IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 5, 2020, accepted October 2, 2020, date of publication October 7, 2020, date of current version October 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3029324

TSM: Topological Scene Map for Representation in Indoor Environment Understanding

ZHIYONG LIAO, YU ZHANG, JUNREN LUO, AND WEILIN YUAN
College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China
Corresponding author: Yu Zhang (redarmy_zy@nudt.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61702528.

ABSTRACT In the field of robotics, it is crucial to obtain a comprehensive semantic understanding of a scene for many applications. Based on the behavioral topological map and scene graph, we propose to employ a semantic map named Topological Scene Map (TSM) for representation in indoor environment understanding. The behavioral topological map we constructed expresses the spatial connection relations and semantically describes the navigation behavior between adjacent topological nodes. The scene graph promotes the TSM to record the objects that appear in the scene and the relations between objects. The addition of spatial and semantic relations makes the expression of the scene more specific, which improves the robot's abilities of scene understanding and human-robotic interaction. In this article, we design a method for topological map construction and apply a novel approach to generate a scene graph from RGB-D data. The semantic representation of the environment generated in the experiments verifies that the TSM construction framework models the scene efficiently and the TSM is conducive to the realization of human-robotic interaction.

INDEX TERMS Scene graph generation, topological map construction, semantic map.

I. INTRODUCTION

In the field of robotics, modeling environment is fundamental before starting some tasks. The environment information is usually stored in the form of maps, such as metric map, topological map, and semantic map. Maps required by robots with different tasks are various.

For robots, an effective environment model should include the following elements:

- Applicability: The robot is able to perform various tasks with the model, not just a specific task.
- Accuracy: The model should be able to accurately describe the environment and provide the robot with correct information.
- Scalability: The model needs to adapt to the size of the environment, and expand the size of the expressed environment step by step.
- Usability: The model should be easy to use and be able to realize human-robotic interaction.

For humans, the dynamic changes of the indoor furniture placement, light intensity, and other factors do not seem to change the general understanding of the environment and will not affect the navigation. Previous biological research showed that the spatial information stored by biological navigation systems are coarse-grained [1]. These coarse-grained representations are the topological descriptions of the environment. The topological map has been widely used in the navigation of robots [2]–[5], and the recent rapid development of deep learning algorithms provides a new perspective for the application of topological map [6]–[8].

Robots need to understand the environment with the human mindset to improve intelligence, so they are capable of executing semantical commands consequently. The semantic map is proposed to describe the environment semantically, which assists robots to understand the environment. The semantic map for the robot contains the space and the entities semantic information. These entities have the attributes of some classes, more knowledge about them is obtained by reasoning with knowledge base [9]. More recent attention has focused on the semantic map to represent the environment intelligently. In [10], the author reviews studies of the semantic map, the research on semantic maps are divided into indoor and outdoor, single-scene and large-scene. The 3D semantic map construction framework proposed in [11] is a masterpiece of semantic map research, which expresses the environment with five levels: Metric-Semantic Mesh, Objects and Agents, Places and Structures, Rooms, Building. Besides, the framework applies the tracking and detection

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

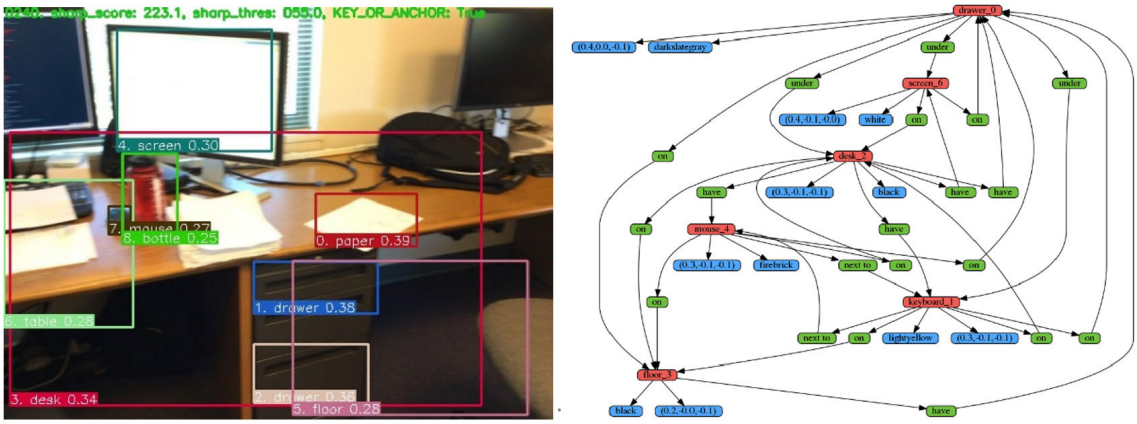

FIGURE 1. A scene graph example of indoor environment.

module for dynamic targets, which eliminate the effects of dynamic changes [12], [13]. However, this framework has strict requirements on hardware and the construction process of the map is complicated.

We divide the tasks performed by intelligent robots in an environment into the following stages. (1)**Environmental perception**. The robot collects information in the environment with sensors. For example, the RGB-D sensor collects images of the environment. (2)**Environmental semantic cognition**. Semantic information is attached to objects and spaces detected in the environmental perception stage to provide a basis for human-robotic interaction. (3)**Semantic task understanding**. Semantic tasks are understood through the semantic representation of the environment and decomposed into executable task sequences, such as navigation and obstacle avoidance, etc. (4)**Task execution**. Tasks in the executable task sequence are realized with the basic control system. This article focuses on environmental semantic cognition. In this article, we propose a lightweight sparse environment representation, Topological Scene Map (TSM), to effectively represent the environment. The methodological approach taken for TSM is a mixed methodology combined with the navigation behavioral topological map construction [6], [14] and the scene graph generation [15]. The navigation behavioral topological map records the connectivity of space and describes the navigation behavior between adjacent nodes, which allows the robot to generate navigation routes with semantic information. The scene graphs [15] is applied to preserve objects and relations appearing in the environment, which is a compact representation that expresses a complicated environment with less memory and integrates data from different domains, such as text and image. One scene graph of an indoor environment is shown in Figure 1. The specific objective of this article is to construct a robotic semantic TSM for the indoor environment with our framework. The TSM aims to model the environment semantically, which contributes to the interpretable reasoning that is the key to semantical tasks, like Q&A(Question and Answer) and navigation routes recommendation. When a robot needs to perform other tasks in the environment, the TSM is loaded as a global static map to construct a new environment representation that will be updated with observations.

In the construction process of TSM, the robot needs to complete the full coverage exploration of the environment firstly. During the exploration, the RGB-D sensor is applied to record the full coverage exploration video, and the laser sensor is used to build the metric map of the environment. Then, the behavioral topological map is constructed based on the metric map, which is employed to divide the exploration video into slice groups used to generate the scene graph for each room. Our contributions are the following:

- We design a process of constructing a topological map that represents the connectivity of the environment, in which the features of a node include ID, coordinates, and semantic labels, the edges include distance and navigation behavior information.
- We propose a method to split the environment exploration video according to the topological map. The full coverage exploration video needs to be split into slices according to the topological map, and door nodes are used to get timestamps for splitting.
- We generate the scene graph for every room of the indoor environment from the split exploration video. The scene graph record a complex environment with less memory, which facilitates semantic query and reasoning.
- The TSM construction framework combines a topological map for global information with a scene graph for local information to provides an efficient semantic representation of the environment. With TSM, the robot generates human-friendly navigation command sequences for human-robotic interaction.

The rest of this article is organized as follows. In Section II, some related works about the topological map, the scene graph, and the semantic map are presented. In Section III, We introduce the methodology of the TSM construction framework, including the process of navigation behavior

topological map construction, the Scene Graph Generation (SGG) model, the method of generating the scene graph from the video, and the method of fusing the scene graph with the topological map. In Section IV, We verify the construction process of TSM with the simulation environment. In Section V, we conclude this article and point out further directions.

II. RELATED WORK

A. TOPOLOGICAL MAP

The topological map is also defined as the roadmap, which is a sparse representation that describes the topological characteristics of the environment. The construction of topological maps is combined with metric maps [16], [17] or not [2]. Generalized Voronoi diagram [18] and spectral clustering [19] are the two main methods [20] to construct the topological map. In [21], spectral clustering and extended Voronoi diagrams are used to construct the topological map from the metric map. Spectral clustering is applied to segment the metric map and obtain the center of the cluster. In [22], a lightweight method for combining the metric and topological maps is proposed. With the combined map, the robot is able to autonomously navigate in a large scale environment and avoid the obstacle. Although the fusion of these two maps achieves obstacle avoidance navigation, the cost of storing the map is extremely. A concise process of topological map construction and effective topological representation are proposed in [5] to resolve the problem. While creating a topological map, it is necessary to determine which topological node each region of the environment belongs to. In [23], a grid middle layer is proposed to rasterize the metric map and distinguish different regions. The topological points that fall into the same region are attached with the same region label. In [24], the segmentation problem of 3D scenes is transformed into an integer programming problem to solve. Against the problem of localization and noise in the metric map, a visual topological map is constructed with the use of structured prior knowledge [8], in which each node is represented by a 360-degree panoramic image and the edges represent the transformings of posture.

