel, P. Jensfelt, and W. Burgard, "Semantic labeling of places using information extracted from laser and vision sensor data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Workshop*, Jan. 2006, pp. 1–7. [Online]. Available: http://www2.informatik.uni-freiburg.de/~omartine/publications/mozos2006%iros_w.pdf

[16] S. Fortune, "A sweepline algorithm for Voronoi diagrams," *Algorithmica*, vol. 2, pp. 153–174, Nov. 1987.

[17] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, vol. 501. Hoboken, NJ, USA: Wiley, 2009.

[18] F. Karimipour and M. Ghandehari, "A stable Voronoi-based algorithm for medial axis extraction through labeling sample points," in *Proc. 9th Int. Symp. Voronoi Diagrams Sci. Eng.*, Jun. 2012, pp. 109–114.

[19] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Feb. 17, 2021, doi: 10.1109/TPAMI.2021.3059968.

[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2015, pp. 91–99.

[21] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.

[22] F. Sandelin, "Semantic and instance segmentation of room features in floor plans using mask R-CNN," Tech. Rep., 2019.

[23] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "DenseASPP for semantic segmentation in street scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3684–3692.

[24] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1451–1460.

[25] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2016.

[26] R. Hu, M. Rohrbach, and T. Darrell, "Segmentation from natural language expressions," in *Proc. Eur. Conf. Comput. Vis.*, Cham, Switzerland: Springer, 2016, pp. 108–124.

[27] Y. Xiang and D. Fox, "DA-RNN: Semantic mapping with data associated recurrent neural networks," 2017, *arXiv:1703.03098*.

[28] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3146–3154.

[29] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-cross attention for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 603–612.

[30] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Proc. 2nd Int. Conf. Auto. Agents (AGENTS)*, 1998, pp. 47–53.

[31] A. Bautin, O. Simonin, and F. Charpillet, "MinPos: A novel frontier allocation algorithm for multi-robot exploration," in *Proc. Int. Conf. Intell. Robot. Appl.*, Oct. 2012, pp. 496–508.

[32] C. Gomez, A. C. Hernandez, and R. Barber, "Topological frontier-based exploration and map-building using semantic information," *Sensors*, vol. 19, no. 4595, pp. 1–20, 2019.

[33] J. J. Lopez-Perez, U. H. Hernandez-Belmonte, J.-P. Ramirez-Paredes, M. A. Contreras-Cruz, and V. Ayala-Ramirez, "Distributed multirobot exploration based on scene partitioning and frontier selection," *Math. Problems Eng.*, vol. 2018, pp. 1–17, Jun. 2018.

[34] R. G. Colares and L. Chaimowicz, "The next frontier: Combining information gain and distance cost for decentralized multi-robot exploration," in *Proc. 31st Annu. ACM Symp. Appl. Comput.*, Apr. 2016, pp. 268–274.

[35] T. Andre and C. Bettstetter, "Collaboration in multi-robot exploration: To meet or not to meet?" *J. Intell. Robotic Syst.*, vol. 82, no. 2, pp. 325–337, May 2016.

[36] F. Amigoni, J. Banfi, and N. Basilico, "Multirobot exploration of communication-restricted environments: A survey," *IEEE Intell. Syst.*, vol. 32, no. 6, pp. 48–57, Nov. 2017.

[37] M. Beetz, M. Tenorth, D. Jain, and J. Bandouch, "Towards automated models of activities of daily life," *Technol. Disability*, vol. 22, nos. 1–2, pp. 27–40, Jun. 2010.

[38] H. Zender, O. Martínez Mozos, P. Jensfelt, G.-J.-M. Kruijff, and W. Burgard, "Conceptual spatial representations for indoor mobile robots," *Robot. Auto. Syst.*, vol. 56, no. 6, pp. 493–502, Jun. 2008.

[39] N. Mavridis, "A review of verbal and non-verbal human–robot interactive communication," *Robot. Auto. Syst.*, vol. 63, pp. 22–35, Jan. 2015.

[40] T. Wang, J. Wang, Y. Wu, and C. Zhang, "Influence-based multi-agent exploration," 2019, *arXiv:1910.05512*.

