

Robot Navigation Based on Situational Awareness

Xilong Liu¹, Zhiqiang Cao¹, Senior Member, IEEE, Yingying Yu¹,
Guangli Ren¹, Junzhi Yu², Fellow, IEEE, and Min Tan¹

Abstract—Autonomous navigation for mobile robots is a prerequisite to complete the tasks. In this article, we propose a novel navigation method based on situational awareness. The scene predictor, scene interpreter and the topological map-based planner are the main components, where the scene interpreter is used to realize the mapping of perception information, and the scene predictor predicts the information in the neighborhood of current position. The basic idea of this research is the feeling of path. Based on the definitions of path and the amount of perception, the situational awareness fitting network is designed as the scene interpreter, and the position of the robot in a path can be implicitly described by a situational awareness value. With the global guidance of the planner, the robot achieves navigation. The proposed navigation method does not rely on Cartesian-based global position and its effectiveness is verified by simulations and experiments.

Index Terms—Path perception, robot navigation, situational awareness fitting network.

I. INTRODUCTION

NAVIGATION is one of the fundamental function of a mobile robot. Autonomous and reliable navigation is a prerequisite for mobile robots to complete the tasks with satisfactory performances [1]. Outdoor navigation can often be guided by satellites-based localization systems, including GPS and Beidou. Such navigation is often ineffective when satellite signals are weak or even there is no signal, especially for service mobile robots in indoor environments. In this case, the robot has to rely on its sensing to realize the navigation. Pioneering researches have been conducted on sensing-based navigation. Generally speaking, robot navigation [2]–[5] involves three aspects: mapping, localization and path planning. Only with effective localization information can the robot make a proper decision.

Manuscript received 29 October 2020; revised 6 March 2021; accepted 15 April 2021. Date of publication 26 April 2021; date of current version 9 September 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62073322, Grant 61633020, Grant 61973302, and Grant 61633017. (Xilong Liu and Yingying Yu contributed equally to this work.) (Corresponding author: Zhiqiang Cao.)

Xilong Liu, Zhiqiang Cao, Yingying Yu, Guangli Ren, and Min Tan are with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: zhiqiang.cao@ia.ac.cn).

Junzhi Yu is with the State Key Laboratory for Turbulence and Complex System, Department of Mechanics and Engineering Science, BIC-ESAT, College of Engineering, Peking University, Beijing 100871, China.

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TCDS.2021.3075862>.

Digital Object Identifier 10.1109/TCDS.2021.3075862

Mapping is actually a process of characterizing the environment with data. Many types of maps have been presented, including grid map [6], feature map [7], and topological map, where the topological map abstracts the environment into the connection between edges and points. Grid map is easy to be created and its spatial resolution depends on the size of the grid. Endres *et al.* [9] used an RGB-D camera to generate a 3-D point cloud map, which is converted to a 3-D occupancy grid map to reduce storage space by the octree-based OctoMap method [8]. Mur-Artal *et al.* [10] presented the ORB-SLAM method, which adopts ORB features with rotational invariance to generate the map. Zhang *et al.* [11] took the straight lines in the environment as features to construct a map, which achieves a better reflection of the structural information of the environment. Zuo *et al.* [12] built an environment map using feature points and lines.

With an environment map, the robot can estimate its pose by matching the environment information provided by the robot sensors. Bosse and Zlot [13] proposed a solution to 3-D scan matching where a continuous 6DOF sensor trajectory is recovered to correct the point cloud alignments, which produces locally accurate maps and thus allows for a reliable estimation of the vehicle motion. LOAM [14] matches feature points extracted on sharp edges and planar surface patches to estimate motion. In some VSLAM, such as ORB-SLAM [10] and VINS [15], the Bag of Words can be used to match the current frame with the keyframes for the robot relocalization, where the features, including ORB, Surf, and FAST are utilized. Whelan *et al.* [16] presented an approach to the dense visual SLAM, which performs time windowed surfel-based dense data fusion in combination with frame-to-model tracking and nonrigid deformation, where the randomized ferns encoding is used [17]. For large-scale navigation, many indistinguishable scenes possibly affect the system performance.

In recent years, deep neural network has become a research hotspot in the field of artificial intelligence, which possesses excellent performance in data expression and nonlinear fitting. A successful application of deep learning is to solve the problem of robot localization where the pose regression networks are used. Kendall *et al.* [18] presented the deep convolutional neural networks for end-to-end 6-DOF camera pose localization. This network is trained to regress the camera pose from a single RGB image. Li *et al.* [19] proposed an indoor relocalization system using a dual-stream convolutional neural network with both color images and depth images as the network inputs. This method performs well in challenging environments, such as motion blur. Walch *et al.* [20]

proposed a CNN+LSTM architecture for camera pose regression in indoor and outdoor scenes, where they reshaped the 2048-D vector of the full connection layer in PoseNet to a 32×64 matrix followed by four LSTM structures. A possible problem of the methods based on deep learning is that they need global pose labels.

On the other hand, deep reinforcement learning has been adopted for robot navigation. Savinov *et al.* [21] proposed a method with the semiparametric topological memory (SPTM), which is evaluated on the classic game Doom [22] with the maze environment. The agent first builds the topological graph by exploring the environment and trains the retrieval network and the locomotion network. Then the agent determines the goal node and the current node by the retrieval network, while the locomotion network outputs the action that allows the agent to move between nodes in the graph. Gupta *et al.* [23] proposed the cognitive mapper and planner (CMP) architecture to learn to map and plan a sequence of actions toward goals. And CMP can be trained end to end. Zhu *et al.* [24] presented an actor-critic model to the task of target-driven visual navigation where the AI2-THOR framework is designed to provide a simulation environment with 3-D scenes and physics engine.

There is also another very promising and important direction, which is inspired by the navigation of human or animals, where the biological mechanisms or behavioral imitation are concerned. Chen and Mo developed an effective brain-like biologically plausible mathematical model for robotic navigation [25] with cerebral cortex layer, hippocampal region layer and basal ganglia layer. Based on computational models of the rodent hippocampus, Milford and Wyeth developed the RatSLAM system, which constructs the experience map [26]. With the constructed global experience map, the robot can execute navigation assisted by local obstacle map [27]. In [28], the cognitive map is built using an RGB-D-based RatSLAM algorithm; on this basis, a grid-based direction planner and a multilayered local navigation module are designed to achieve navigation. Tang *et al.* proposed a neuroinspired cognitive navigation model combining the cognitive map with episodic memory [3], where the former is implemented by a 3-D continuous attractor network structure. For a human, navigation toward a destination can be easily completed, which may be regarded as the selection and tracking of paths. After determining the appropriate paths according to the prior knowledge of the environment, a human marches along the scheduled paths without the support of Cartesian-based global position. Inspired by this line of thought, a novel navigation method is proposed in this article. With a definition of path, the scene interpreter implemented by the situational awareness fitting network is introduced for path perception. And the overall environment is represented as a series of paths and each path is modeled as a situational awareness fitting network. As a result, the position of the robot in a path may be mapped to a situational awareness value, and the proposed method does not rely on the Cartesian-based global position.

The organization of this article is as follows. Section II presents the framework of robot navigation. Section III gives the definition of path and path perception theory. In

Section IV, the navigation based on situational awareness is presented. The simulation and experimental results are demonstrated in Sections V and VI, respectively. Section VII concludes this article.

II. FRAMEWORK OF ROBOT NAVIGATION

Arriving at the destination in a familiar environment is simple for human. The process of human navigation relies on path. Path is a concept abstracted from the perception of environment, which makes the navigation easier because it simplifies the description of environments. From the perspective of imitation, it is required to answer the questions “what is path” and “how to describe the path quantitatively.” Human is good at abstracting objects’ information into topological description. It is rational to take the concept “path” as an element of the topological graph abstracted from the environment. When paths can be perceived quantitatively, navigation may be achieved even with imprecise motion control. Fig. 1 shows the framework of the proposed situational awareness-based robot navigation, where each position of the robot is represented by a situational awareness value. The situational awareness values of both current scene and destination with the topological map are regarded as inputs, and the decision is made by a planner and a motion optimizer. The former provides a global planning based on the topological map and the latter determines the motion of the robot using scene predictor and scene interpreter. For the computable implementation of the scene interpreter and predictor, the concept of path is defined and the path perception theory is presented in the next section.

III. PATH AND PATH PERCEPTION THEORY

Just like the road network in human society, in this article, the robot environment is divided into many paths constrained by mechanical structure and the obstacles, where the path describes feasible zones that the robot can traverse. For a path, there exist some traversable routes. A route in mathematics is often represented as a curve connecting the starting point and ending point, which may be expressed by multistage analytic form or point array interpolation.

A. Definition of Path

Before the path is defined, feasible route and connected region are first introduced.

1) *Feasible Route*: The feasible route Λ is defined as a sequence of location points where the robot moves safely. The first point and the end point of a route Λ are labeled as ${}^{\Lambda}P_0$ and ${}^{\Lambda}P_1$, respectively, and its length is $\|\Lambda\|$.

2) *Connected Region*: For a set Γ of location points, if $\forall A, B \in \Gamma, \exists \Lambda \subseteq \Gamma$ and $A, B \in \Lambda$, the robot can move along the route Λ from point A to point B , the set Γ is denoted as a connected region.

We label Γ_0 and Γ_1 as two connected regions and $\Gamma_0 \cap \Gamma_1 = \emptyset$. If a feasible routes set \mathbb{C} satisfies $\forall \Lambda \in \mathbb{C}$ with ${}^{\Lambda}P_0 \in \Gamma_0$ and ${}^{\Lambda}P_1 \in \Gamma_1$, and the location points set of all feasible routes in \mathbb{C} is a connected region, it means that there

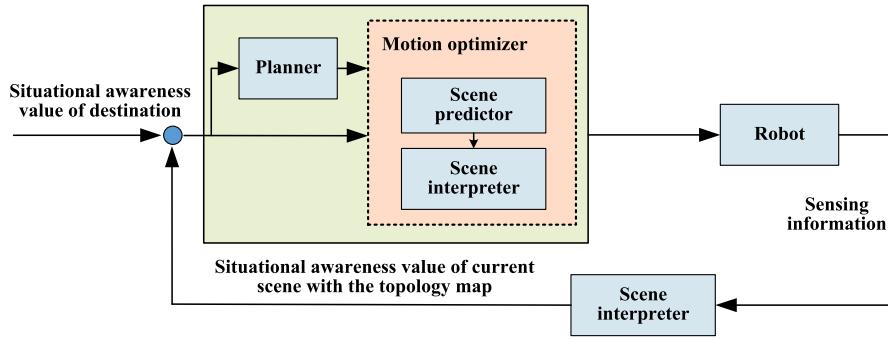


Fig. 1. Framework of situational awareness-based robot navigation.

exists a path. The location points set \mathbb{Q} of all feasible routes in \mathbb{C} is defined as a path connecting Γ_0 and Γ_1 .