According to the way humans model the environment, some new methods of topological map construction and application are proposed. In [25], a semi-parametric topology memory framework is constructed inspired by the landmark-based navigation, which consists of two parts, a parameterless topological map for memory and a parameterized deep network used to retrieve nodes from the topological map with observations. With the landmark, humans verify the understanding of the environment. In [26], a topological map based on landmarks is constructed, in which the landmark allows the robot to execute a task with some interference reliably. For humans, prior knowledge about some types of the environment plays an important role, while a new environment needs to be expressed. In [27], a new storage structure, named Bayesian Relational Memory (BRM), is proposed to store the prior knowledge. With BRM, robots construct the unknown environment representation with prior knowledge quickly. Maps in the human mind are usually attached with semantic descriptions. Inspired by this idea, the topological map attached with semantic information will be more practical. In [14], a navigation behavior topological map is constructed, in which the nodes and edges are attached with semantic labels. Inspired by the navigation behavior topological map, we propose a method to construct our topological map.

B. SCENE GRAPH

The scene graph is a sparse representation of semantic information [15], where nodes represent entities in the scene and edges represent spatial or logical relations. The objects appearing in the image are displayed as semantic elements in the scene graph and the relations in the scene graph will contribute to scene understanding and interpretable reasoning. Much of the current literature on the scene graph pays particular attention to generation and application [28].

The data for SGG is not limited to images, but also text and video [29]–[32]. Generally, the scene graph is not generated with a single image but related images, which improve the effectiveness of the scene graph. Besides, the scene graph is regarded as a commonsense knowledge graph generated according to the scene [33]. Therefore, SGG is regarded as a bridge between the scene graph and the commonsense graph. There are currently two main kinds of methods for SGG. The first kind of method is divided into two-stage, objects detection and then recognizes the relation between them [34]. The other applies region proposal to jointly reason the classes of objects and relations [35]. In [34], The SGG model *MOTIFNET* is proposed, which divides the scene analysis into three stages: delineating the object region, calibrating the region type, and predicting the relation between the regions. Each stage combines the global features of contextual information through bidirectional Long Short-Term Memory (LSTM), and the output of each stage is defined as the input of the next stage. In [35], The SGG model *Factorizable Network* is proposed, which introduces the subgraph to reduce the cost of SGG. In the TSM, the SGG model *Factorizable Network* is applied to generate a scene graph from one image.

C. SEMANTIC MAP

Much of the previous research on environment representation is carried for navigation tasks. The robot's navigation tasks are generally divided into global navigation and local navigation. Some studies construct a hierarchical map for global navigation and local navigation. In [36], a hybrid metric-topological-semantic map structure, called MTS-map, is established, which allows robots to implement fine metric-based navigation and coarse query-based localization. Although grid-based representation supports most of the navigation tasks, a large amount of calculation is needed to obtain the optimal path on the grid map of a large scene. To reduce the cost of calculation, a two-layer map is proposed

in [37]. In this article, the first layer is a region roadmap for representing the connectivity between different regions in the environment, and the second layer is the local roadmap. Each node in the region roadmap is related to a local roadmap. With this map, there will be less cost for finding the navigation route.

Understanding the environment with the human mindset, robots are able to implement some human-robotic interaction tasks. For planning and exploration in open and uncertain worlds, a semantic map is constructed with the commonsense knowledge [38]. The semantic representation describes the environment information perceived by the robot in detail and deal with uncertainties. In [39], uncertainty is also considered in the process of semantic map construction. Multiple semantic maps are constructed with probability. Conditional Random Fields (CRFs) are used to model the background relations and uncertainties during object recognition. Expressing the environment with 3D information preserved supports the scene graph to record the environment detail, but it requires powerful computation capabilities. In [13], a framework for constructing a 3D scene semantic map is proposed. The map constructed by this framework is composed of four layers, which is more in line with human thinking and perception. In [12], the 3D environment is expressed with a scene graph. Among the scene graph, each node represents the object and attributes, and each edge represents the relation between the objects. Based on [12], [13], MIT SPARK laboratory combines with the previous semantic mapping work, visual-inertial odometry, deep learning, and other methods to construct a scene graph of a dynamic 3D environment [11]. They propose a more comprehensive 3D semantic SGG framework, which adds the detection and tracking modules for dynamic targets, thus some of the impacts of dynamic changes is eliminated. Since the scene graph constructed from a single image is not specific enough, the scene graph generated from multiple images may miss some objects or repeatedly detect some objects. We refer to the method mentioned in [11]–[13] to build the scene graph from RGB-D videos.

III. METHOD

In this section, we describe the method of constructing the TSM. The first stage is the environment exploration. The robot carrying laser and RGB-D sensors is placed in an unknown indoor environment to complete the full coverage exploration. During the exploration, the laser sensor is service for collecting laser data to generate the metric map, the RGB-D sensor is applied to record the environment exploration video. The second stage is the topological map construction. After completing the exploration of the environment, a navigation behavioral topological map based on the metric map is constructed. The third stage is the scene graph generation. When generating the scene graph, the video of environment exploration needs to be split into different slices and then classify these slices according to the room they describe. Finally, the TSM of the environment is obtained by attaching the generated scene graph to the room node and combining it with the topological map. Our TSM makes the metric map as the bottom layer, scene graph as the middle layer, and topological map as the top layer. The process of TSM construction is shown in Figure 2.

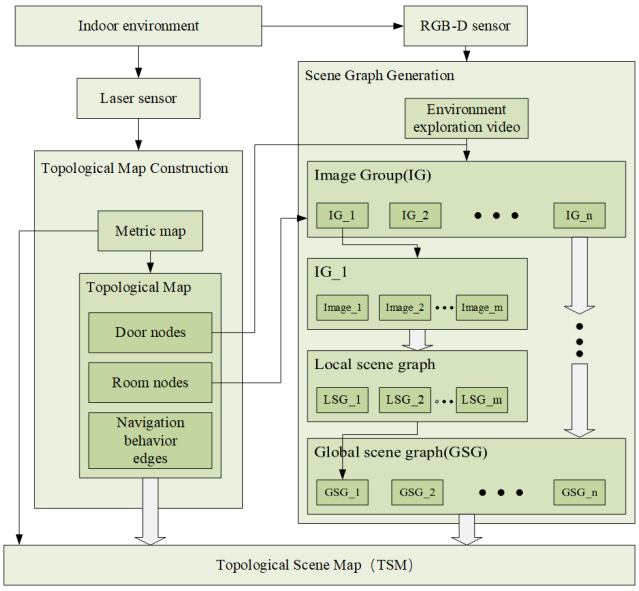

FIGURE 2. The TSM construction framework, the laser data are used to construct the metric map and the topological map, and RGB-D data are used to construct the scene graph. The door and room nodes of the topological map are applied to split the environment exploration video into slices. The video slices are grouped by the room where it was shot, and then the grouped slices are clipped into the image group. The images in the group are filtered by modules of video processing. Take IG_1 as an example, all the images in IG_1 are used to generate local scene graphs. All the local scene graphs from the same image group are applied to merge and update a global scene graph. Finally, the metric map, topological map, and scene graph are fused to construct TSM.

A. TOPOLOGICAL MAP CONSTRUCTION

Generally, the topological nodes indicate positions and edges present connectivity and distance. Although this kind of topological map meets the needs of most tasks, it is not enough for human-robotic interaction tasks. When asking the robot for directions, the robot needs to answer in the way that humans understand, rather than providing a simple node sequence. Here, we attach semantic descriptions to the nodes and edges based on the common topological maps.

1) TOPOLOGICAL MAP DESIGN

When constructing the topological map, we refer to the method adopted by [6]. A node in the topological map represents a location, and an edge represents the navigation behavior and distance. The navigation behavior helps to guide the robot from one location to another, such as walking through the corridor, leaving the room, and other navigation behaviors [14].

We define behaviors for the navigation behavior topological map, include enter the room (ER), leave the room (LR), and cross the room (CR). Although there are only three

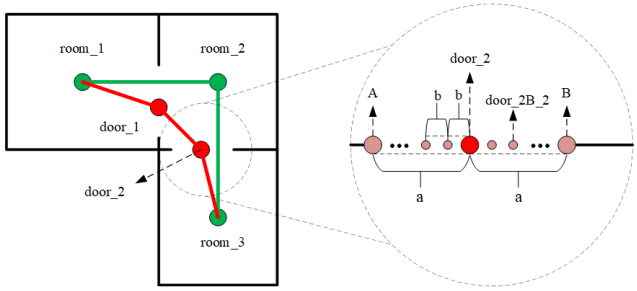

FIGURE 3. Navigation routes with or without door node and a example of door node generation.

behaviors in the behavior set, the set contributes to achieving semantic navigation in our indoor environments and also allows us to simplify the design of the topological map. For the "cross the room (CR)" behavior, we define the behavior as transforming from a door to another door of the same room. For example, the navigation behavior is specifically expressed as "from *door_1*, cross the *room_1*, to *door_2*". For the "leave the room (LR)" behavior, we define the behavior as transforming from a room to an exit(door) of the room, such as "from *room_1*, leave the *room_1*, to *door_1*". For the "enter the room (ER)" behavior, we define the behavior as entering from an exit(door) into a room, such as "from *door_1*, enter the *room_1*, to *room_1*".