[41] A. Mahajan, T. Rashid, M. Samvelyan, and S. Whiteson, "MAVEN: Multi-agent variational exploration," 2019, *arXiv:1910.07483*.

[42] T. Luo, B. Subagdja, D. Wang, and A.-H. Tan, "Multi-agent collaborative exploration through graph-based deep reinforcement learning," in *Proc. IEEE Int. Conf. Agents (ICA)*, Oct. 2019, pp. 2–7.

[43] T. Wang, H. Dong, V. Lesser, and C. Zhang, "ROMA: Multi-agent reinforcement learning with emergent roles," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 9876–9886.

[44] H. Bae, G. Kim, J. Kim, D. Qian, and S. Lee, "Multi-robot path planning method using reinforcement learning," *Appl. Sci.*, vol. 9, no. 15, p. 3057, Jul. 2019.

[45] A.-H. Tan, B. Subagdja, D. Wang, and L. Meng, "Self-organizing neural networks for universal learning and multimodal memory encoding," *Neural Netw.*, vol. 120, pp. 58–73, Dec. 2019.

[46] J. Shi and Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 1994, pp. 593–600.

[47] C. Tomasi and T. Kanade, "Detection and tracking of point features," Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep., CMU-CS-91-132, 1991.

[48] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968.

[49] K. Bryan and T. Leise, "The $25,000,000,000 eigenvector: The linear algebra behind Google," *SIAM Rev.*, vol. 48, no. 3, pp. 569–581, Aug. 2006., doi: 10.1137/050623280.

[50] L.-P. de las Heras, O. R. Terrades, S. Robles, and G. Sánchez, "CVC-FP and SGT: A new database for structural floor plan analysis and its groundtruthing tool," *Int. J. Document Anal. Recognit. (IJDAR)*, vol. 18, no. 1, pp. 15–30, Mar. 2015, doi: 10.1007/s10032-014-0236-5.

[51] S. Dodge, J. Xu, and B. Stenger, "Parsing floor plan images," in *Proc. 15th IAPR Int. Conf. Mach. Vis. Appl. (MVA)*, May 2017, pp. 358–361.

TIANZE LUO received the B.Eng. degree (Hons.) in electrical and electronics engineering from the Nanyang Technological University, Singapore, in 2017, where he is currently pursuing the Ph.D. degree in computer science. His research interests include multi-agent systems, deep reinforcement learning, graph representation learning, and computer vision.

ZICHEN CHEN received the B.B.A. degree (Hons.) in information system from the Chung Yuan Christian University, Taiwan, and the M.Eng. degree in computer science and engineering from the Nanyang Technological University, Singapore. She is currently a Research Associate with the Joint NTU-WeBank Research Centre, Fintech. Her research interests include machine learning, federated learning, distributed learning, and security.

BUDHITAMA SUBAGDJA received the bachelor's and master's degrees in computer science from the Faculty of Computer Science, University of Indonesia, and the Ph.D. degree in information systems from the Department of Information Systems, the University of Melbourne, Australia. He was a Senior Research Fellow at Nanyang Technological University (NTU), Singapore. He was a Research Assistant and a Lecturer at the University of Indonesia. He is currently a Research Scientist with the School of Computing and Information Systems, Singapore Management University (SMU). His current research interests include biologically-inspired cognitive architecture for intelligent agents, self-awareness in social robots, planning, reasoning, and reinforcement learning in autonomous agents and multi-agent systems.

AH-HWEE TAN (Senior Member, IEEE) received the B.Sc. (Hons.) and M.Sc. degrees in computer and information science from the National University of Singapore, Singapore, and the Ph.D. degree in cognitive and neural systems from Boston University, Boston, MA, USA. He is currently a Professor of computer science and an Associate Dean of Research with the School of Computing and Information Systems, Singapore Management University (SMU). His current research interests include cognitive and neural systems, brain inspired intelligent agents, machine learning, knowledge discovery, and text mining. He is an Associate Editor of IEEE Transactions on Neural Networks and Learning Systems.

● ● ●