Biomimetic Intelligence and Robotics 5 (2025) 100203

Contents lists available at ScienceDirect

Biomimetic Intelligence and Robotics

journal homepage: www.elsevier.com/locate/birob

Review

A survey of autonomous robots and multi-robot navigation: Perception, planning and collaboration

Weinan Chen [a,1,2], Wenzheng Chi [b,1], Sehua Ji [a], Hanjing Ye [d], Jie Liu [b], Yunjie Jia [c], Jiajie Yu [b], Jiyu Cheng [e,*,2]

[a] State Key Laboratory of Precision Electronic Manufacturing Technology and Equipment, Guangdong University of Technology, Guangzhou 510006, China
[b] Robotics and Microsystems Center, School of Mechanical and Electric Engineering, Soochow University, Suzhou 215137, China
[c] School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai 264209, China
[d] Shenzhen Key Laboratory of Robotics and Computer Vision, Southern University of Science and Technology, Shenzhen 518055, China
[e] School of Control Science and Engineering, Shandong University, Jinan 250061, China

ARTICLE INFO

Article history:
Received 30 September 2024
Revised 14 November 2024
Accepted 22 November 2024
Available online 7 December 2024

Keywords:
Autonomous robots
Planning
Perception
Collaboration

ABSTRACT

The development of autonomous robots and the wide range of communication resources hold significant potential for enhancing multi-robot collaboration and its applications. Over the past decades, there has been a growing interest in autonomous navigation and multi-robot collaboration. Consequently, a comprehensive review of current trends in this field has become crucial for both novice and experienced researchers. This paper focuses on automation systems and multi-robot navigation to support their operations. The review is structured around three potential benefits: perception, planning, and collaboration. This review has systematically explored a broad spectrum of autonomous robots and multi-robot navigation strategies with over 170 references. Also, we point out the challenges of the existing work, as well as the development direction. We believe that this review can build a bridge between autonomous robots and their applications.

1. Introduction

Autonomous robots play an essential role in various applications, in which perception and planning are the key points to achieve autonomy. In addition to a single robot, swarm intelligence has been verified to have the ability to collaborate efficiently. This review aims to provide an overview of autonomous robots in terms of perception, planning, and collaboration. The topic has gained significant attention and has been extensively explored by various communities, all with the goal of enabling autonomous agents in real-world applications. However, the diversity of approaches and theories has led to a fragmented research landscape, hindering potential collaborations between different research lines. Through this review, we aim to bridge this gap by presenting a comprehensive overview of the field and highlighting the challenges it faces.

The perception of autonomy aims to understand surrounding environments and the robot itself on multiple levels, e.g., at the geometric and semantic levels. Scene understanding is a part of perception that aims to enable computers to comprehend and interpret the world in a manner akin to human cognition. Humans typically perceive and understand their environment through rich multi-modal inputs, including but not limited to vision, language, audio, and touch. Given that vision is primarily utilized in daily human decision-making, our focus is on algorithms pertaining to this modality. In this context, we review a substantial body of work from various communities and introduce a novel taxonomy. We categorize the state-of-the-art methods, discussing their typical properties, advantages, and drawbacks, and we highlight open challenges for future research. Also, localization and mapping are crucial for achieving robot autonomy and have been widely applied and studied. Accurate pose information is especially vital for the efficiency of inter-robot collaboration in swarm robotics. We provide a review of localization and mapping, discuss the classification of localization methods based on scene information obtained from different sensors, and analyze different forms of mapping methods.

In the field of planning, effective planning is essential for determining collision-free paths that meet specific constraints. As robots become increasingly autonomous, continuous advancements in planning methods, such as dynamic replanning and optimization-based techniques, play a critical role. By surveying

* Corresponding author.
E-mail address: jycheng@sdu.edu.cn (J. Cheng).
1 Weinan Chen and Wenzheng Chi contribute equally to this paper.
2 Given their role as Guest Editor of this journal, Weinan Chen and Jiyu Cheng had no involvement in the peer-review of this article and had no access to information regarding its peer-review. The article was handled by Prof. Rui Song.