According to the path definition, for $\forall X \in \mathbb{Q}$, there exists a set of feasible routes $\tilde{\mathbb{C}}$ that $\forall \Lambda \in \tilde{\mathbb{C}}, {}^{\Lambda}P_0 = X, {}^{\Lambda}P_1 \in \Gamma_1$. The shortest length in $\tilde{\mathbb{C}}$ is used as the evaluation of the position X , and it is defined as situational awareness value ${}^{\mathbb{Q}}v_X$ of X in \mathbb{Q} . ${}^{\mathbb{Q}}v_X = \min_{\Lambda \in \tilde{\mathbb{C}}} \|\Lambda\|$. Clearly, for a given path \mathbb{Q} , ${}^{\mathbb{Q}}v_X$ is actually only related to the robot position. In some sense, the situational awareness value may be regarded as an implicit expression of position of the robot.

Generally speaking, the robot perception for the environment can be encoded as a perception vector $\mathbf{s} : (s_1, s_2, \dots, s_D)$, where D is the dimension of \mathbf{s} . For a path \mathbb{Q} , the perception vector of a point is expressed by ${}^{\mathbb{Q}}\mathbf{s}$, and all perception vectors constitute a set ${}^{\mathbb{Q}}\mathbf{s}$. The mapping from perception vectors to corresponding situational awareness values can be expressed as $g : {}^{\mathbb{Q}}\mathbf{s} \rightarrow {}^{\mathbb{Q}}v$, where ${}^{\mathbb{Q}}v$ is the set of ${}^{\mathbb{Q}}v$ in \mathbb{Q} . To ensure that the mapping $g : {}^{\mathbb{Q}}\mathbf{s} \rightarrow {}^{\mathbb{Q}}v$ is operational, the situational awareness values of sample points in the path can be labeled, and a computational model is needed to map the perception vector to an exact value. For the former, one can determine the approximated situational awareness value of a point using odometer, while for the latter, the powerful data abstraction and fitting capability of deep neural networks paves the foundation of our computational model.

B. Path Perception Theory

For a small-scale environment, it is enough to build the computational model by a single deep neural network with enough layers and parameters. However, it does not always work especially for large-scale environments. With the increasing of complexity of large-scale environments, this single deep neural network becomes more complex with a heavier computation burden. A preferable solution is to divide the environment into many paths and each path is modeled independently. It is worth noting that designing a common deep network with fixed number of layers and parameters is not suitable for each path due to different complexity of paths. The more complex a path is, the larger the capacity of corresponding deep network is. Also, it is difficult to find a proper deep network by directly relying on original perception vector. How to determine the capacity of the deep network remains unsolved.

In order to solve this problem, the amount of perception for a path is introduced, and it reflects the complexity of

the path. A typical feature of the amount of perception of a path is that its value is relatively fixed although the local information perceived by the robot in the path is time-varying. Mathematically, the amount of perception from one measurement may be characterized as the difference between the perception vector and the corresponding predicted one based on historical information. The bigger the difference is, the larger the amount of perception is. The total sum of all differences in a path constitutes of the amount of perception of a path. The larger amount of perception of a path is corresponding to a more complex environment, and thus a larger network capacity is required. The positive correlation between the amount of perception of a path and the network capacity provides a basis to determine the deep network.

As mentioned above, to calculate the amount of perception from one measurement, the predicted vector should be represented quantitatively based on historical information. For the vision sensor, laser, or ultrasonic sensor used in navigation, the perception process can be formalized as $\Theta \rightarrow P$, which is a mapping from the observation angles Θ to the perception results P . First, we consider a continuous perception with $p = f_1(\theta)$, where $p \in P$, $\theta \in \Theta$, and θ is continuous. Under the world coordinate system, θ can be mapped to a spatial position $W = f_2(\theta)$. The set of all spatial positions corresponding to the domain of θ is labeled as the perceptible domain Ω . With $W \in \Omega$, the perception process may be expressed as a function $p = {}^{\{St\}}g(W, t)$, where $\{St\}$ refers to the sensor coordinate system at time t . All perceived results at time t are expressed as ${}^{\{St\}}g(\Omega_t, t)$, where Ω_t represents the perceptible domain at time t . If there is a movement for the robot between time t_1 and t_2 , the prediction at time t_2 based on the perceptual result at time t_1 involves two types with intersection or nonintersection points, according to whether a spatial point in Ω_{t_2} also belongs to the set Ω_{t_1} .

The prediction with intersection points considers those points that belong to both Ω_{t_1} and Ω_{t_2} . It is expressed as follows:

$$\left. {}^{\{St_1\}}g(\Omega_{t_1}, t_1) \right\} \Rightarrow {}^{\{St_2\}}g(W, t_2) \\ W \in \Omega_{t_1} \cap W \in \Omega_{t_2} \quad (1)$$

where ${}^{\{St_2\}}g(\Omega_{t_1}, t_1)$ is the transformation matrix of coordinate system from $\{St_1\}$ to $\{St_2\}$, and the symbol “ \Rightarrow ” represents the prediction.

The prediction with nonintersection points only concerns the points in Ω_{t_2} without in Ω_{t_1} , and they constitute the set $\Psi_{\tilde{t}_1 t_2} = \{W | W \in \Omega_{t_2} \cap W \notin \Omega_{t_1}\}$ with the number of $\|\Psi_{\tilde{t}_1 t_2}\|$.

It is noted that the information provided by sensors is actually discrete samplings of the environment. We label $\mathbf{s}_i : (s_1^i, s_2^i, \dots, s_D^i)$ ($i = 1, 2, \dots, N$) as the perception vector of i th frame in a route where N is the number of the frames. The time t_i is corresponding to \mathbf{s}_i . The amount of perception from \mathbf{s}_i to \mathbf{s}_{i+1} is given by

$$F(\mathbf{s}_i, \mathbf{s}_{i+1}) = f(\mathbf{s}_i, \mathbf{s}_{i+1}) + \mu \|\Psi_{\tilde{t}_i t_{i+1}}\| \quad (2)$$

where $f(\mathbf{s}_i, \mathbf{s}_{i+1})$ is the amount of perception for the points that belong to both Ω_{t_1} and Ω_{t_2} , $\mu \|\Psi_{\tilde{t}_i t_{i+1}}\|$ corresponds to the points in Ω_{t_2} without in Ω_{t_1} . $\mu (> 0)$ is a given constant.

All perception vectors in a route constitute the set $S = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_N\}$. We define the metric space as $\aleph : (S, F)$, where F is a metric defined on S according to (2). For given perception vectors $\mathbf{s}_a, \mathbf{s}_b, \mathbf{s}_c \in S$, where $a, b, c \in [1, N]$, the amount of perception should possess the following properties.

- 1) *Nonnegativity*: $F(\mathbf{s}_a, \mathbf{s}_b) \geq 0$.
- 2) $F(\mathbf{s}_a, \mathbf{s}_a) = 0$.
- 3) *Symmetry*: $F(\mathbf{s}_a, \mathbf{s}_b) = F(\mathbf{s}_b, \mathbf{s}_a)$.
- 4) *Triangle Inequality*: $F(\mathbf{s}_a, \mathbf{s}_c) \leq F(\mathbf{s}_a, \mathbf{s}_b) + F(\mathbf{s}_b, \mathbf{s}_c)$.

With (2), an analytical model

$$\begin{aligned} F(\mathbf{s}_a, \mathbf{s}_b) &= f(\mathbf{s}_a, \mathbf{s}_b) + \mu \|\Psi_{\tilde{t}_a t_b}\| \\ &= \hat{g}[\tilde{E}(\mathbf{s}_a), \tilde{E}(\mathbf{s}_b)] + \mu \|\Psi_{\tilde{t}_a t_b}\| \\ &= \int_{\tau \in \Theta_{t_a t_b}} h(\|e_a(\tau), e_b(\tau)\|) d\tau + \mu \|\Psi_{\tilde{t}_a t_b}\| \end{aligned} \quad (3)$$

is constructed, where $\tilde{E}(\mathbf{s}_a)$ and $\tilde{E}(\mathbf{s}_b)$ are the estimates of the overlapping domain $\Theta_{t_a t_b}$ by using \mathbf{s}_a and \mathbf{s}_b , respectively, and $\Theta_{t_a t_b} = \{\theta | \theta \in f_2^{-1}(\Omega_{t_a}) \cap \theta \in f_2^{-1}(\Omega_{t_b})\}$. Besides, we label $\Theta_{\tilde{t}_a t_b}$ as $\{\theta | \theta \notin f_2^{-1}(\Omega_{t_a}) \cap \theta \in f_2^{-1}(\Omega_{t_b})\}$. $e_a(\tau)$ and $e_b(\tau)$ are the values corresponding to the observation angle τ in $\tilde{E}(\mathbf{s}_a)$ and $\tilde{E}(\mathbf{s}_b)$, respectively. $h(\|e_a(\tau), e_b(\tau)\|) = (-e^{-\frac{20}{\sigma/\xi}(\|e_a(\tau), e_b(\tau)\|)} + 1) \log_2(\sigma/\xi)$ refers to the amount of perception produced by the difference of $e_a(\tau)$ and $e_b(\tau)$, $\|e_a(\tau), e_b(\tau)\|$ is the Euclidean distance between $e_a(\tau)$ and $e_b(\tau)$. σ is the range of the sensor and ξ is a given resolution. μ is set to $\log_2(\sigma/\xi)$. Next, the aforementioned analytical model is proved as a metric of the metric space \aleph .