There are two kinds of node in the topological map, the room node and the door node. Each room node is related to a specific space, such as a kitchen and a corridor. Each door node is related to an exit of one room. The door nodes are employed to improve navigation routes and help understand navigation behaviors. Without the door node, the room nodes along the path are visited when navigating from source to target, and there will be more costs. As shown in the left of Figure 3, an order of navigating from *room_1* to *room_3* is issued. If there are no door nodes, the navigation route will include the *room_2* node, and the navigation route is shown as the green line. If there is a door node, the navigation route will include the door nodes of *room_2* and will not include the *room_2* node. The navigation route is shown as the red line. Comparing the two navigation routes, we know that the employment of door nodes is conducive to generate shorter navigation routes. The cost of each edge is determined by the distance between adjacent nodes calculated by A star algorithm [40].

The process of *door_2* node generation will work as an example to illustrate the process of door node generation. The generation process is as follows:

(1) Find the region of *door_2* in the metric map, as shown in the right of Figure 3.
(2) Select the two points A and B on the two sides of the door.
(3) Make the midpoint of points A and B as *door_2* node and record its position.
(4) Obtain a point group by taking points in order by every distance b in the direction of A (or B) (the distance between the last point in a direction and the endpoint A (or B) is less than b), and record the position and label of these points, such as the second point taken in the direction of B is labeled as *door_2B_2*.
(5) Store all points between A and B in a dictionary with *door_2* as the key. In the topological map, only the *door_2* node is displayed, and the dictionary related to *door_2* will be used to split the video of environment exploration.

2) TOPOLOGICAL MAP CONSTRUCTION PROCESS

The process of the construction is given below:

(1) Construct the metric map of the environment with the Gmapping algorithm [41]. The metric map is saved in the form of the occupancy grid.
(2) According to previous requirements, some locations in the metric map are selected as nodes in the topological map, and the coordinate of the location is stored as the node features. There are two types of nodes, namely door nodes and room nodes.
(3) Attach semantic labels to each node, such as kitchen-1, corridor-1, etc.
(4) Define the navigation behavior between nodes in the topological map and attach the navigation behavior to all edges.
(5) Calculate the distance between nodes with the A star algorithm and preserve the distance as the edge features.
(6) Storage the topological map.

B. SCENE GRAPH GENERATION

The scene graph is defined as a directed acyclic graph: $G = (O, E)$, where $O = \{o_1, \ldots, o_n\}$ defined as objects set, $E \subseteq O \times \mathcal{R} \times O$ defined as edges set of relations. For each object $o_i = (c_i, A_i)$, it includes label $c_i \in \mathcal{C}$ of class and attributes $A_i \subseteq \mathcal{A}$ of object. We apply the Factorizable Network [35] to generate the scene graph from each image captured in the environment exploration video.

Factorizable Network represents the scene graph as a connection graph based on subgraphs during the inference process to improve the effectiveness of SGG. The subgraphs are generated by clustering to represent a set of relations with similar features, which simplifies the calculation of SGG. The Spatial-weighted Message Passing (SMP) structure and Spatial-sensitive Relation Inference (SRI) module of Factorizable Network reserved spatial structure for relational reasoning.

The process of SGG with Factorizable Network is summarized as follows:

(1) Region Proposal Network (RPN) is used to generate object region proposals.
(2) A fully connected graph is established for all object region proposals, in which any two objects have two edges in different directions that represent relations

between them. The features of these edges are extracted by the union box of the two connected objects.

(3) All relations are clustered from the bottom up, and relations with the similar features are clustered together. After a relation class is obtained, all the relations in the same class are represented by the same subgraph node. Thus, a subgraph-based representation of the fully connected graph is obtained, and it includes subgraph and object nodes.

(4) Feature vectors and 2D feature maps are generated by employing Region Of Interest (ROI) pooling to object feature and subgraph features respectively.

(5) Refined object and subgraph features are generated with the use of spatial-weight message passing.

(6) Object classes and their relations are recognized by the object features and fusion of objects and subgraph features respectively.

For features, object features focus on the detail of an object, while subgraph features focus on the relation between objects. The representation combined with the features of these two types of nodes is beneficial to recognize object and relation classes.

In step (5) of SGG, SMP based on the inner product attention mechanism is applied to combine the object and subgraph features to obtain the representation of refined features. 2D feature map is employed to express subgraph features, so the spatial information is retained. Assume that the object feature vector and subgraph feature map input to SMP is $\mathbf{o}_i$ and $\mathbf{S_k}$ respectively. Since the dimensions of object and subgraph features are different, two different methods are needed to exchange information between the object and subgraph nodes.

From subgraph to object nodes The purpose of this process is to convert the 2D subgraph feature map to vector space of the object feature and fuse these two types of features. Firstly, $\mathbf{S_k}$ is directly converted into $\mathbf{s_k}$ through 2D average pooling. Then, all $\mathbf{s_k}$ are aggregated by weights. The weighted sum $\tilde{\mathbf{s}}_i$ of $\mathbf{s_k}$ is computed as follows:

$$\tilde{\mathbf{s}}_i = \sum_{\mathbf{S_k} \in \mathbb{S}_i} p_i\left(\mathbf{S}_k\right) \cdot \mathbf{s}_k \tag{1}$$

$$p_i\left(\mathbf{S}_k\right) = \frac{\exp\left(\mathbf{o}_i \cdot \mathrm{FC}^{(att\_s)}\left(\mathrm{ReLU}\left(\mathbf{s}_k\right)\right)\right)}{\sum_{\mathbf{S_k} \in \mathbb{C}_i} \exp\left(\mathbf{o}_i \cdot \mathbf{FC}^{(att\_s)}\left(\mathrm{ReLU}\left(\mathbf{s}_k\right)\right)\right)} \tag{2}$$

where $\mathbb{S}_i$ represents a set composed of subgraph nodes connected to object i. $p_i\left(\mathbf{S}_k\right)$ indicates the weight for $\mathbf{s_k}$. $\mathbf{FC}^{(att\_s)}$ convert the feature vector $\mathbf{s_k}$ to the domain of $\mathbf{o}_i$. Finally, refined object features $\hat{\mathbf{o}}_i$ are generated by combining the weighted sum $\tilde{\mathbf{s}}_i$ with the object feature $\mathbf{o}_i$.

$$\hat{\mathbf{o}}_i = \mathbf{o}_i + \mathrm{FC}^{(s \to o)}\left(\mathrm{ReLU}\left(\tilde{\mathbf{s}}_i\right)\right) \tag{3}$$

where $\mathrm{FC}^{(s \to o)}$ represents a fully connected network for converting $\tilde{\mathbf{s}}_i$ to the domain of $\mathbf{o}_i$.

From object to subgraph nodes The process aims to get the weighted sum of object features and map these features to the domain of the subgraph feature map to combined with the 2-D feature map. When converting the object feature vector to the domain of the subgraph feature map, the position information of the object needs to be considered. The weighted sum $\tilde{\mathbf{O}}_k(x, y)$ at location (x, y) after object feature projection is calculated as follows:

$$\tilde{\mathbf{O}}_k(x, y) = \sum_{\mathbf{o_i} \in \mathbb{O}_k} \mathbf{P}_k\left(\mathbf{o}_i\right)(x, y) \cdot \mathbf{o}_i \tag{4}$$

$$\mathbf{P}_k\left(\mathbf{o}_i\right)(x, y) = \frac{\exp\left(\mathrm{FC}^{(att\_o)}\left(\mathrm{ReLU}\left(\mathbf{o}_i\right)\right) \cdot \mathbf{S}_k(x, y)\right)}{\sum_{\mathbf{S_k} \in \mathbb{C}_i} \exp\left(\mathrm{FC}^{(att\_o)}\left(\mathrm{ReLU}\left(\mathbf{o}_i\right)\right) \cdot \mathbf{S}_k(x, y)\right)} \tag{5}$$

where $\mathbb{O}_k$ represents the set of object nodes connected to subgraph k. $\mathbf{P}_k\left(\mathbf{o}_i\right)(x, y)$ represents the weight of the $\mathbf{o}_i$ at location (x, y) of subgraph k. $\mathrm{FC}^{(att\_o)}$ convert $\mathbf{o}_i$ to the domain of $\mathbf{S}_k(x, y)$. Then, refined subgraph feature $\hat{\mathbf{S}}_k$ is abtained by combining $\tilde{\mathbf{O}}_k$ and $\mathbf{S}_k$.

$$\hat{\mathbf{S}}_k = \mathbf{S}_k + \mathrm{Conv}^{(o \to s)}\left(\mathrm{ReLU}\left(\tilde{\mathbf{O}}_k\right)\right) \tag{6}$$

where $\mathrm{Conv}^{(o \to s)}$ is a convolution layer that transforms the object features into the subgraph domain.