https://doi.org/10.1016/j.birob.2024.100203

various approaches, we aim to provide a comprehensive overview of the state-of-the-art methods and highlight emerging trends and challenges in the field of autonomous robot planning. Additionally, in the field of multi-robot collaboration and planning, the dynamic integration and interpretation of data from multiple sources are critical to orchestrating effective and synchronized movement among robots. By leveraging multi-sensor perceptual information, communication protocols, and collaborative algorithms, robots can seamlessly navigate complex environments, execute collaborative tasks, and maintain formation integrity. Effective multirobot planning not only enhances operational efficiency, but also expands the scope of tasks that can be performed in parallel, significantly increasing the versatility and utility of robotic swarms in various applications. Additionally, we examine the prevailing challenges in this field and outline potential future directions for advancement.

While numerous surveys comprehensively address single-robot navigation and autonomous systems, there remains a significant gap in the literature regarding multi-robot collaboration and navigation. As the demand for autonomous robot applications increases, the limitations of single-robot models in meeting task objectives across diverse and challenging operational environments have become increasingly apparent. Consequently, the multi-robot collaboration model is gaining significant attention in the field. However, many survey papers fail to adequately address the critical role of multi-robot collaboration within autonomous robotic systems in their statistics and analyses. For example, the survey presented in [1] focuses primarily on the impact of deep learning on single-robot systems, resulting in a constrained analytical perspective. Similarly, [2] restricts its analysis to the performance of a LiDAR-only SLAM system, thereby neglecting broader implications. While [3] offers a comprehensive discussion on mobile robots and the principal sensors utilized in robotics over nearly a century, it lacks an examination of multi-robot collaborative scenarios, limiting the applicability of its findings. In recent years, multi-robot collaboration has emerged as a prominent research focus, with numerous significant contributions documented in various survey papers. For instance, the relationship between multi-robot collaboration and motion planning is explored in [4], while [5] addresses its connection to localization. Additionally, [6] and [7] concentrate on the applications of deep learning within multi-robot collaboration frameworks. However, these studies primarily emphasize comparative analyses of different multi-robot collaboration approaches or their associations with specific functionalities. They often overlook a more comprehensive examination of the role of robot autonomy in enhancing the effectiveness of multi-robot collaboration strategies.

Single-robot surveys typically focus on individual path planning, obstacle avoidance, and sensor integration, providing in-depth analyses of algorithms and technologies tailored to solitary operations. However, these studies often overlook the complexities introduced by multiple interacting agents, such as inter-robot communication, coordination strategies, scalability issues, and collective decision-making processes. Additionally, single-robot frameworks do not adequately address the dynamic and emergent behaviors that arise in multi-robot systems, nor do they explore the synergistic advantages of cooperative navigation and swarm intelligence. This gap underscores the need for dedicated multi-robot surveys that systematically examine the unique challenges and advancements in coordinating multiple autonomous robots, thereby advancing the field towards more robust, efficient, and scalable multi-robot applications.

In this paper, we comprehensively examine the relationship between autonomous robots and multi-machine collaboration methods, as well as the future development prospects of both, particularly in the areas of navigation, perception, and planning for autonomous robots. The logical framework of this paper is shown in Fig. 1.

Table 1
Scene understanding techniques for autonomous mobile robot navigation.

Level	Method	References
Appearance-level	Local features	[8–17]
	Global features	[18–25]
Geometry-level	3D points/depth estimation	[26–30]
Semantic-level	Object detection	[31–41]
	Multiple object tracking	[42–46]
	Semantic segmentation	[47–52]

2. Perception

In the perception aspect of autonomous robots, we focus on sensor-based scene understanding, along with robot localization and mapping.

2.1. Scene understanding

We categorize scene understanding algorithms into data representation and data association. Data representation focuses on single-frame understanding and is subdivided into three levels: appearance, geometry, and semantics. Data association, on the other hand, focuses on multiple-frame matching to determine which representations correspond to the same object. In this paper, we mainly discuss scene understanding methods most related to downstream robotics tasks, e.g., SLAM and path planning. Appearance-level representation includes local keypoint detection and feature description for tasks such as robot pose estimation, as well as global features for visual place recognition. Geometry-level representation includes geometric attributes in a scene, mainly 3D points discussed in this paper. Semantic-level representation, as shown in Fig. 2, covers object detection and semantic segmentation.