Proof: With the function $h(\|e_a(\tau), e_b(\tau)\|) \geq 0$, we have $\int_{\tau \in \Theta_{t_a t_b}} h(\|e_a(\tau), e_b(\tau)\|) d\tau \geq 0$. Obviously, $F(\mathbf{s}_a, \mathbf{s}_b) \geq 0$ and the nonnegativity is proved.

$F(\mathbf{s}_a, \mathbf{s}_a) = \int_{\tau \in \Theta_{t_a t_a}} h(\|e_a(\tau), e_a(\tau)\|) d\tau + \mu \|\Psi_{\tilde{t}_a t_a}\| = 0$, and thus the designed analytical model meets the second property.

Due to the fact that $h(\|e_a(\tau), e_b(\tau)\|) = h(\|e_b(\tau), e_a(\tau)\|)$ and $\|\Psi_{\tilde{t}_a t_b}\| = \|\Psi_{\tilde{t}_b t_a}\|$, one can see that

$$\begin{aligned} F(\mathbf{s}_a, \mathbf{s}_b) &= \int_{\tau \in \Theta_{t_a t_b}} h(\|e_a(\tau), e_b(\tau)\|) d\tau + \mu \|\Psi_{\tilde{t}_a t_b}\| \\ &= \int_{\tau \in \Theta_{t_b t_a}} h(\|e_b(\tau), e_a(\tau)\|) d\tau + \mu \|\Psi_{\tilde{t}_b t_a}\| \\ &= F(\mathbf{s}_b, \mathbf{s}_a). \end{aligned} \quad (4)$$

Therefore, the symmetry property is satisfied for the proposed analytical model.

To prove that the analytical model satisfies the fourth property, first, we conduct the following expansion:

$$\begin{aligned} F(\mathbf{s}_a, \mathbf{s}_c) &= \int_{\tau \in \Theta_{t_a t_b t_c}} h(\|e_a(\tau), e_c(\tau)\|) d\tau \\ &+ \int_{\tau \in \Theta_{t_a \tilde{t}_b t_c}} h(\|e_a(\tau), e_c(\tau)\|) d\tau \\ &+ \mu \|\Psi_{\tilde{t}_a t_b t_c}\| + \mu \|\Psi_{t_a \tilde{t}_b t_c}\|. \end{aligned} \quad (5)$$

Similarly, we obtain $F(\mathbf{s}_a, \mathbf{s}_b)$ and $F(\mathbf{s}_b, \mathbf{s}_c)$, which is summed as

$$\begin{aligned} F(\mathbf{s}_a, \mathbf{s}_b) + F(\mathbf{s}_b, \mathbf{s}_c) &= \int_{\tau \in \Theta_{t_a t_b t_c}} h(\|e_a(\tau), e_b(\tau)\|) d\tau \\ &+ \int_{\tau \in \Theta_{t_a t_b \tilde{t}_c}} h(\|e_a(\tau), e_b(\tau)\|) d\tau \\ &+ \mu \|\Psi_{\tilde{t}_a t_b t_c}\| + \mu \|\Psi_{t_a t_b \tilde{t}_c}\| \\ &+ \int_{\tau \in \Theta_{t_a t_b t_c}} h(\|e_b(\tau), e_c(\tau)\|) d\tau \\ &+ \int_{\tau \in \Theta_{\tilde{t}_a t_b t_c}} h(\|e_b(\tau), e_c(\tau)\|) d\tau \\ &+ \mu \|\Psi_{\tilde{t}_a t_b t_c}\| + \mu \|\Psi_{t_a \tilde{t}_b t_c}\|. \end{aligned} \quad (6)$$

It is known that the Euclidean distances satisfy

$$\|e_a(\tau), e_c(\tau)\| \leq \|e_a(\tau), e_b(\tau)\| + \|e_b(\tau), e_c(\tau)\|. \quad (7)$$

Considering that $-e^{-(20/(\sigma/\xi))(\|\cdot\|)}$ is a monotonic increasing function of $\|\cdot\|$, therefore, we have

$$-e^{-\frac{20}{\sigma/\xi}(\|e_a(\tau), e_c(\tau)\|)} \leq -e^{-\frac{20}{\sigma/\xi}(\|e_a(\tau), e_b(\tau)\| + \|e_b(\tau), e_c(\tau)\|)}. \quad (8)$$

With $\|\cdot\| \geq 0$, one can deduce that $e^{-(20/(\sigma/\xi))\|\cdot\|} \in (0, 1]$ and $(1 - e^{-(20/(\sigma/\xi))\|e_a(\tau), e_b(\tau)\|}) \geq 0$, $0 < e^{-(20/(\sigma/\xi))\|e_b(\tau), e_c(\tau)\|} \leq 1$. Thus

$$\begin{aligned} &e^{-\frac{20}{\sigma/\xi}(\|e_b(\tau), e_c(\tau)\|)} (1 - e^{-\frac{20}{\sigma/\xi}(\|e_a(\tau), e_b(\tau)\|)}) \\ &\leq (1 - e^{-\frac{20}{\sigma/\xi}(\|e_a(\tau), e_b(\tau)\|)}). \end{aligned} \quad (9)$$

According to (9), one can obtain

$$\begin{aligned} &-e^{-\frac{20}{\sigma/\xi}(\|e_a(\tau), e_b(\tau)\|)} e^{-\frac{20}{\sigma/\xi}(\|e_b(\tau), e_c(\tau)\|)} \\ &\leq -e^{-\frac{20}{\sigma/\xi}(\|e_a(\tau), e_b(\tau)\|)} - e^{-\frac{20}{\sigma/\xi}(\|e_b(\tau), e_c(\tau)\|)} + 1. \end{aligned} \quad (10)$$

Combining $-e^{-(20/(\sigma/\xi))(\|e_a(\tau), e_c(\tau)\|)}$ with (10), we have

$$\begin{aligned} &-e^{-\frac{20}{\sigma/\xi}(\|e_a(\tau), e_c(\tau)\|)} \leq -e^{-\frac{20}{\sigma/\xi}(\|e_a(\tau), e_b(\tau)\|)} \\ &- e^{-\frac{20}{\sigma/\xi}(\|e_b(\tau), e_c(\tau)\|)} + 1. \end{aligned} \quad (11)$$

Correspondingly, the following inequality is satisfied:

$$\begin{aligned} h(\|e_a(\tau), e_c(\tau)\|) &\leq h(\|e_a(\tau), e_b(\tau)\|) + h(\|e_b(\tau), e_c(\tau)\|). \end{aligned} \quad (12)$$

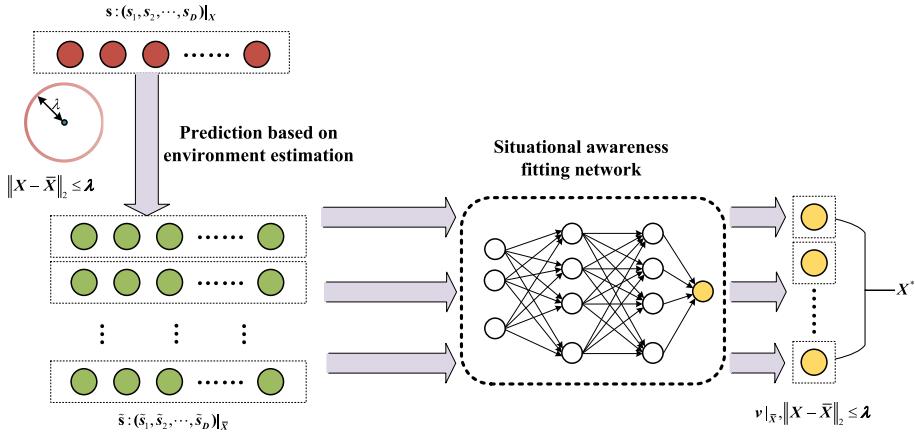


Fig. 2. Navigation of a robot in a path.

For the integral over the common domain $\Theta_{t_a t_b t_c}$ of $F(\mathbf{s}_a, \mathbf{s}_c)$, $F(\mathbf{s}_a, \mathbf{s}_b)$ and $F(\mathbf{s}_b, \mathbf{s}_c)$, it is obtained that

$$\begin{aligned} \int_{\tau \in \Theta_{t_a t_b t_c}} h(\|e_a(\tau), e_c(\tau)\|) d\tau &\leq \int_{\tau \in \Theta_{t_a t_b t_c}} h(\|e_a(\tau), e_b(\tau)\|) d\tau \\ &+ \int_{\tau \in \Theta_{t_a t_b t_c}} h(\|e_b(\tau), e_c(\tau)\|) d\tau. \end{aligned} \quad (13)$$

For the integral over the noncommon domain, $h(\cdot) \leq \mu$ and $\int_{\tau \in \Theta_{t_a t_b t_c}} h(\|e_a(\tau), e_c(\tau)\|) d\tau \leq \mu \|\Psi_{t_a t_b t_c}\|$ holds. Therefore

$$\begin{aligned} \int_{\tau \in \Theta_{t_a t_b t_c}} h(\|e_a(\tau), e_c(\tau)\|) d\tau &\leq \int_{\tau \in \Theta_{t_a t_b t_c}} h(\|e_a(\tau), e_b(\tau)\|) d\tau \\ &+ \int_{\tau \in \Theta_{t_a t_b t_c}} h(\|e_b(\tau), e_c(\tau)\|) d\tau \\ &+ \mu \|\Psi_{t_a t_b t_c}\| + \mu \|\Psi_{t_a t_b t_c}\|. \end{aligned} \quad (14)$$

Combining (5), (6), (13), and (14), one can see that the proposed analytical model is suitable for the fourth property. Thus, the analytical model is a metric of the metric space \mathbb{N} .

With the proposed analytical model, the amount of perception of the robot in one route can be described as

$$I = \sum_{i=1}^{N-1} F(\mathbf{s}_i, \mathbf{s}_{i+1}). \quad (15)$$

The amount I of perception provides the reference to determine the capacity of the deep network, which can be used to map perception vectors to corresponding situational awareness values for subsequent navigation without explicit environment description.