In step (6) of SGG, refined object and subgraph features are used to recognize the classes of objects and relations. The object classes are directly predicted by the object features. The classes of relations are predicted by the fusion of subject and object features and the corresponding subgraph feature.

$$\mathbf{p}^{\langle i,k,j \rangle} = \mathbf{f}\left(\mathbf{o}_i, \mathbf{S}_k, \mathbf{o}_j\right) \tag{7}$$

Each object connected with subgraph node is relate to a region in subgraph feature map, so the object feature is applied as convolution kernel to extract visual cue in subgraph feature map. The convolution result $\mathbf{S}_k^{(i)}$ is calculated as:

$$\mathbf{S}_k^{(i)} = \mathrm{FC}\left(\mathrm{ReLU}\left(\mathbf{o}_i\right)\right) \otimes \mathrm{ReLU}\left(\mathbf{S}_k\right) \tag{8}$$

Then, the relation is predicted with the use of a fully-connected layer on the fusion convolution result $\mathbf{S}_k^{(i)}$, $\mathbf{S}_k^{(j)}$, and subgraph feature map $\mathbf{S}_k$.

$$\mathbf{p}^{(i,k,j)} = \mathrm{FC}^{(p)}\left(\mathrm{ReLU}\left(\left[\mathbf{S}_k^{(i)}; \mathbf{S}_k; \mathbf{S}_k^{(j)}\right]\right)\right) \tag{9}$$

C. VIDEO PROCESSING FOR SGG

The images for SGG come from the video taken during the environment exploration. To obtain all the scene information in the space for a certain topological node, the scene graph generated from the images taken from the same room needs to be merged. In this process, it is necessary to select appropriate images to generate the scene graph and eliminate duplicate elements when merging multiple scene graphs. The frames extracted from the video need to be processed by the region proposal network(RPN). The RPN extracts the regions of objects from the image. The SGG module takes the output of the RPN module as input and calculates the probabilities that the object in the proposed region belongs to different classes and the probabilities that the relation between objects belongs to different classes.

There are three modules for processing the video for SGG, Adaptive Blurry Image Rejection (ABIR), Keyframe Group Extraction (KGE), and Spurious Detection Rejection (SDR). We employ the local scene graph and the global scene graph to distinguish scene graphs generated from a single image and images describing the same room.

1) ADAPTIVE BLURRY IMAGE REJECTION

For better performance, the object and relation recognition modules need to input clear images. But some blurry images may be collected due to the movement of the camera in the process of images collection. In blurry images, the shape, size, and color of objects may change, which will harm objects and relations recognition. To eliminate the influence of blurry images, the variance of Laplacian is used to measure the intensity variations between pixels in an image:

$$V = \frac{1}{WH}\sum_{x=1}^{W}\sum_{y=1}^{H}L(x,y)^2 - \left\{\frac{1}{WH}\sum_{x=1}^{W}\sum_{y=1}^{H}L(x,y)\right\}^2 \tag{10}$$

$$L(x,y) = \left(\partial^2 I/\partial x^2\right) + \left(\partial^2 I/\partial y^2\right) \tag{11}$$

where W, H represent the width and height of the image respectively. $L(x, y)$ is the Laplacian operator. However, some low texture images may be filtered as blurry images, as the intensities of low texture images are also not significantly changed.

To overcome the problem of texture, the ABIR algorithm is adopted. Over the Laplacian variances, ABIR evaluates the exponential moving average (EMA):

$$S_t = \begin{cases} V_t & t = 1 \\ \alpha \cdot S_{t-1} + (1-\alpha)\cdot V_t & t > 1 \end{cases} \tag{12}$$

where t represents time step, V_t is the variance of Laplacian at t, and α is a constant smoothing factor in the interval [0, 1), which represents the influence size of previous observations on the current S_t. The initial EMA does not follow the observed values as the few previous observations, which will produce some deviation in the final result. To correct this deviation, we process the final S_t:

$$S_t' = \frac{S_t}{1-\alpha^t} \tag{13}$$

S_t' is the bias-corrected average value, which is further processed to obtain the adaptive threshold:

$$t_{\text{blurry}} = g \cdot \ln\left(1 + S_t'\right) + b \tag{14}$$

where gain and offset correspond g and b, respectively.

2) KEYFRAME GROUP EXTRACTION

The process of the KGE module is divided into three steps: (1) Accepting a series of processed images. (2) Filtering out unnecessary frames by dividing the input image into three parts. (3) Forming a keyframe group. Note that the input images are divided into the following three parts: leftmargin=*,labelsep=5.5mm

- Keyframe: The first anchor frame and the coverage of the keyframe group is determined with keyframe as reference.
- Anchor Frame: Apart from the latest anchor frame, all other anchor frames are inactive. The next anchor frame is determined by the active anchor frame.
- Garbage Frame: Except for the keyframes and anchor frames, all the other frames are garbage frames, which are regarded as redundant frames and discarded.

Specifically, the process of the KGE module is as follows: This module defines the first keyframe by the first nonblurry frame. Other frames need to be classified and each incoming frame needs to be compared with the current keyframe and the active anchor frame. When the overlap between the frame and the active anchor frame is lower than t_{anchor} %, the frame is reserved as the next anchor frame. When extracting the first anchor frame, the input frame needs to compare with the keyframe. When a new anchor frame is detected, the current active anchor frame will turn into inactive, and the new anchor frame will become active. If the overlaps value of an incoming frame and the keyframe is lower than t_{keyframe} %, the frame will become the new keyframe, and the previous keyframe and anchor frames will form the keyframe group.

To compute the overlap between two frames, one frame is mapped to the coordinate of the other:

$$\text{overlap} = \frac{1}{W \cdot H}\sum_{x}\sum_{y}\overrightarrow{\mathbf{1}}_{I_{W,H}}\left(p'(i,j)\right) \tag{15}$$

$$I_{W,H} = \left\{(x,y) \mid 0 \le x < W, 0 \le y < H, (x,y) \in \mathbb{Z}^2\right\} \tag{16}$$

where $\overrightarrow{\mathbf{1}}_A(\cdot)$ is an indicator function for set A. Projection function $p'(i, j)$ project source frame to target frame. It is defined as:

$$p' = K \cdot T_{i,j} \cdot D(p) \cdot K^{-1} \cdot p \tag{17}$$

where K is intrinsic to the camera, $T_{i,j}$ are the relative poses between i frame and j frame, p is original point, $D(p)$ means point p depth.

With the application of keyframes and anchor frames, the KGE module effectively removes the redundant information in continuous image sequences. Besides, even if the camera still for a long time, the module effectively handles redundant frames.

3) SPURIOUS DETECTION REJECTION

During generating scene graphs from images clipping from video, the recognition module cannot perform perfectly as the frames captured by the camera are affected by noise. The SDR module aims to eliminate errors and repeated detections with prior knowledge. The SDR module is applied to multiple modules: region proposal, object recognition, and relation extraction modules. First, the SDR module deletes

the redundant target area in the region proposal module with the use of non-maximum suppression (NMS) [42]. For the object recognition module, the SDR module deletes irrelevant objects. Besides, the SDR module deletes predefined irrelevant objects such as roads, sky, buildings, and moving objects.

For the relation extraction module, the SDR module counts the possible relations of all object pairs in all frames from one frame group and keeps the most frequent relations in the scene graph. If multiple relations appear the same number of times for an object pair, they will all be added to the graph. Then, the module SDR employ a relation dictionary, which is extracted from the statistical information of the visual genome dataset, as prior knowledge. The relation dictionary includes the statistical data between object pairs, and the pixel distance d_{pixel} stored in the form of Gaussian distribution.

As to recognizing the relation between object pairs, the process is shown as follows: (1) the SDR module detects the related object pairs. (2) Searching the relation dictionary for the detected object pairs. (3) calculating the probability of the detected relation. To calculate the probability, the pixel distance between objects and prior statistical information is applied to filter out relations with the probability lower than the threshold. The probability of relation $\Pr\left(r \mid d_{\text{pixel}}\right)$ is calculated as follows:

$$\Pr(r \mid d_{\text{pixel}}) = \Pr_{\text{dict}}\left(r \mid d_{\text{pixel}}\right) \cdot \phi_{\mu,\sigma^2}\left(d_{\text{pixel}}\right) / \phi_{\mu,\sigma^2}(\mu) \tag{18}$$

$$\phi_{\mu,\sigma^2}(k) = (1/\sqrt{2\pi\sigma^2}) \exp\left(-[1/2]([k-\mu]/\sigma)^2\right) \tag{19}$$

where $\Pr_{\text{dict}}\left(r \mid d_{\text{pixel}}\right)$ mean the statistical probability given pixel distance. $\phi_{\mu,\sigma^2}(k)$ is Gaussian function. The normalized probability density function is employed to approximate the probability of distance between points.

4) LOCAL SCENE GRAPH

There will be a temporary ID in the local scene graph and a permanent ID in the global scene graph for the same object. With the recognition module, the semantic label of the objects and probability scores of the label is obtained easily. The top-k labels and the scores are kept for the same node detection. Besides, to eliminate the measurement error caused by representing the object position with the center point, the object position is represented in the form of the Gaussian distribution. After dividing the object region into 5×5 sub-regions, the center rectangle is cut from the object bounding box given by the region proposal module. Then, the 3D position of each point in the center rectangle relative to the first keyframe is calculated by

$$p'' = T_{i,o} \cdot D(p) \cdot K^{-1} \cdot p \tag{20}$$

where i and o mean the indices of the current frame and first keyframe. We then evaluate the mean and variance of the Gaussian distribution $\mathcal{N}\left(\mu, \sigma^2\right)$ of the 3D position. Each dimension x, y, and z are assumed to be independent and identically distributed, and the points number is reserved for evaluation.