2.1.1. Appearance-level representation

Appearance-level representation can be categorized into two main types: local features and global features. Local features involve selectively extracting interesting or notable parts of an image, typically requiring a detector to identify these significant elements. These feature-matching results are essential for robot pose estimation. Global features, on the other hand, represent the overall content or scene depicted in the images. In most cases, they are used for *place recognition*, determining whether two images depict the same location. Appearance-level representation, incorporating both local and global features, facilitates accurate association at point and place levels. Correctly associated points across frames establish constraints essential for estimating the robot's relative pose, while place-level associations aid in map-based location recognition. These functions are critical for loop closure detection in SLAM, reducing localization drift, and are also valuable for robot re-localization (see Table 1).

Local Features: Traditional approaches to local features rely on hand-crafted designs involving detection and description. ORB-SLAM3 [53] uses FAST [8] for detection and BRIEF [9] for description, while Mei et al. [10] combined FAST with SIFT [11]. Brute force matching is commonly used for associating features, and while hand-crafted features are efficient and robust, they struggle with changes in viewpoint, lighting, and scale [10,53]. Recently, deep neural networks have advanced local feature representations to handle severe condition changes. SuperPoint [12] uses self-supervised Homographic Adaptation for joint detection and description, facilitating transfer from synthetic to real-world images. DISK [13] applies reinforcement learning for denser detections. SuperGlue [14] employs a graph convolutional network for feature matching, achieving accurate pose or homography estimation.

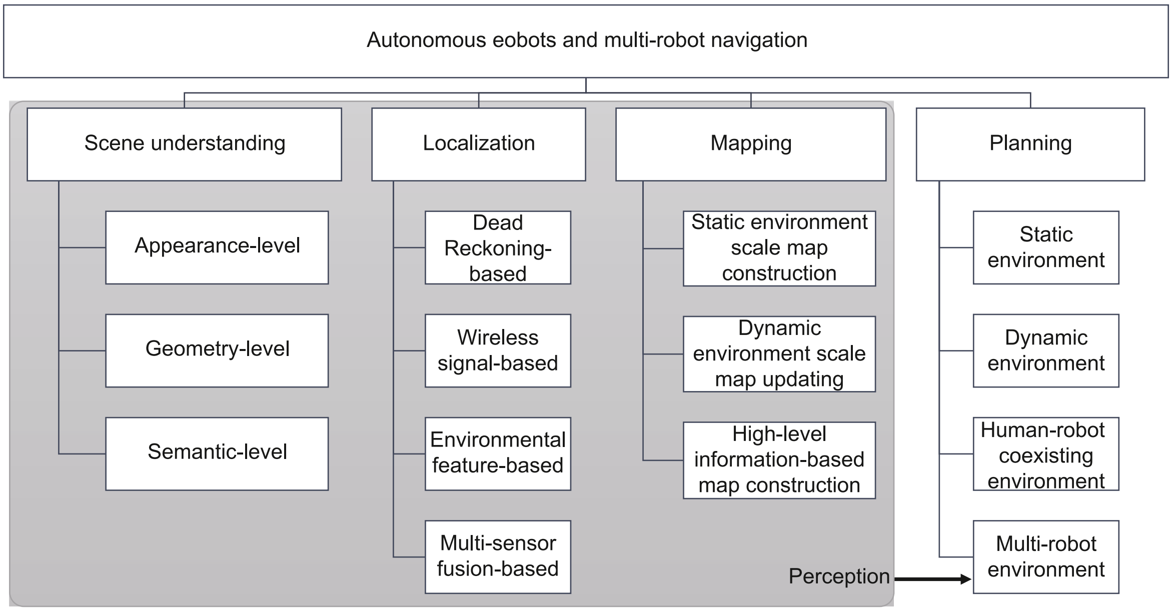

Fig. 1. A categorization of Autonomous Robot Methods in the paper.