C. Calculation of the Amount of Perception

The robot uses the sensor, such as a laser sensor or RGBD camera to obtain point cloud of each frame. With dense samplings, it is enough for environment estimation using

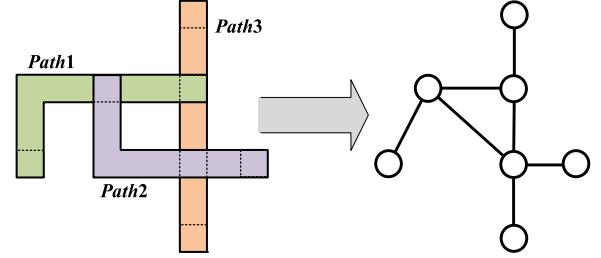


Fig. 3. Illustration of topological map.

the methods, such as the linear interpolation. We label the maximum detection range of the sensor as d_L , and the dimension D of the perception vector is equal to the number of point cloud data per frame. Herein, we consider the 2-D point cloud. The world coordinate system of the environment is denoted as $\{W\} : O_W X_W Y_W Z_W$, and the time corresponding to the i th frame is denoted with t_i . A sensor coordinate system $\{St_i\} : O_{St_i} X_{St_i} Y_{St_i} Z_{St_i}$ is established whose origin point O_{St_i} is located at the position of the robot at t_i , where the X_{St_i} and Y_{St_i} axes are in coincidence with the X_W and Y_W axes, respectively. For the perception vector $\mathbf{s}_i = (s_1^i, s_2^i, \dots, s_D^i)$, the position corresponding to s_d^i is labeled as \mathbf{W}_d^i , where $d = 1, 2, \dots, D$. we establish a polar coordinate system $\{\tilde{S}t_i\}$ whose pole point is O_{St_i} with the X_{St_i} axis as the polar axis. The coordinates of the \mathbf{W}_d^i in $\{\tilde{S}t_i\}$ and $\{St_i\}$ are described as $\{\tilde{S}t_i\} \mathbf{W}_d^i = (\rho_d^i, \theta_d^i)$ and $\{St_i\} \mathbf{W}_d^i = (x_d^i, y_d^i)$, respectively.

The points $\{\tilde{S}t_i\} \mathbf{W}_d^i$ of the i th frame are used to obtain environment estimation \tilde{E}^i using linear interpolation. For two adjacent points \mathbf{W}_d^i and \mathbf{W}_{d+1}^i , when $d = D$, $\mathbf{W}_d^i = \mathbf{W}_D^i$ whereas $\mathbf{W}_{d+1}^i = \mathbf{W}_1^i$. For a point $P^i \in \tilde{E}^i$, the coordinates of P^i in $\{\tilde{S}t_i\}$ and $\{St_i\}$ are given by $\{\tilde{S}t_i\} P^i = (\rho^i, \theta^i)$ and $\{St_i\} P^i = (x^i, y^i)$, respectively. When $\theta^i \in [\min(\theta_d^i, \theta_{d+1}^i), \max(\theta_d^i, \theta_{d+1}^i)]$, according to the linear interpolation, we have $y^i = ([x^i - x_{d+1}^i]/[x_d^i - x_{d+1}^i]) y_d^i + ([x^i - x_d^i]/[x_{d+1}^i - x_d^i]) y_{d+1}^i$. With θ^i of the point P^i , ρ^i is calculated as follows:

$$\rho^i = \frac{x_d^i y_{d+1}^i - x_{d+1}^i y_d^i}{[(x_d^i - x_{d+1}^i) \sin \theta^i - (y_d^i - y_{d+1}^i) \cos \theta^i]}. \quad (16)$$

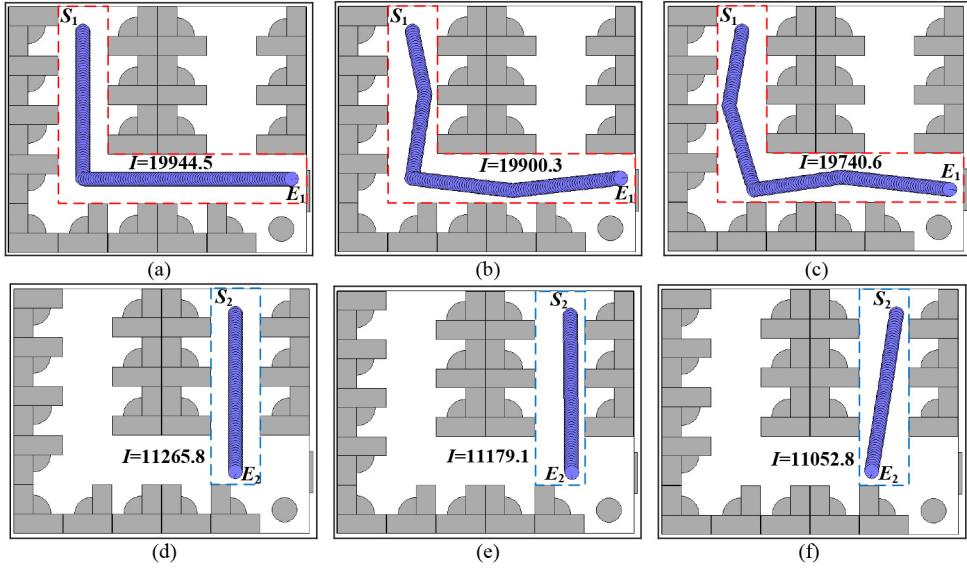


Fig. 4. Amount of perception I of different routes. (a)–(c) Different routes in a path. (d)–(f) Different routes in another path.

Next, the points' coordinates of \tilde{E}^i in a new coordinate system $\{A\}$ are then calculated, where $\{A\}$ is considered as a neighboring one of $\{St_i\}$ and the origin point O_A of $\{A\}$ is in the neighborhood of O_{St_i} . The X_A and Y_A axes of $\{A\}$ have the same directions as X_{St_i} axis and Y_{St_i} axis of $\{St_i\}$, respectively. A polar coordinate system $\{\tilde{A}\}$ corresponding to $\{A\}$ is also established with a polar axis direction of X_A axis. The coordinates of the point $P^i \in \tilde{E}^i$ in $\{A\}$ and $\{\tilde{A}\}$ are described as ${}^{\{A\}}P^i = ({}^{\{A\}}x^i, {}^{\{A\}}y^i)$ and ${}^{\{\tilde{A}\}}P^i = ({}^{\{\tilde{A}\}}\rho^i, {}^{\{\tilde{A}\}}\theta^i)$, respectively. When ${}^{\{\tilde{A}\}}\theta^i \in [\min({}^{\{\tilde{A}\}}\theta_d^i, {}^{\{\tilde{A}\}}\theta_{d+1}^i), \max({}^{\{\tilde{A}\}}\theta_d^i, {}^{\{\tilde{A}\}}\theta_{d+1}^i)]$, according to the coordinate transformation between $\{A\}$ and $\{St_i\}$, i.e., ${}^{\{St_i\}}P^i = {}^{\{A\}}P^i + {}^{\{St_i\}}O_A$, where ${}^{\{St_i\}}O_A = ({}^{\{St_i\}}x_{OA}^i, {}^{\{St_i\}}y_{OA}^i)$ is the position vector of the origin point of $\{A\}$ in $\{St_i\}$, we can obtain

$$\begin{aligned} {}^{\{A\}}y^i &= \frac{{}^{\{A\}}x^i - {}^{\{A\}}x_{d+1}^i}{{}^{\{A\}}x_d^i - {}^{\{A\}}x_{d+1}^i} \left({}^{\{A\}}y_d^i + {}^{\{St_i\}}y_{OA}^i \right) \\ &+ \frac{{}^{\{A\}}x^i - {}^{\{A\}}x_d^i}{{}^{\{A\}}x_{d+1}^i - {}^{\{A\}}x_d^i} \left({}^{\{A\}}y_{d+1}^i + {}^{\{St_i\}}y_{OA}^i \right) - {}^{\{St_i\}}y_{OA}^i. \end{aligned} \quad (17)$$

Substituting ${}^{\{A\}}x^i$ and ${}^{\{A\}}y^i$ of (17) with ${}^{\{\tilde{A}\}}\rho^i \cos {}^{\{\tilde{A}\}}\theta^i$ and ${}^{\{\tilde{A}\}}\rho^i \sin {}^{\{\tilde{A}\}}\theta^i$, respectively, and we have

$${}^{\{\tilde{A}\}}\rho^i = \frac{{}^{\{A\}}x_d^i ({}^{\{A\}}y_{d+1}^i + {}^{\{St_i\}}y_{OA}^i) - {}^{\{A\}}x_{d+1}^i ({}^{\{A\}}y_d^i + {}^{\{St_i\}}y_{OA}^i) - {}^{\{St_i\}}y_{OA}^i}{{}^{\{A\}}x_d^i - {}^{\{A\}}x_{d+1}^i} \sin {}^{\{\tilde{A}\}}\theta^i - ({}^{\{A\}}y_d^i - {}^{\{A\}}y_{d+1}^i) \cos {}^{\{\tilde{A}\}}\theta^i. \quad (18)$$

So far, the coordinates $({}^{\{\tilde{A}\}}\rho^i, {}^{\{\tilde{A}\}}\theta^i)$ of the points of \tilde{E}^i in $\{\tilde{A}\}$ have been obtained. Actually, $\{A\}$ is a neighboring coordinate system near $\{St_i\}$. To ensure the accuracy of environment estimation, $\{A\}$ is selected as $\{St_{i+1}\}$. Equation (3) is discretized as follows:

$$F(\mathbf{s}_i, \mathbf{s}_{i+1}) = \sum_{d=1}^D h(\|e_i(\theta_d), e_{i+1}(\theta_d)\|) \gamma + \mu \|\Psi_{\tilde{t}_i t_{i+1}}\| \quad (19)$$

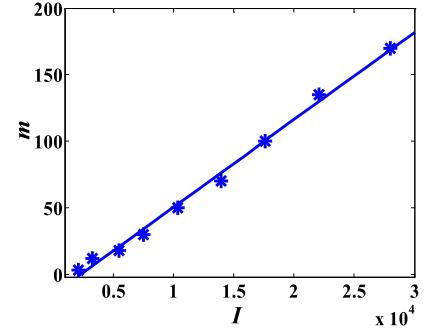


Fig. 5. Relationship between the amount of perception I and the parameter m .

where $\theta_d \in \Theta_{t_i t_{i+1}}$, γ is the angular resolution of the sensor, and

$$\begin{aligned} e_i(\theta_d) &= |{}^{\{St_{i+1}\}}\rho^i|_{\{St_{i+1}\}\theta^i=\theta_d} \\ e_{i+1}(\theta_d) &= \rho^{i+1}|_{\theta^{i+1}=\theta_d}. \end{aligned} \quad (20)$$

Finally, the amount of perception of the robot in a route is given as follows:

$$\begin{aligned} I &= \sum_{i=1}^{N-1} F(\mathbf{s}_i, \mathbf{s}_{i+1}) \\ &= \sum_{i=1}^{N-1} \sum_{d=1}^D \left[\left(-e^{-\frac{20}{\sigma/\xi}} |{}^{\{St_{i+1}\}}\rho^i|_{\{St_{i+1}\}\theta^i=\theta_d} - \rho^{i+1}|_{\theta^{i+1}=\theta_d} \right) + 1 \right] \gamma \\ &\quad \times \log_2(\sigma/\xi) + \mu \|\Psi_{\tilde{t}_i t_{i+1}}\|. \end{aligned} \quad (21)$$

The calculation process of the amount of perception is given in Algorithm 1.