The color histogram of the object is got by

$$h_{H,S,V} = N \cdot \Pr(H = h, S = s, V = v) \tag{21}$$

where (H, S, V) represent the three axes of the color space, and N represents the number of pixels. Each axis of the color space is divide into c-bins, that is, the size of the histogram is c^3. In the end, a thumbnail of the object for the region in the bounding box is obtained.

The attributes extracted in this step will be updated and modified by subsequent modules, which collect object information from multiple frames and then make the final decision.

5) GLOBAL SCENE GRAPH

The updating and merging of the scene graph will integrate the local scene graph generated from a single image into the global scene graph. Richer scene information will be collected with the changes of camera position and perspective, and the recognition module extracts different features from the same object. To integrate different features and eliminate the repeated extraction, we propose a module for same node detection.

When adding nodes to the global scene graph, the same node detection needs to be performed. In the same node detection, the following features are employed, object label, 3D position, and color histogram. The similarities of these features between the newly added node and the previous node are calculated, respectively. The similarity scores of each feature are calculated as follows:

For label similarity, the label similarity is defined as s_{label}, which is calculated as

$$s_{\text{label}} = \begin{cases} |C_o \cap C_c| \cdot \text{ score} & |C_o \cap C_c| > 0 \\ \{1 - d_{wv}\left(f_{wv}\left(l_o\right), f_{wv}\left(l_c\right)\right)\} & \\ \cdot \text{ score} & \text{otherwise} \end{cases} \tag{22}$$

where o and c respectively represent the original node in the global scene graph and the candidate node in the current frame. C_o and C_c contain top-k object prediction category labels. l_o and l_c represent the label with the highest score. The number of common elements in C_o and C_c is multiplied by the score. If there are no common elements, the scores are related to the distance between the word vectors of l_o and l_c. The score is calculated as follows:

$$\text{score} = \max_{i \in \{o,c\}} \left\{f_{s_i}(l) : l \in C_i\right\} \tag{23}$$

where the input is a label in candidate set C_i, and the score function f_{s_i} returns the score of the label.

For position similarity, we define position similarity as s_{position}, which is calculated as

$$s_{\text{position}} = \prod_{j \in \{x,y,z\}} I^j\left(\mu_c^j\right) \tag{24}$$

where μ_c represents the mean of an object position in the candidate set. The similarity of the position information in the x, y, and z directions is calculated by

$$I^j = \begin{cases} 1 & \left|\mu_c^j - \mu_o^j\right| < \sigma_o^j \\ \frac{1-\phi\left(Z_{\mu_c^j}\right)+\phi\left(-Z_{\mu_c^j}\right)}{1-\phi\left(Z_{\sigma_o^j}\right)+\phi\left(-Z_{\sigma_o^j}\right)} & \text{otherwise} \end{cases} \tag{25}$$

where Z is the z-score of a normal distribution, $\phi(\cdot)$ output the area of standard normal distribution. If the difference between the position of the candidate object and the position of the object in the global scene graph is less than σ_o, the position similarity is considered to be 1. Otherwise, the position similarity is inversely proportional to the distance.

For color similarity, the color similarity is defined as s_{color} accoding to another color similarity:

$$d_h(h^i, h^j) = \frac{\sum_X \sum_Y \sum_Z \min\left(h^i(x,y,z), h^j(x,y,z)\right)}{\min\left(\left|h^i\right|, \left|h^j\right|\right)} \tag{26}$$

which is calculated with the intersection of histograms. h^i and h^j are the two histograms for comparison. X, Y, and Z is the axis of 3-D space. $|\cdot|$ return the magnitude of a histogram. When measuring the distance between histograms, the intersection of the histograms ensures the efficient calculation and effective comparison of the color histograms. The final color similarity is:

$$s_{\text{color}} = 1 - d_h\left(h^o_{H,S,V}, h^c_{H,S,V}\right) \tag{27}$$

Finally, the above similarities are combined to get the total similarity:

$$s_{\text{total}} = \sum_{i \in F} w_i \cdot s_i \tag{28}$$

where $F = [\text{label}, \text{position}, \text{color}]$. When the similarity between the candidate node and a node in the global scene graph is greater than the threshold, these nodes are considered to be the same.

The process of merging and updating for global scene graph generation is as follows. The global scene graph is initialized by the first keyframe. With the first frame inputted, the local scene graph is generated and then merge with the global scene graph. During the merging process, the nodes generated in the local scene graph are compared with the nodes in the global scene graph, and the same nodes will be deleted. Only the nodes that do not appear in the global scene graph are added to the global scene graph. During the updating process, the label with the highest score in the top-k label set $C_o \cup C_c$ used for the same node detection is selected as the latest label for one node. The 3D position of the object will also be updated according to the latest observation information. The color histogram is combined with the incoming color histograms. The number of points that remain in the node becomes the sum of the original and new number of points. If the label with the highest score comes from the incoming scene graph, the thumbnail will be replaced by the incoming thumbnail.

D. FUSION OF TOPOLOGICAL MAP AND SCENE GRAPH

To obtain the scene graph of each room, the video obtained by the robot during the full coverage exploration needs to be split. During the environment exploration, the robot records the video of the environment and record the position of the robot and the timestamp in the video every n seconds. These position information and video timestamps are reserved for splitting the exploration video.

The topological map is constructed based on the metric map, in which the door nodes describe the connectivity between two rooms. The exploration video will be split with the positions and timestamp recorded during environmental exploration and the door nodes of the topological map. While splitting the video, the door dictionary containing point group created during door node generation is applied to judge whether the robot passes the door. After splitting the video, we get the image group for each room easily and then construct the scene graph of the room with the image group. Every room node has its scene graph, thus the fusion of the topological map and the scene graph is obtained.

The process of splitting the video is as follows:

(1) Choose a door node and take its door dictionary with position information of the point group.
(2) Select all positions whose distance to any point in the point group is less than the threshold $D_{threshold}$ from the positions recorded during environmental exploration.
(3) Obtain the timestamps in the video for the selected positions and record these timestamps for splitting video.
(4) Repeat steps (1) (2) (3) until all the timestamps are obtained by all door nodes.
(5) Split the environment exploration video according to the timestamps recorded in (4).
(6) According to the connectivity between the door node and the room node in the topological map, the split video is divided into slices of the rooms.

IV. EXPERIMENTS

In this section, we constructed a topological map in a simulation environment, verified the effectiveness of generating the scene graph from the video, and given a TSM instance derived from the simulated indoor environment and scannet data set. The compositions of TSM are vividly illustrated in Figure 4.

A. EXPERIMENTAL SETUP

The turtlebot3 [43] with the Robot Operating System (ROS) is applied [44] in the simulation environment (Gazebo and Rviz) to construct a topological map of TSM. Figure 5 presents the indoor environment model in Gazebo and Rviz. The Gmapping [41] package is employed to build the metric map of the environment from data collected by a laser sensor.

The Factorizable Network is trained on the Visual Genome [45] dataset to obtain an effective SGG model. The Visual Genome dataset connects images with semantic concepts, which include 108,077 images with semantic

FIGURE 4. Constructed map: (a) The metric map, above the figure, is built with the collected 2D laser scan in the form of an occupancy grid map. (b) The topological map, below the figure, is constructed with manual assistance, nodes and edges are attached with semantic information. Each room node has a scene graph.

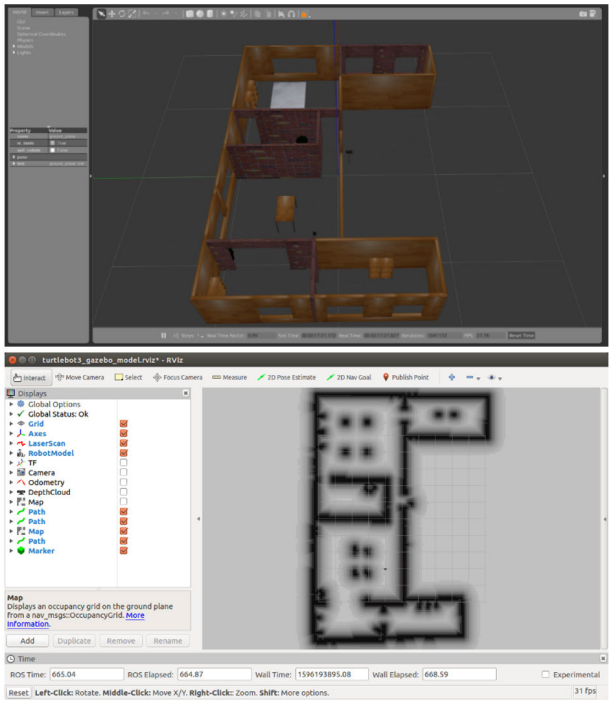

FIGURE 5. Simulation model: (a) Gazebo(top) is a 3D dynamic simulator that is able to accurately and effectively simulate in complex indoor and outdoor environments. The indoor simulation environment is loaded in gazebo. (b) Rviz(bottom) is a 3D visualization tool officially provided by ROS. Almost all robot-related data we use is displayed in Rviz.

annotations, like objects, relations, attributes, etc. When training the model, what we need mainly includes the attributes of objects in images, the types of objects, and the relations between the objects.