Table 2
Local feature extraction and matching techniques for autonomous mobile robot navigation.

Method	Category	Architecture	Training datasets	Comments
FAST [8] + BRIEF [9]	Detector	Handcrafted-based	–	Fast inference and widely employed in SLAM systems, e.g., ORBSLAM3 [53] but sensitive to viewpoint changes
FAST [8] + SIFT [11]	Detector	Handcrafted-based	–	Fast inference and viewpoint-robust but sensitive to large-scale appearance and viewpoint changes
SuperPoint [12]	Detector	DL-based; CNN-based encoder and two keypoint and descriptor heads	Synthetic Shapes + MSCOCO [55]	Self-supervised adaptation by jointly learning detection and description; Facilitating the transfer of knowledge from synthetic datasets to real-world images
DISK [13]	Detector	DL-based; CNN-based encoder and two keypoint and descriptor heads	MegaDepth [56]	Reinforcement-learning-based optimization
SuperGlue [14]	Matcher	DL-based; GCN + Attention Layers	ScanNet [57]	GCN-based feature matching
MESA [17]	Matcher	DL-based; Markov random field and Bayesian network	–	First utilizing SAM [51] to construct an semantic-level area-graph for filtering unreliable feature matching
LoFTR [15]	Detector-free	DL-based; CNN-based encoder and transformer-based matcher	MegaDepth [56] + ScanNet [57]	First utilizing Transformer's long-range capturing ability to predict semi-dense matches and detections jointly
DKM [16]	Detector-free	DL-based; CNN-based encoder and kernel-based matcher	MegaDepth [56] + ScanNet [57]	Formulating feature matching as a Gaussian Process, yielding accurate dense matching results

Detector-dependent methods generate multiple feature matches, processed by estimating relative pose and using RANSAC to reject outliers, a technique referred to as computer-vision-based. In contrast, detector-free methods like LoFTR [15] use visual Transformers to predict semi-dense matches, while DKM [16] models feature matching as a Gaussian Process for dense matching. Despite the promise of detector-free methods, they still struggle with viewpoint changes. To address this, semantic-cue-based methods like MESA [17] leverage SAM [51] to extract semantically meaningful image areas for coarse-to-fine matching, improving accuracy in detector-free matching like DKM. We summarize existing local feature extraction methods in Table 2. For a deeper review of local feature matching via deep learning, see [54].

Global Features: Global features are essential for place recognition, such as loop closure detection. Traditional methods aggregate local features using techniques like bag-of-words [18], VLAD [19], and Fisher Vector [20]. While effective, these approaches struggle in large-scale environments with significant appearance changes. Recent DL-based methods [21–25] have shown improved performance in achieving condition-invariant place recognition. Typically, these approaches involve feature extraction followed by aggregation. NetVLAD [21] introduces a learnable VLAD module to compact CNN features into global representations. SFRS [24] enhances image-to-region similarity through self-supervised learning, addressing challenges like noisy GPS labels. CAE-VPR [23] uses a convolutional autoencoder to compress high-dimensional CNN features into a low-dimensional space. With the rise of visual foundation models, approaches like CricaVPR [25] adapt these models using multi-scale convolutional techniques for robust place recognition. For further reading on global feature methods, see [58,59].

2.1.2. Geometry-level representation

Geometry-level representation captures the world in a geometric form, which is crucial for tasks like SLAM and path planning. For example, it provides the geometric constraints necessary for applications like path planning and collision avoidance,