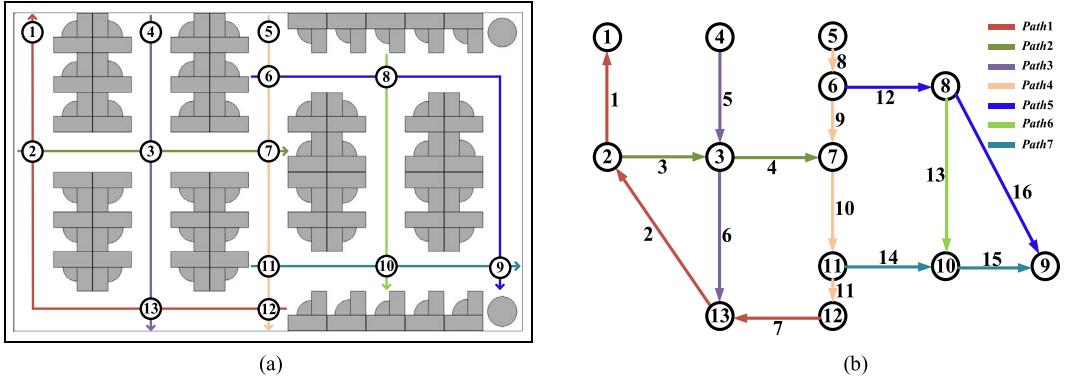


Fig. 6. Environment and the topological map for simulation 2. (a) Environment. (b) Topological map.

Algorithm 1 Calculation of the Amount of Perception

Input: The set of all perception vectors $S = \{s_1, s_2, \dots, s_N\}$.
Output: The amount of perception of the robot.

1. $I \leftarrow 0$;
2. **for** $i=1$ to $N-1$ **do**
3. $F(s_i, s_{i+1}) \leftarrow 0$;
4. **for** $d=1$ to D **do**
5. $\{\tilde{s}_{t_{i+1}}\}_{\theta^i} \leftarrow \theta_d, \theta^{i+1} \leftarrow \theta_d$;
6. **for** $d'=1$ to D **do**
7. **if** $(d \leq D-1)$ **then** $d_1 \leftarrow d, d_2 \leftarrow d+1$;
8. **else** $d_1 \leftarrow D, d_2 \leftarrow 1$;
9. **end if**
10. calculate $(\{\tilde{s}_{t_{i+1}}\}_{\theta^i_{d_1}}, \{\tilde{s}_{t_{i+1}}\}_{\theta^i_{d_2}})$ and $(\{\tilde{s}_{t_{i+1}}\}_{\theta^i_{d_2}}, \{\tilde{s}_{t_{i+1}}\}_{\theta^i_{d_1}})$;
11. **if** $(\{\tilde{s}_{t_{i+1}}\}_{\theta^i_{d_1}}, \{\tilde{s}_{t_{i+1}}\}_{\theta^i_{d_2}})$, $\max(\{\tilde{s}_{t_{i+1}}\}_{\theta^i_{d_1}}, \{\tilde{s}_{t_{i+1}}\}_{\theta^i_{d_2}})$ **then**
12. calculate $\{\tilde{s}_{t_{i+1}}\}_{\rho^i}$ using (18);
13. **break**;
14. **end if**
15. **end for**
16. calculate $e_i(\theta_d), e_{i+1}(\theta_d)$ using (20);
17. $F(s_i, s_{i+1}) \leftarrow F(s_i, s_{i+1}) + h(\|e_i(\theta_d), e_{i+1}(\theta_d)\|)\gamma$;
18. **end for**
19. $F(s_i, s_{i+1}) \leftarrow F(s_i, s_{i+1}) + \mu \|\Psi_{\tilde{t}_{i+1}}\|$;
20. $I \leftarrow I + F(s_i, s_{i+1})$;
21. **end for**
22. **return** I

IV. NAVIGATION BASED ON SITUATIONAL AWARENESS

As mentioned in Section III, the situational awareness values of a path can be regarded as implicit expression of corresponding position of the robot. Thus, the robot can navigate in a path according to the situational awareness values. In reality, the environment is usually large, and it is hard to describe the whole environment using only a path. Topological map provides a simplified environmental expression with less needs of accuracy and storage. In this article, the environment is divided into some paths. And the navigation is realized based on the motion optimizer and topological map-based planner.

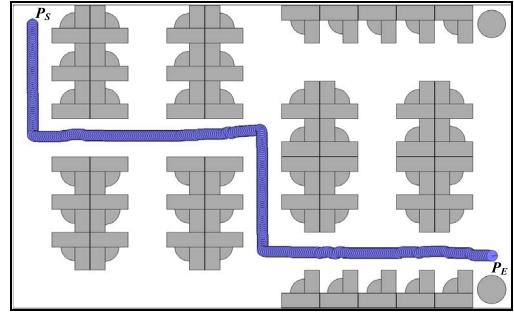


Fig. 7. Motion trajectory of the robot in simulation 2.

A. Navigation in a Path

Inspired by the deep neural network, such as the multilayer perceptron with automatic features extraction, in this article, deep network is adopted to establish a situational awareness fitting network whose capacity is determined according to the amount of perception of the robot in a path. Once the situational awareness fitting network is confirmed, it can output a situational awareness value that is related to the position of the robot in a path. Then, the robot navigates toward the destination only by the situational awareness values of its current position and destination.

In the following, we depict the navigation of the robot in a path based on the motion optimizer, as shown in Fig. 2, where the scene interpreter with the situational awareness fitting network is trained by end-to-end learning. With a perception vector $s : (s_1, s_2, \dots, s_D)|_X$ whose sampling position is X , we can predict the perception vector $\tilde{s} : (\tilde{s}_1, \tilde{s}_2, \dots, \tilde{s}_D)|_{\tilde{X}}$ in the neighborhood of X with the scene predictor using environment estimation, where \tilde{X} is also a sampling position, $\|X - \tilde{X}\|_2 \leq \lambda$, λ is a given threshold. Each predicted perception vector is sent to the fitting network for outputting situational awareness value $v|_{\tilde{X}}$. During the fitting network training, the loss function shown in (22) is adopted

$$\text{loss}(s) = \text{MSE}(v, \hat{v}) = \frac{1}{N_s} \sum_{i=1}^{N_s} \|v - \hat{v}\|_2^2 \quad (22)$$

where N_s is the number of the training samples, \hat{v} and v are the labeled situational awareness value and the network output, respectively.

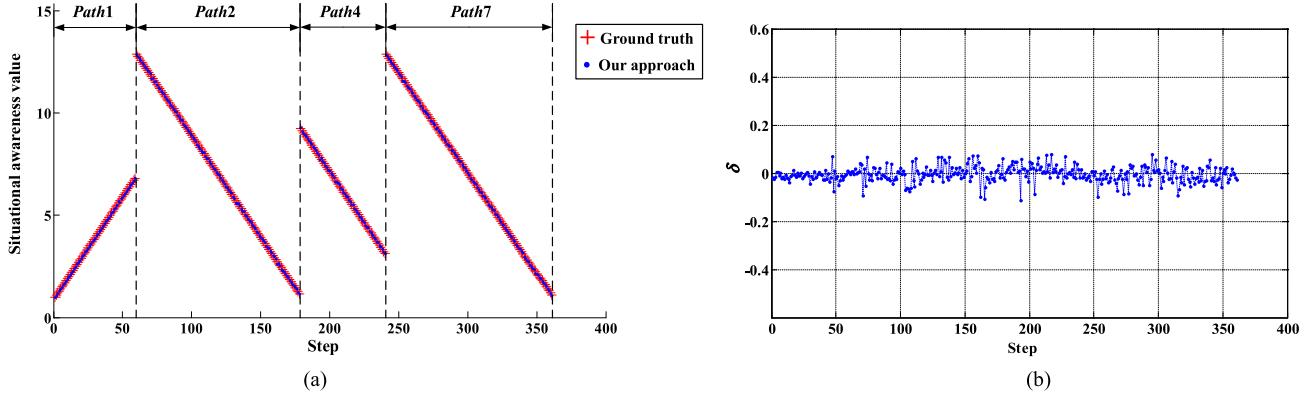


Fig. 8. Results of simulation 2. (a) Variation curves of situational awareness values for our method and ground truth. (b) Curve of the error δ .

Note that the outputs of the fitting network for different perception vectors $\tilde{s} : (\tilde{s}_1, \tilde{s}_2, \dots, \tilde{s}_D)|_{\tilde{X}}$ are different. To make the final decision for the robot, all outputs should be considered. For a path, the robot can move from one end to the other, and *vice versa*. Correspondingly, there exist a forward direction and a reverse direction where the forward direction corresponds to the decreasing one for the situational awareness values in this path. Actually, the forward direction and the reverse direction can be exchanged, and it only depends on personal preference. We denote the motion modes of the robot along the forward direction and the reverse direction with M_f and M_r , respectively. According to different motion modes, the robot makes corresponding decisions.