To verify the effectiveness of our TSM construction framework in a simulation environment, some videos from the indoor scene video dataset ScanNet [46] are selected as the videos of environment exploration. This dataset consists of 1513 sequences, which are collected by RGB-D cameras. Among them, the resolution of the frames is 1296 × 968 (color) and 640 × 480 (depth), and the frequency of image collection is 30Hz.

B. TOPOLOGICAL MAP CONSTRUCTION

The construction of the topological map is carried out in a simulation environment. Above all, an indoor environment model is loaded, as shown in the top of Figure 5. The robot is controlled by the keyboard to explore the environment and complete the full coverage exploration of the environment quickly. The metric map generated during the exploration is viewed through Rviz, as shown in the bottom of Figure 5. The metric map is presented at the top of Figure 4. Then, the topological map is constructed based on the metric map. We select the topological nodes on the map and define semantic labels for them. These labels are divided into room nodes and door nodes. As shown in the bottom of Figure 4, 15 topological nodes are selected for this environment, including 7 door nodes and 8 room nodes.

TABLE 1. Label and Scannet sequence for each room.

Room ID	Room Label	Sequence in Scannet	Num of frame
room_1	kitchen_1	scene0000_00	5578
room_2	dining_room_1	scene0010_00	2513
room_3	living_room_1	scene0060_00	1187
room_4	corridor_1	scene0070_00	1322
room_5	outdoor_1	scene0080_00	1124
room_6	bedroom_1	scene0090_00	534
room_7	living_room_2	scene0100_00	1015
room_8	bedroom_2	scene0110_00	1155

The label of each room we defined is listed in Table 1. Finally, the edges are defined according to the spatial connectivity. To get the cost of each edge, the distance between connected topological node is obtained by the A star algorithm according to the calculation rules of the actual navigation route of the robot. Combined with the navigation behavior generation rules, the navigation behavior for each edge is generated. The navigation behavioral topological map is visualized as Figure 6

When the robot navigates with the use of the behavioral topological map, it is able to generate a sequence of commands for the robot to execute and generate a recommended route for humans. For example, the robot is placed in *room_3* and get a command, like go to *room_8*. The robot obtain a topological node sequence for robot navigating, like "*room_3*, *door_3*, *door_5*, *door_7*, *room_8*". By inquiring about the semantic information of the

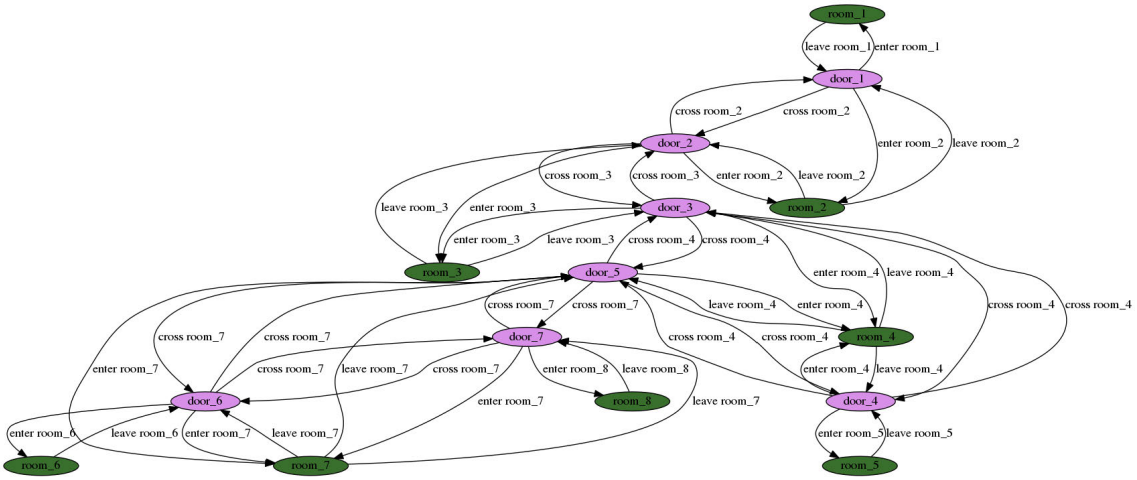

FIGURE 6. The navigation behavioral topological map.

topological map, the robot generate a recommended route for humans, like: (leave living_room_1, cross *corridor_1*, cross *living_room_2*, enter *bedroom_2*).

C. SCENE GRAPH GENERATION

The scene graph is generated with video slices classified by rooms. The SGG model Factorizable Network needs to be trained, and then modify the parameters of other modules for generating scene graphs from the video.

The SGG model Factorizable Network is trained on the Visual Genome dataset. The compute with Intel Core i7-9750H CPU@2.60 GHz×12 and GPU RTX 2060 is employed for the experiment. Based on the pre-trained Factorizable Network [35], the final SGG model used in the experiment achieves 29.574% of Recall@50 and 38.476% of Recall@100 on the Visual Genome dataset.

With the trained Factorizable Network, the scene graph is generated for each image from the exploration video. In the process of capturing images from the video, the ABIR module is applied to eliminate the effects of blurred images, and the key parameters α, g, and b are designed as 0.9, 30, and 25. After removing a host of blurred images, the KGE module is employed to extract the keyframe group from the remaining images. To reduce the budget of computing overlaps between frames, the source image is mapped to the target image and 1000 points are sampled for calculating. For eliminating the influence of uncommon objects in the indoor environment during object recognition, the SDR module ignores 68 objects out of 400 objects in the recognition process. The probability threshold of SDR is set to 0.5 to remove the relation with great uncertainty. When building color histograms, each direction is divided into 8 bins, and the size of the histogram is 512. During performing the same node detection, w_{label},w_{color},$w_{position}$ is set to 0.375, 0.25, and 0.375, respectively, and set the same node detection threshold to 0.5. In Figure 7, there is an instance of generating a scene graph from the keyframes in the captured image group. With the continuous input of frames, the global scene graph becomes more complete. With the representation of the scene graph, objects in the environment and relations between the objects are clearly displayed.

According to the previously topological map, eight sequences are selected from the ScanNet dataset as videos obtained in eight different rooms, as shown in Table 1. We conducted some comparative experiments to verify the effect of the object detection module, the KGE module, and the same node detection module with the use of scene0010_00. Table 2 presents the results obtained from comparative experiments. The first experiment serves as a reference for comparison. The same node detection threshold, anchor frame threshold, and object detection threshold are adjusted, and the anchor frame number, node num, and total time are compared. From Table 2, we know that the larger the threshold of the same node detection, the fewer nodes are judged as the same node, and more nodes are finally obtained. At the same time, the more nodes to be processed, the more time it takes. The number of anchor frames is adjusted by the anchor frame threshold. If the overlap between a frame and the active anchor frame is less than the anchor frame threshold, this frame is judged as a new active anchor frame. It can be seen that the larger the anchor frame threshold, the larger the anchor frames number, nodes number, and total times. The object detection threshold adjust the number of objects detected in the image, that is, the larger the threshold, the fewer objects are detected and the less total time it takes.

The results of SSG from the video indicate that the scene graph describes the environment in the form of JSON with less memory. As the accuracy of the object detector and the SGG module increases, the robot will also establish a more accurate environment model, which will improve the robot's intelligence. The ABIR, SDR, and KGE modules are effective to reduce the redundancy of the scene graph,