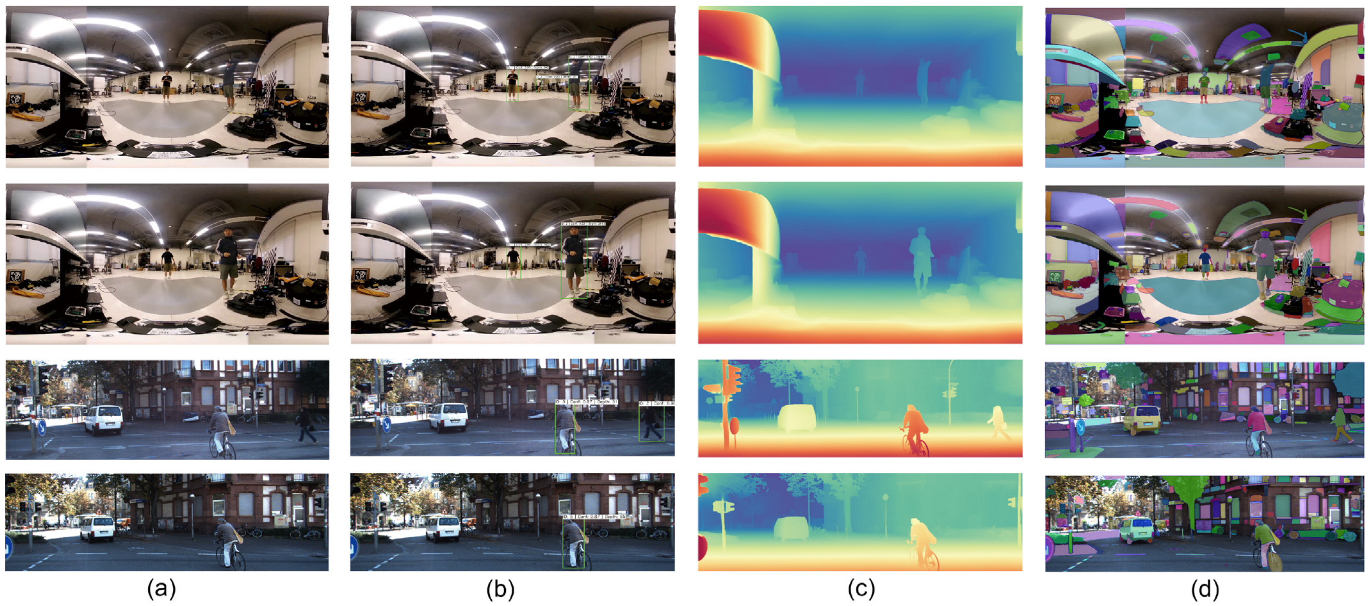

Fig. 2. Examples of scene understanding tasks include object detection using YOLO [36], multiple object tracking with ByteTrack [46], depth estimation by DepthAnything [29], and open-set segmentation via SAM2 [52]. The top two images depict sequential recordings in indoor environments, while the bottom two are outdoor images selected from the KITTI MOT dataset [60]. Copyright©, KITTI. Geometric-level representation (e.g., depth estimation) provides critical constraints for path planning and obstacle modeling. Semantic-level representation (e.g., object detection, tracking, segmentation) enables contextual understanding for social navigation and reduces the impact of dynamic objects on SLAM, enhancing system robustness. (a) Raw image. (b) Detection and multiple object tracking. (c) Depth estimation. (d) Open-set segmentation.

where obstacle representation is crucial. The most basic geometric attribute is the 3D point, from which lines, planes, and surfaces can be derived. In robotics, 3D point representation is fundamental. Range sensors like LiDARs and RGB-D cameras capture 3D points directly but come with limitations: RGB-D cameras are light-sensitive, and LiDARs are costly. Monocular depth estimation offers an alternative, estimating pixel-wise depth from a single image, although this is an ill-posed problem. Early methods used RGB-D and LiDAR sensors to train networks but struggled with generalization due to limited and diverse training data. To overcome this, self-supervision using stereo cameras has been employed. Monodepth [26] introduced a training loss enforcing disparity consistency between stereo images, achieving accurate depth estimation. Monodepth2 [27] improved this with innovations like minimum reprojection loss and auto-masking, enhancing depth accuracy even with occlusions.

MiDaS [28] showed that deep neural networks can adapt to diverse scenes, though limited by the lack of dense ground-truth data. Depth Anything [29] leveraged a visual transformer and pre-trained encoders (DINOv2 [61]) for strong generalization, producing relative depth maps suitable for most applications. For precise metric depth estimation (MDE), fine-tuning pre-trained encoders on metric datasets is a common solution, though this often limits generalization. UniDepth [30] tackles this by using a pseudo-spherical representation to disentangle the camera and depth optimization, improving depth consistency across different scenes and camera setups.