For the motion mode M_f , the motion situation can be determined as follows:

$$\overrightarrow{XX^*} = \overrightarrow{X \arg\min_{\|X-\tilde{X}\|_2 \leq \lambda} v|_{\tilde{X}}}. \quad (23)$$

When the robot is in M_r , the motion situation is given by

$$\overrightarrow{XX^*} = \overrightarrow{X \arg\max_{\|X-\tilde{X}\|_2 \leq \lambda} v|_{\tilde{X}}}. \quad (24)$$

As mentioned above, \tilde{X} is located in the λ -neighborhood of X , i.e., $\|X-\tilde{X}\|_2 \leq \lambda$. It is obvious that more \tilde{X} is beneficial to determine the motion situation. However, it also brings heavier computation burden. In this article, we choose eight points evenly distributed in the circle of $\|X-\tilde{X}\|_2 = \lambda$.

B. Global Navigation

Fig. 3 illustrates an example of a topological map. The topological map of the global environment can be abstracted as a connected graph $G = (V, E)$, where V and E are the sets of nodes and edges, respectively. Then, we use the planner to determine the optimal nodes sequence.

We label V_s and V_d as the nodes of starting position and destination, respectively, which are involved in nodes planning. Take the difference of situational awareness values of two nodes in the same path as a representation of the weight between these two nodes, the optimal nodes sequence χ^* is

obtained using Dijkstra algorithm

$$\chi^* = \arg \min_{\chi \in \Omega_{sd}} \sum_{q=1}^{N_\chi-1} \left| \mathbb{Q}_v|_{V_{q+1}(\chi)} - \mathbb{Q}_v|_{V_q(\chi)} \right| \quad (25)$$

where χ is a feasible nodes sequence from V_s to V_d whose number of nodes is N_χ , and all χ constitute a set Ω_{sd} ; $\mathbb{Q}_v|_{V_{q+1}(\chi)}$ and $\mathbb{Q}_v|_{V_q(\chi)}$ are the situational awareness values of the nodes $V_{q+1}(\chi)$ and $V_q(\chi)$ in the same path, respectively.

According to the solved χ^* , starting from V_s , one can successively connect the neighboring nodes to form a series of directional edges. On this basis, the global navigation of the robot can be achieved by moving along the directional edges in sequence with a local obstacle avoidance algorithm [29].

When the robot switches from its current edge to next edge, there exists an overlapping zone of these two edges. As the situational awareness fitting networks corresponding to these two edges are different, one has to determine a switching point. We label the situational awareness value of the overlapping zone' center p_c in current edge as $v|_{p_c}$. When the situational awareness value of the robot in current edge is within $[v|_{p_c} - \eta, v|_{p_c} + \eta]$, it means that the robot has entered the overlapping zone and the switch of edges can be executed, where η is a given threshold. The process of the robot global navigation is given in Algorithm 2, where $Path(V_q(\chi^*), V_{q+1}(\chi^*))$ refers to the path formed by the nodes $V_q(\chi^*)$ and $V_{q+1}(\chi^*)$, and $Collision_Avoidance(\cdot)$ is the local obstacle avoidance algorithm. α^* is the direction of the next motion.

V. SIMULATIONS

Simulations are conducted to testify the proposed navigation method. The robot navigates in the environments where a laser sensor is equipped to obtain 2-D point cloud. For each path, a situational awareness fitting network is trained.

A. Relationship of the Amount of Perception With the Capacity of the Situational Awareness Fitting Network

First, we demonstrate the relationship of the amount of perception I with the path. Simulation 1 describes the values of the amount of perception in an office-like environment, as

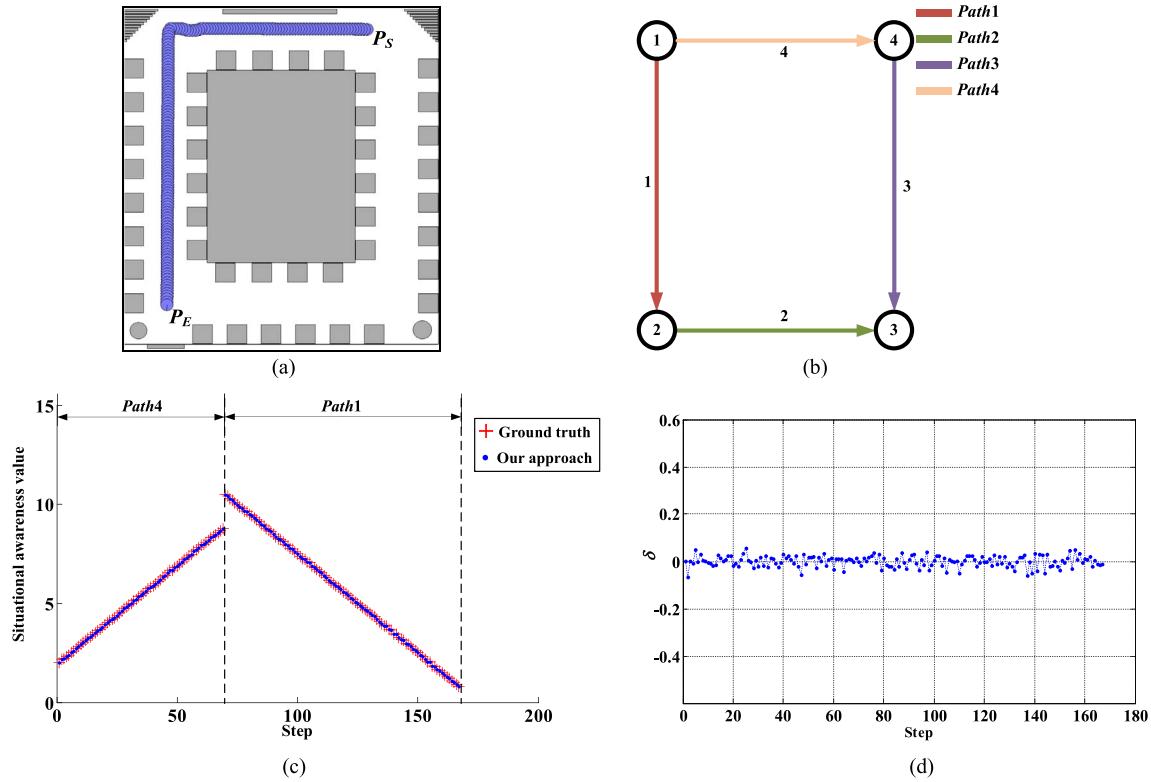


Fig. 9. Results of simulation 3. (a) Motion trajectory of the robot. (b) Topological map. (c) Variation curves of situational awareness values for our method and ground truth. (d) Curve of the error δ .

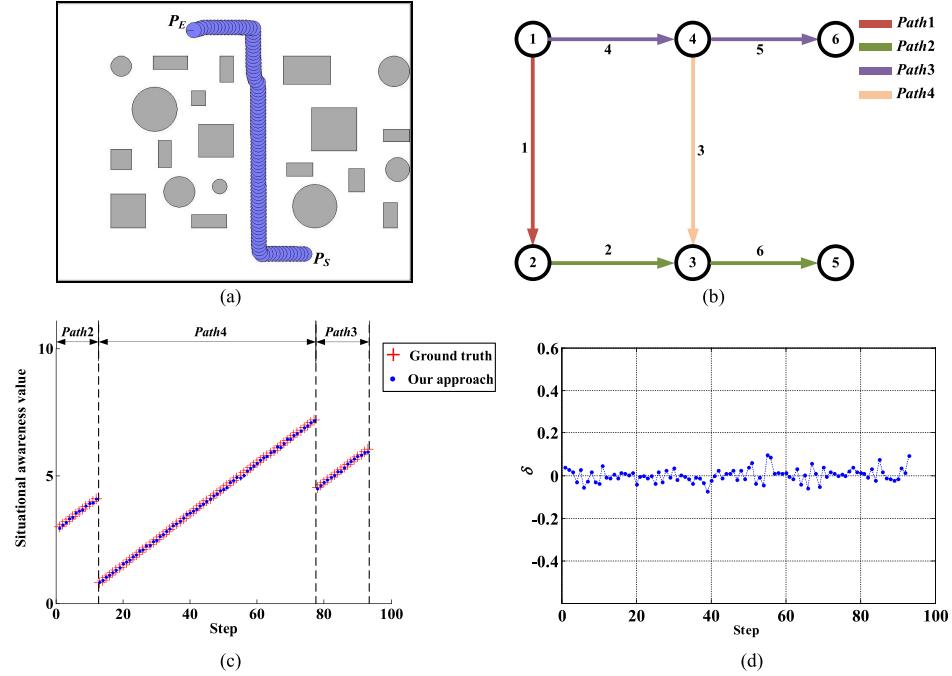


Fig. 10. Results of simulation 4. (a) Motion trajectory of the robot. (b) Topological map. (c) Variation curves of situational awareness values for our method and ground truth. (d) Curve of the error δ .

shown in Fig. 4. Fig. 4(a)–(c) corresponds to the values of I with different routes in a path with the red dashed box, whereas Fig. 4(d)–(f) depicts the values of I with different routes in another path with the blue dashed box. One can see that different routes in the same path have little difference in

the amount of perception, which can be considered as a path attribute. Moreover, there exists an obvious difference in the amount of perception for different paths with varied complexity, and the amount of perception may be regarded as a metric of the path complexity.

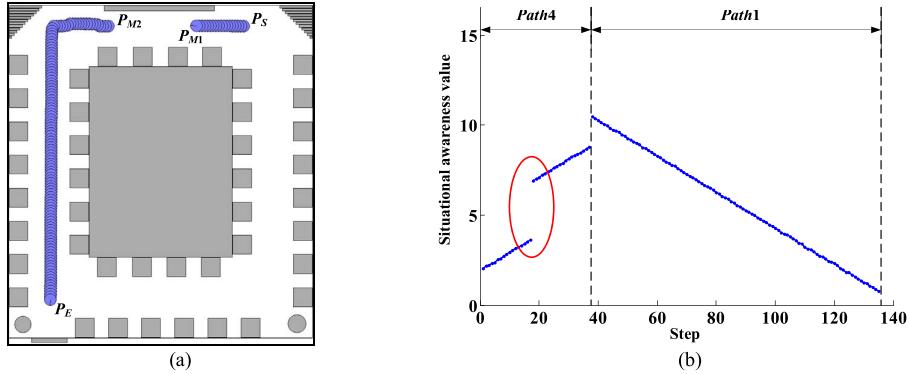


Fig. 11. Results of simulation 5, where the robot is manually moved to a new position. (a) Motion trajectory of the robot. (b) Variation curves of situational awareness values for our method.