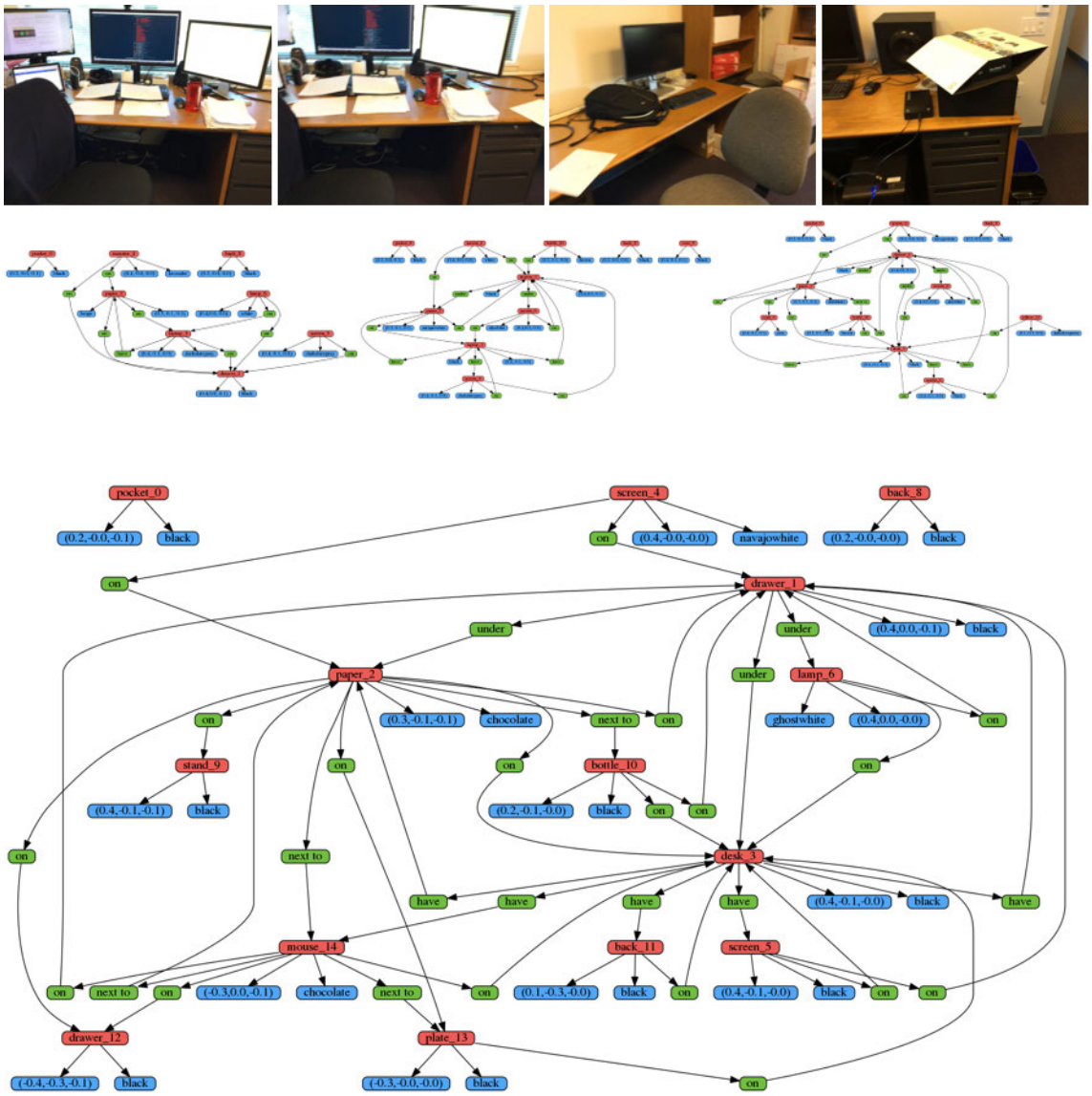

FIGURE 7. Example of generating scene graph from keyframe group.

TABLE 2. The influence of some modules on the global scene graph generation.

	object detection threshold	anchor frames threshold	same node threshold	anchor frames num	nodes num	total time(s)
1	0.25	0.68	0.5	159	17	1076.6
2	0.25	0.68	**0.2**	159	5	1064.5
3	0.25	0.68	**0.8127**	159	39	1103.2
4	0.25	**0.25**	0.5	66	13	932.3
5	0.25	**1**	0.5	2149	32	11439.2
6	**0.15**	0.68	0.5	159	42	1273
7	**0.35**	0.68	0.5	159	16	980.3

which is conducive to selecting clearer images from the image group to generate the scene graph and improves the accuracy of object detection. Since there are many blurred images in the image group, there is no quantitative evaluation of the generated scene graph. Through analysis, the performance of the SGG module is able to be improved from the following aspects. Firstly, training the object detector with appropriate images and objects. In our experiment, the object detector is trained on the Visual Genome dataset. The images of this dataset include both indoor scenes and outdoor scenes, and the 400 objects used in training may not always appear in indoor scenes. Thus, we need to set a reasonable object recognition threshold, and selectively ignore some objects to eliminate the influence of uncertain objects. Secondly, capturing clearer images. Images for the room is captured from the ScanNet video dataset. Although most of the blurred

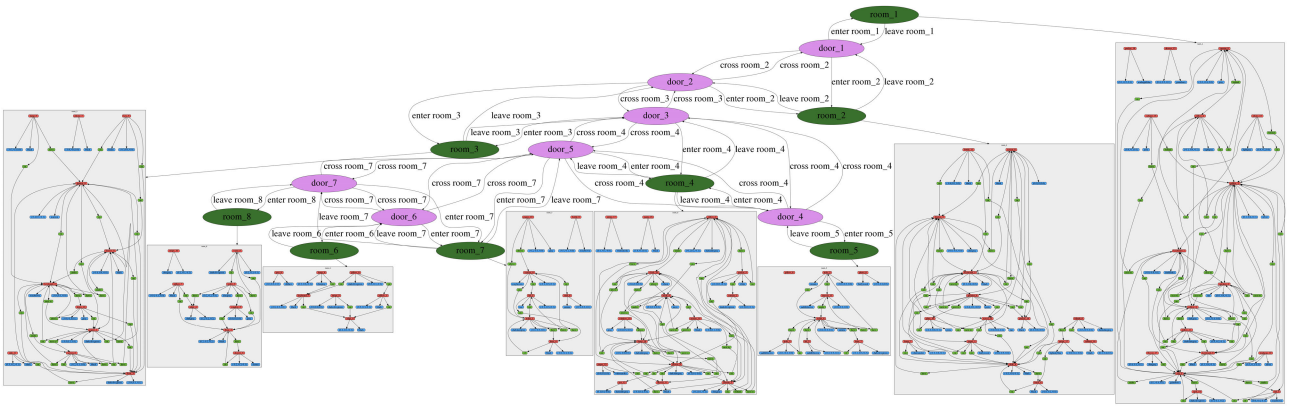

FIGURE 8. The fusion of topological map and scene graph.

images are eliminated through the ABIR and KGE module, images intercepted from the video are inevitably not clear enough, which also affects the accuracy of the detector.

D. FUSION OF TOPOLOGICAL MAP AND SCENE GRAPH

In the simulation experiment, we select some sequences from the ScanNet dataset as the video of each room. If the video for each room is needed to be split from the exploration video with the topological map, the threshold for judging the video split timestamp is required to be computed. For instance, the speed of turtlebot3 is $v = 0.2\ m/s$. The time interval for recording coordinate information during environment exploration is $n = 2\ s$. The distance between two adjacent points in the point group in the door node dictionary is $b = 0.4\ m$. The threshold is computed as follow:

$$D_{threshold} = \sqrt{b^2 + \left(\frac{nv}{2}\right)^2} \approx 0.45m \tag{29}$$

With the video slices of each room, the modules for generating a scene graph from a video are employed to get the scene graph of each room. The topological map and scene graphs are finally integrated into one JSON file in the form of a dictionary. The key of the dictionary is the node ID from the topological map and the values include the topological information and the scene graph information. The fusion of the topological map and scene graph in this experiment is shown in Figure 8, in which each room node in the topological map is related to a scene graph.

V. CONCLUSION AND FUTURE WORK

In this article, we propose a scene semantic map construction framework to build TSM. The TSM is a combined representation of the topological map and the scene graph for improving the robot's capability of understanding the environment intelligently. In general, the topological map based on navigation behavior enables the robot to efficiently and quickly generate a global navigation route with a semantic description, while the scene graph preserving objects and relations between objects makes the representation of the scene more specific. The purpose of the TSM is to record environmental information and assist the robot to realize interpretable reasoning for completing a multitude of human-robotic interaction tasks, such as question and answer.

The simulation experiments verify the effectiveness of the process for topological map construction and the various modules for generating scene maps from videos. These experiments suggest that TSM is capable of modeling the environment with the navigation behavioral topological map and scene graphs for providing semantic navigation routes and describing the details of scenes. However, the framework for constructing TSM is still immature. The TSM cannot be built in real-time, so related applications need to be based on the completed TSM, such as semantic question and answer, semantic search, etc. During the process of constructing TSM, the dynamic objects are not considered, which will disturb the generation of the global scene graph and even limit the use of the TSM construction framework. Since humans and animals are the main dynamic targets in the environment, we avoid detecting these objects by the SDR module to reduce the impact of dynamic targets in global SSG.

Future work is needed to expand the application fields of the TSM construction framework, which includes the detecting and tracking of dynamic target [11], real-time TSM construction, SGG with knowledge graph [33], and accuracy improvement for SSG [47], etc. Additionally, future work will be carried out, such as semantic navigation, semantic question and answer, and other human-robotic interaction tasks.

REFERENCES

[1] E. C. Tolman, "Cognitive maps in rats and men," *Psychol. Rev.*, vol. 55, no. 4, p. 189, 1948.
[2] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization," *IEEE Trans. Robot. Autom.*, vol. 17, no. 2, pp. 125–137, Apr. 2001.
[3] F. Blochliger, M. Fehr, M. Dymczyk, T. Schneider, and R. Siegwart, "Topomap: Topological mapping and navigation based on visual SLAM maps," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1–9.
[4] H. Oleynikova, Z. Taylor, R. Siegwart, and J. Nieto, "Sparse 3D topological graphs for micro-aerial vehicle planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1–9.