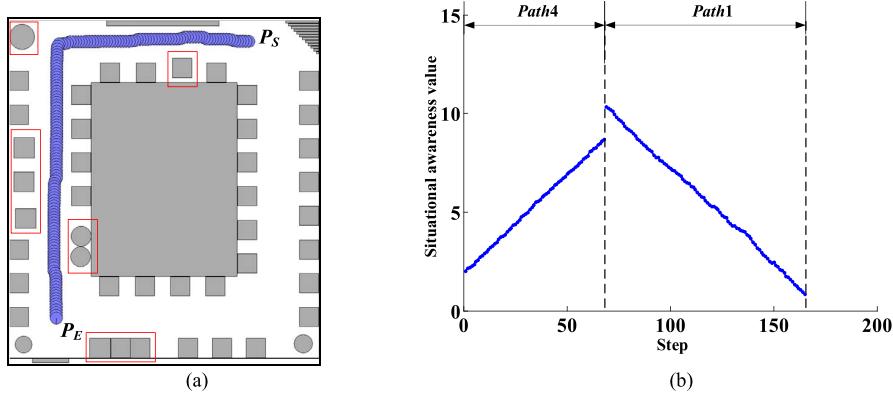


Fig. 12. Results of simulation 6, where the several objects are changed (see the red rectangle marks). (a) Motion trajectory of the robot. (b) Variation curves of situational awareness values for our method.

Algorithm 2 Global Navigation

Input: The graph G , the start node V_s , and the end node V_d .
Output: The optimal nodes sequence χ^* and the best direction α^* .

$$1. \chi^* \leftarrow \arg \min_{\chi \in \Omega_{sd}} \sum_{q=1}^{N_\chi-1} \left| \mathbb{Q}_v \Big|_{V_{q+1}(\chi)} - \mathbb{Q}_v \Big|_{V_q(\chi)} \right|;$$

2. **for** $q=1$ **to** $N_{\chi^*}-1$ **do**

3. $\mathbb{Q} = \text{Path}(V_q(\chi^*), V_{q+1}(\chi^*))$;

4. **if** $\mathbb{Q}_v|_X \notin [\mathbb{Q}_v|_{p_c} - \eta, \mathbb{Q}_v|_{p_c} + \eta]$ **then**

5. **if** ($M_f=1$) **then** $\overrightarrow{XX^*} \leftarrow \overrightarrow{X} \frac{\text{argmin}_v v|_{\tilde{X}}}{\|X-\tilde{X}\|_2 \leq \lambda}$;

6. **else**

7. $\overrightarrow{XX^*} \leftarrow \overrightarrow{X} \frac{\text{argmax}_v v|_{\tilde{X}}}{\|X-\tilde{X}\|_2 \leq \lambda}$;

8. **end if**

9. $\alpha^* \leftarrow \text{Collision_Avoidance}(\overrightarrow{XX^*})$;

10. **else if** $\mathbb{Q}_v|_X \in [\mathbb{Q}_v|_{p_c} - \eta, \mathbb{Q}_v|_{p_c} + \eta]$ **then**

11. **if** ($q < N_{\chi^*} - 1$) **then**

12. $q \leftarrow q+1$; **continue**;

13. **else if** ($q = N_{\chi^*} - 1$) **then**

14. **break**;

15. **end if**

16. **end if**

17. **end for**

18. **return**

In the following, the structure of the situational awareness-fitting network is chosen as $[1440, 6m, 3m, m, 1]$ where m is the parameter. The traditional solution is to try many times to

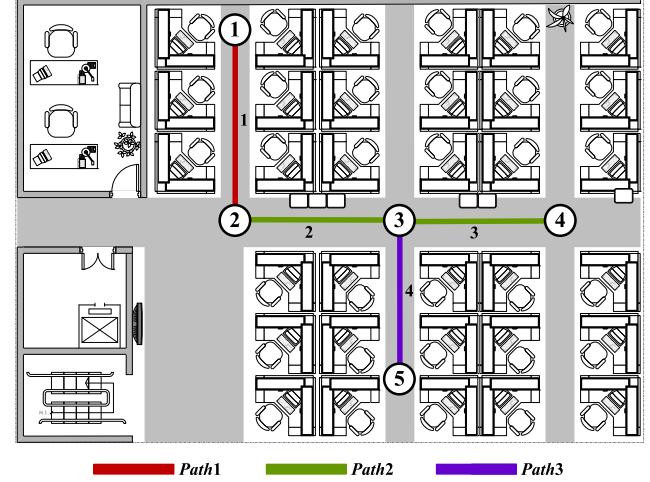


Fig. 13. Schematic of the experiment environment as well as the topological map.

choose a good one, which leads to a tedious process. To solve this problem, we endeavor to explore the relationship between the amount of perception I and the parameter m . From the above simulation, we know that the amount of perception I can be used to measure the complexity of path. In this article, we select nine typical environments to determine the relationship between the amount of perception I and the parameter m , as shown in Fig. 5. It is seen that with the increasing of I ,

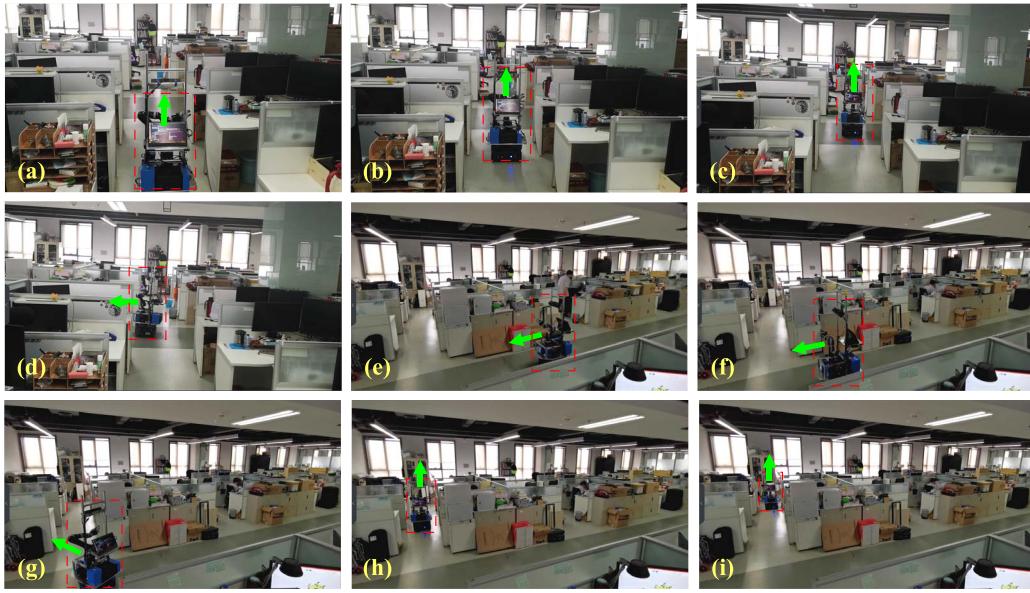


Fig. 14. Video snapshots of experiment 1.

the values of m are growing almost linearly. And the relationship between I and m may be described by $m = 6.6 \times 10^{-3}I - 15$. Afterward, the parameter m of a situational awareness fitting network may be approximately obtained.

B. Results of the Navigation

In this part, three simulations are conducted to verify our proposed navigation method. Note that the environments of these simulations are different from the aforementioned nine environments.

Simulation 2 considers an office environment, as shown in Fig. 6(a). Fig. 6(b) demonstrates the corresponding topological map with 13 nodes and 16 edges, and there are seven paths all together where the direction of the arrow refers to the decreasing one of the situational awareness value. The robot is required to be moved from its starting position P_S to destination P_E based on the established situational awareness fitting networks, and the navigation trajectory is given in Fig. 7. The variation curves of situational awareness values estimated by our method and ground truth are illustrated in Fig. 8(a), and the curve of their error δ is depicted in Fig. 8(b), where the ground truth may be obtained by the odometer. It is seen that the robot moves along $Path1 \rightarrow Path2 \rightarrow Path4 \rightarrow Path7$, and finally arrives at the destination.

The results of simulation 3 are presented in Fig. 9 with an conference room environment. Fig. 9(a)–(d) describes the motion trajectory of the robot, the topological map, the variation curves of situational awareness values for our method and ground truth, and the curve of the error δ , respectively. In simulation 4, a cluttered environment is considered and the results are depicted in Fig. 10. It is seen that the navigation can be completed.

C. Robustness Verification

Simulation 5 is conducted where the robot is manually moved to a new position, which is used to testify the

robustness of the proposed method that relies on the situational awareness values of each path and a topological map. Fig. 11(a) shows the motion trajectory of the robot. When the robot arrives at the position P_{M1} , it is directly moved to a new position P_{M2} . It is seen from Fig. 11(b) that there exists a sudden change on situational awareness value. Still, the robot can continue to move toward the destination.

Simulation 6 considers the case where several objects are changed either in position or in shape, as indicated in the red rectangle marks in Fig. 12. We can see that the robot completes the navigation using the proposed method.

VI. EXPERIMENTS

In the following, the experiments are conducted in an office environment. Fig. 13 demonstrates the experimental topological map with five nodes and four edges. Accordingly, there are three paths: 1) $Path1$; 2) $Path2$; and 3) $Path3$, which are represented in red, green, and purple lines, respectively.

In experiment 1, the starting position P_S and destination P_E are set nearby node 5 and node 1, respectively. The video snapshots are shown in Fig. 14. The robot moves from P_S in Fig. 14(a). After it gets to the node 3 along $Path3$ [see Fig. 14(b)–(c)], it turns left and continues to march along $Path2$. Afterwards, the robot turns right and finally it arrives at the position in Fig. 14(i). It is shown that the proposed method enables the robot to switch between different paths to complete the task.