[5] F. Wang, Y. Liu, L. Xiao, C. Wu, and H. Chu, "Topological map construction based on region dynamic growing and map representation method," *Appl. Sci.*, vol. 9, no. 5, p. 816, Feb. 2019.
[6] K. Chen, J. Pablo de Vicente, G. Sepulveda, F. Xia, A. Soto, M. Vázquez, and S. Savarese, "A behavioral approach to visual navigation with graph localization networks," 2019, *arXiv:1903.00445*. [Online]. Available: http://arxiv.org/abs/1903.00445
[7] L. Mezghani, S. Sukhbaatar, A. Szlam, A. Joulin, and P. Bojanowski, "Learning to visually navigate in photorealistic environments without any supervision," 2020, *arXiv:2004.04954*. [Online]. Available: http://arxiv.org/abs/2004.04954
[8] D. S. Chaplot, R. Salakhutdinov, A. Gupta, and S. Gupta, "Neural topological slam for visual navigation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 12875–12884.
[9] A. Nüchter and J. Hertzberg, "Towards semantic maps for mobile robots," *Robot. Auto. Syst.*, vol. 56, no. 11, pp. 915–926, Nov. 2008.
[10] I. Kostavelis and A. Gasteratos, "Semantic mapping for mobile robotics tasks: A survey," *Robot. Auto. Syst.*, vol. 66, pp. 86–103, Apr. 2015.
[11] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone, "3D dynamic scene graphs: Actionable spatial perception with places, objects, and humans," 2020, *arXiv:2002.06289*. [Online]. Available: http://arxiv.org/abs/2002.06289
[12] U.-H. Kim, J.-M. Park, T.-J. Song, and J.-H. Kim, "3-D scene graph: A sparse and semantic representation of physical environments for intelligent agents," *IEEE Trans. Cybern.*, early access, Aug. 13, 2020, doi: 10.1109/TCYB.2019.2931042.
[13] I. Armeni, Z.-Y. He, A. Zamir, J. Gwak, J. Malik, M. Fischer, and S. Savarese, "3D scene graph: A structure for unified semantics, 3D space, and camera," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 5664–5673.
[14] G. Sepulveda, J. C. Niebles, and A. Soto, "A deep learning based behavioral approach to indoor autonomous navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 4646–4653.
[15] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei, "Image retrieval using scene graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3668–3678.
[16] S. Thrun, J.-S. Gutmann, D. Fox, W. Burgard, and B. Kuipers, "Integrating topological and metric maps for mobile robot navigation: A statistical approach," in *Proc. AAAI/IAAI*, 1998, pp. 989–995.
[17] N. Tomatis, I. Nourbakhsh, and R. Siegwart, "Combining topological and metric: A natural integration for simultaneous localization and map building," in *Proc. 4th Eur. Workshop Adv. Mobile Robots (Eurobot)*, Zürich, Switzerland, 2001.
[18] R. Ramaithitima, M. Whitzer, S. Bhattacharya, and V. Kumar, "Automated creation of topological maps in unknown environments using a swarm of resource-constrained robots," *IEEE Robot. Autom. Lett.*, vol. 1, no. 2, pp. 746–753, Jul. 2016.
[19] B. Kaleci, C. M. Senler, O. Parlaktuna, and U. Gürel, "Constructing topological map from metric map using spectral clustering," in *Proc. IEEE 27th Int. Conf. Tools with Artif. Intell. (ICTAI)*, Nov. 2015, pp. 139–145.
[20] M. Liu, F. Colas, F. Pomerleau, and R. Siegwart, "A Markov semi-supervised clustering approach and its application in topological map extraction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 4743–4748.
[21] B. Kaleci, O. Parlaktuna, and U. Gürel, "A comparative study for topological map construction methods from metric map," in *Proc. 26th Signal Process. Commun. Appl. Conf. (SIU)*, May 2018, pp. 1–4.
[22] A. A. Ravankar, A. Ravankar, T. Emaru, and Y. Kobayashi, "A hybrid topological mapping and navigation method for large area robot mapping," in *Proc. 56th Annu. Conf. Soc. Instrum. Control Eng. Jpn. (SICE)*, Sep. 2017, pp. 1104–1107.
[23] R. Capobianco, G. Gemignani, D. D. Bloisi, D. Nardi, and L. Iocchi, "Automatic extraction of structural representations of environments," in *Intelligent Autonomous Systems 13*. Springer, 2016, pp. 721–733.
[24] S. Ochmann, R. Vock, and R. Klein, "Automatic reconstruction of fully volumetric 3D building models from oriented point clouds," *ISPRS J. Photogramm. Remote Sens.*, vol. 151, pp. 251–262, May 2019.
[25] N. Savinov, A. Dosovitskiy, and V. Koltun, "Semi-parametric topological memory for navigation," 2018, *arXiv:1803.00653*. [Online]. Available: http://arxiv.org/abs/1803.00653
[26] S. Gupta, D. Fouhey, S. Levine, and J. Malik, "Unifying map and landmark based representations for visual navigation," 2017, *arXiv:1712.08125*. [Online]. Available: http://arxiv.org/abs/1712.08125
[27] Y. Wu, Y. Wu, A. Tamar, S. Russell, G. Gkioxari, and Y. Tian, "Bayesian relational memory for semantic visual navigation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 2769–2779.
[28] P. Xu, X. Chang, L. Guo, P.-Y. Huang, X. Chen, and A. G. Hauptmann, "A survey of scene graph: Generation and application," EasyChair, Tech. Rep., Tech. Rep., 2020.
[29] H. Qi, Y. Xu, T. Yuan, T. Wu, and S.-C. Zhu, "Scene-centric joint parsing of cross-view videos," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018.
[30] R. Wang, Z. Wei, P. Li, Q. Zhang, and X. Huang, "Storytelling from an image stream using scene graphs," in *Proc. AAAI*, 2020, pp. 9185–9192.
[31] J. Gu, S. Joty, J. Cai, H. Zhao, X. Yang, and G. Wang, "Unpaired image captioning via scene graph alignments," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 10323–10332.
[32] S. Schuster, R. Krishna, A. Chang, L. Fei-Fei, and C. D. Manning, "Generating semantically precise scene graphs from textual descriptions for improved image retrieval," in *Proc. 4th Workshop Vis. Lang.*, 2015, pp. 70–80.
[33] A. Zareian, S. Karaman, and S.-F. Chang, "Bridging knowledge graphs to generate scene graphs," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020.
[34] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, "Neural motifs: Scene graph parsing with global context," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5831–5840.
[35] Y. Li, W. Ouyang, B. Zhou, J. Shi, C. Zhang, and X. Wang, "Factorizable net: An efficient subgraph-based framework for scene graph generation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 335–351.
[36] R. Drouilly, P. Rives, and B. Morisset, "Fast hybrid relocation in large scale metric-topologic-semantic map," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 1839–1845.
[37] B. Park, J. Choi, and W. K. Chung, "Incremental hierarchical roadmap construction for efficient path planning," *ETRI J.*, vol. 40, no. 4, pp. 458–470, 2018.
[38] M. Hanheide, M. Göbelbecker, G. S. Horn, A. Pronobis, K. Sjöö, A. Aydemir, P. Jensfelt, C. Gretton, R. Dearden, M. Janicek, H. Zender, G.-J. Kruijff, N. Hawes, and J. L. Wyatt, "Robot task planning and explanation in open and uncertain worlds," *Artif. Intell.*, vol. 247, pp. 119–150, Jun. 2017.
[39] J.-R. Ruiz-Sarmiento, C. Galindo, and J. Gonzalez-Jimenez, "Building multiversal semantic maps for mobile robot operation," *Knowl.-Based Syst.*, vol. 119, pp. 257–272, 2017.
[40] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, T. Fico, and L. Jurišica, "Path planning with modified a star algorithm for a mobile robot," *Procedia Eng.*, vol. 96, pp. 59–69, Jan. 2014.
[41] *Gmapping*. Accessed: Aug. 1, 2020. [Online]. Available: http://wiki.ros.org/gmapping/
[42] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 3, 2006, pp. 850–855.
[43] *Turtlebot*. Accessed: Aug. 1, 2020. [Online]. Available: https://www.turtlebot.com
[44] *Ros*. Accessed: Aug. 1, 2020. [Online]. Available: https://www.ros.org/
[45] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. S. Bernstein, and F.-F. Li, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," 2016, *arXiv:1602.07332*. [Online]. Available: http://arxiv.org/abs/1602.07332
[46] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017.
[47] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang, "Unbiased scene graph generation from biased training," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 3716–3725.

ZHIYONG LIAO received the B.Eng. degree from the College of Mechanical and Vehicle Engineering, Hunan University (HNU), Changsha, China, in 2018. He is currently pursuing the M.S. degree in control science and engineering with the National University of Defense Technology (NUDT), Changsha.

His current research interests include intelligence decision-making and control, knowledge graph, and human–robotic interaction.

YU ZHANG received the B.Eng., M.S., and Ph.D. degrees in automatic control from the National University of Defense Technology (NUDT), Changsha, China, in 2004, 2007, and 2012, respectively.

In 2012, he joined the College of Mechatronics and Automation, NUDT. He is currently an Associate Professor with the College of Intelligence Science and Technology, NUDT. He has directed five research projects. He has coauthored two books and has published over 30 papers in refereed international journals and academic conferences proceedings. His research interests include intelligence decisions, mission planning, automation and control engineering, and complex systems.

JUNREN LUO received the B.Eng. degree in command automation engineering from Information Engineering University, Zhengzhou, China, in 2012, and the M.S. degree in command automation engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2019, where he is currently pursuing the Ph.D. degree in control science and engineering.

His current research interests include goal recognition-based location and routing planning, multi-agent learning for cross-domain heterogeneous swarms, and graph representation learning-based combinatorial optimization for network analysis.

WEILIN YUAN received the B.Eng. and M.S. degrees in automatic control from the National University of Defense Technology (NUDT), Changsha, China, in 2016 and 2019, respectively, where he is currently pursuing the Ph.D. degree in control science and engineering.

His current research interests include intelligence decision-making and control, adversarial reasoning, and behavior game theory.