Experiment 2 considers an artificial disturbance during the robot motion and the results are depicted in Fig. 15, where P_S is located between the nodes 3 and 4, and P_E is nearby node 1. The robot starts its motion from P_S . When it moves to the position in Fig. 15(c) along $Path2$, it is forced to pause all programs. Then it is manually dragged to the position in Fig. 15(d) and the programs restore normal. Although this disturbance causes a sudden change on situational awareness

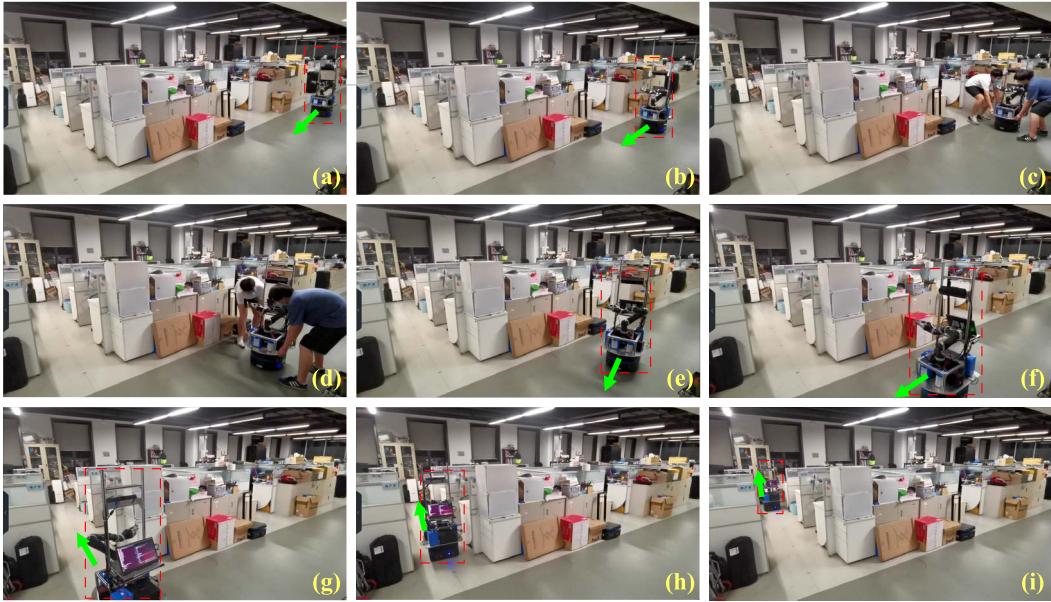


Fig. 15. Video snapshots of experiment 2.

value, from Fig. 15(e)–(i), the robot can still continue to move and fulfill its task smoothly.

VII. CONCLUSION

In this article, a novel method based on situational awareness is proposed to achieve the robot navigation, where the situational awareness value is regarded as an implicit expression of position of the robot. With a definition of path as well as the amount of perception, the situational awareness fitting network is introduced as the scene interpreter, which provides a parameterized path model. The proposed method does not rely on the Cartesian-based global position and its effectiveness is verified through simulations and experiments.

REFERENCES

- [1] Y. Kim, J. An, and J. Lee, “Robust navigational system for a transporter using GPS/INS fusion,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 4, pp. 3346–3354, Apr. 2018.
- [2] B. Talbot, F. Dayoub, P. Corke, and G. Wyeth, “Robot navigation in unseen spaces using an abstract map,” *IEEE Trans. Cogn. Develop. Syst.*, early access, May 12, 2020, doi: [10.1109/TCDS.2020.2993855](https://doi.org/10.1109/TCDS.2020.2993855).
- [3] H. J. Tang, R. Yan, and K. C. Tan, “Cognitive navigation by neuro-inspired localization, mapping, and episodic memory,” *IEEE Trans. Cogn. Develop. Syst.*, vol. 10, no. 3, pp. 751–761, Sep. 2018.
- [4] W. H. Chin, Y. Toda, N. Kubota, C. K. Loo, and M. Seera, “Episodic memory multimodal learning for robot sensorimotor map building and navigation,” *IEEE Trans. Cogn. Develop. Syst.*, vol. 11, no. 2, pp. 210–220, Jun. 2019.
- [5] O. Montiel, R. Sepúlveda, and U. Orosco-Rosas, “Optimal path planning generation for mobile robots using parallel evolutionary artificial potential field,” *J. Intell. Robot. Syst.*, vol. 79, no. 2, pp. 237–257, 2015.
- [6] H. Moravec and A. Elfes, “High resolution maps from wide angle sonar,” in *Proc. IEEE Int. Conf. Robot. Autom.*, St. Louis, MO, USA, 1985, pp. 116–121.
- [7] S. T. Pfister, S. I. Roumeliotis, and J. W. Burdick, “Weighted line fitting algorithms for mobile robot map building and efficient data representation,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, 2003, pp. 1304–1311.
- [8] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “OctoMap: An efficient probabilistic 3D mapping framework based on octrees,” *Auton. Robots*, vol. 34, no. 3, pp. 189–206, 2013.
- [9] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, “3-D mapping with an RGB-D camera,” *IEEE Trans. Robot.*, vol. 30, no. 1, pp. 177–187, Feb. 2014.
- [10] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM: A versatile and accurate monocular SLAM system,” *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.
- [11] G. Zhang, J. H. Lee, J. Lim, and I. H. Suh, “Building a 3-D line-based map using stereo SLAM,” *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1364–1377, Dec. 2015.
- [12] X. Zuo, X. Xie, Y. Liu, and G. Huang, “Robust visual SLAM with point and line features,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Vancouver, BC, Canada, 2017, pp. 1775–1782.
- [13] M. Bosse and R. Zlot, “Continuous 3D scan-matching with a spinning 2D laser,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, 2009, pp. 4312–4319.
- [14] J. Zhang and S. Singh, “LOAM: Lidar odometry and mapping in real-time,” in *Robotics: Science and Systems*, Berkeley, CA, USA, 2014.
- [15] T. Qin, P. Li, and S. Shen, “VINS-mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.
- [16] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, “ElasticFusion: Real-time dense SLAM and light source estimation,” *Int. J. Robot. Res.*, vol. 35, no. 14, pp. 1697–1716, 2016.
- [17] B. Glocker, J. Shotton, A. Criminisi, and S. Izadi, “Real-time RGB-D camera relocalization via randomized ferns for keyframe encoding,” *IEEE Trans. Visual. Comput. Graph.*, vol. 21, no. 5, pp. 571–583, May 2015.
- [18] A. Kendall, M. Grimes, and R. Cipolla, “PoseNet: A convolutional network for real-time 6-DoF camera relocalization,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, 2015, pp. 2938–2946.
- [19] R. Li, Q. Liu, J. Gui, D. Gu, and H. Hu, “Indoor relocalization in challenging environments with dual-stream convolutional neural networks,” *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 2, pp. 651–662, Apr. 2018.
- [20] F. Walch, C. Hazirbas, L. Leal-Taixé, T. Sattler, S. Hilsenbeck, and D. Cremers, “Image-based localization using LSTMs for structured feature correlation,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, 2017, pp. 627–637.
- [21] N. Savinov, A. Dosovitskiy, and V. Koltun, “Semi-parametric topological memory for navigation,” in *Proc. Int. Conf. Learn. Represent.*, 2018.
- [22] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jaśkowski, “ViZDoom: A doom-based ai research platform for visual reinforcement learning,” in *Proc. IEEE Conf. Comput. Intell. Games*, Santorini, Greece, 2016, pp. 1–8.
- [23] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, “Cognitive mapping and planning for visual navigation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 7272–7281.

- [24] Y. Zhu *et al.*, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom.*, Singapore, 2017, pp. 3357–3364.
- [25] Q. Y. Chen and H. W. Mo, "A brain-inspired goal-oriented robot navigation system," *Appl. Sci.*, vol. 9, p. 4869, Nov. 2019.
- [26] M. J. Milford and G. F. Wyeth, "Mapping a suburb with a single camera using a biologically inspired SLAM system," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1038–1053, Oct. 2008.
- [27] M. Milford and G. Wyeth, "Persistent navigation and mapping using a biologically inspired SLAM system," *Int. J. Robot. Res.*, vol. 29, no. 9, pp. 1131–1153, 2010.
- [28] V. A. Shim, B. Tian, M. L. Yuan, H. J. Tang, and H. Z. Li, "Direction-driven navigation using cognitive map for mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Chicago, IL, USA, 2014, pp. 2639–2646.
- [29] Y. Y. Yu, Z. Y. Wu, Z. Q. Cao, L. Pang, L. Ren, and C. Zhou, "A laser-based multi-robot collision avoidance approach in unknown environments," *Int. J. Adv. Robot. Syst.*, vol. 15, no. 1, pp. 1–10, 2018.



Xilong Liu received the B.S. degree from Beijing Jiaotong University, Beijing, China, in 2009, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 2014.

He is currently an Associate Professor with the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences. His current research interests include image processing, pattern recognition, visual measurement, and visual scene cognition.



Zhiqiang Cao (Senior Member, IEEE) received the B.S. and M.S. degrees from the Shandong University of Technology, Jinan, China, in 1996 and 1999, respectively, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2002.

He is currently a Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include service robot and multirobot coordination.



Yingying Yu received the B.S. degree from the University of Science and Technology Beijing, Beijing, China, in 2015. She is currently pursuing the Ph.D. degree in control theory and control engineering with the Institute of Automation, Chinese Academy of Sciences, Beijing.

Her current research interests include navigation and manipulation of service robot.



Guangli Ren received the B.S. degree in intelligent science and technology from Dalian Maritime University, Dalian, China, in 2015, and the M.S. degree in technology of computer application from the Capital Normal University, Beijing, China, in 2018. He is currently pursuing the Ph.D. degree in control theory and control engineering with the Institute of Automation, Chinese Academy of Sciences, Beijing.

His current research interests include visual SLAM and robotic manipulation.



Junzhi Yu (Fellow, IEEE) received the B.E. degree in safety engineering and the M.E. degree in precision instruments and mechanology from the North University of China, Taiyuan, China, in 1998 and 2001, respectively, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2003.

From 2004 to 2006, he was a Postdoctoral Research Fellow with the Center for Systems and Control, Peking University, Beijing. He was an

Associate Professor with the Institute of Automation, Chinese Academy of Sciences, in 2006, where he was a Full Professor in 2012. In 2018, he joined the College of Engineering, Peking University, as a tenured Full Professor. His current research interests include intelligent robots, motion control, and intelligent mechatronic systems.



Min Tan received the B.E. degree from Tsinghua University, Beijing, China, in 1986, and the Ph.D. degree in control theory and control engineering from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 1990.

He is currently a Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include advanced robot control, biomimetic robot, and multirobot system.