2.1.3. Semantic-level representation

Semantic-level representation understands the world through meaningful abstractions, i.e., object category. This approach aligns with human understanding and facilitates more human-like decision-making, which is beneficial for SLAM and path planning in semantically rich scenarios. This human-like representation enables socially aware path planning, which considers active and socially contextual planning. Specifically, for social navigation in warehouses or hospitals, the robot needs to understand surrounding contextual cues with semantic understanding ability and track people's trajectories. Additionally, it mitigates the impact of dynamic objects within SLAM, enhancing the robustness of the system. For object-category-level recognition necessary for downstream robotics tasks, we mainly review two areas: object detection and semantic segmentation.

Object detection involves identifying and locating objects of interest in an image using 2D bounding boxes. Semantic segmentation, on the other hand, recognizes pixel-wise object categories within an image. Depending on whether we have pre-defined object classes, each of these categories can be further divided into closed-set and open-set recognition. Furthermore, by associating these recognitions across sequential frames, we can achieve object tracking or video object segmentation. In this paper, we primarily review object tracking, as it is essential for collision avoidance and social behavior recognition in path planning.

Object Detection: Object Detection is a fundamental task for robotics applications like SLAM and path planning, enabling tasks such as 3D object mapping and robot localization [62]. Powered by deep learning and large-scale data, object detection methods are commonly divided into region-based, pixel-based, and query-based approaches. Region-based methods follow a two-stage process: object proposals are generated, followed by classification and bounding box regression. R-CNN uses selective search for proposals [31], with Fast R-CNN [32] introducing RoI pooling to speed up the process, and Faster R-CNN [33] incorporating a Region Proposal Network (RPN) for efficiency.

Pixel-based methods like YOLO [34] predict object classes and bounding boxes in a single step, dividing the image into grids for real-time detection, with later versions [35,36] refining the model for better accuracy and speed. Query-based methods [37,38] leverage Transformers to model relationships and global context. DETR [37] frames detection as a set prediction problem, while Deformable DETR [38] improves scale handling and training speed with deformable attention. These closed-set methods are trained on predefined categories, but open-set detection has emerged to identify objects beyond fixed vocabularies. Inspired by vision-language pre-training [63], GLIP [39] expands detection vocabularies using image-text pairs, and Grounding DINO [40] enhances object-text association. YOLO-World [41] further improves real-time performance with a novel pre-training strategy.

Multiple Object Tracking: Multiple Object Tracking can be divided into detection-based and detector-free methods, as well as online and offline approaches. For autonomous robot navigation, we focus on detection-based online tracking, which matches tracklets and detections using similarity metrics. This process typically involves improving the accuracy of matching through better similarity measures or strategies. Location, motion, and appearance are key factors in matching tracklets with detections. SORT [42] assumes linear motion and uses a Kalman filter to predict object locations, matching them with detections using the IoU metric. Some methods adopt non-linear models to handle more complex motion patterns. However, motion-based methods struggle with long-term occlusion, prompting the use of ReID features to capture appearance. DeepSORT [43] uses appearance similarity (cosine similarity) to improve tracking in occlusion-heavy environments, while FairMOT [44] combines detection and ReID in a unified network for enhanced similarity calculation. Matching strategies associate tracklets with new detections using algorithms like the Hungarian Algorithm or greedy assignment. SORT [42] uses a single-step matching process, while DeepSORT [43] introduces cascaded matching, first aligning with recent tracklets and then lost ones. QDTrack [45] employs bidirectional softmax for consistent instance matching, and ByteTrack [46] innovates by using both high- and low-score detection boxes, performing a second association for unmatched tracklets.

Semantic Segmentation: Semantic Segmentation methods primarily use Fully Convolutional Networks (FCNs) with encoder–decoder architectures. These models generate low-resolution features via convolutions and then upsample to create high-resolution segmentation maps. U-Net [47] mirrors the encoder path by upsampling and merging feature maps through skip connections, while DeepLabv3+ [48] uses atrous convolutions to capture multi-scale context while maintaining spatial resolution. However, convolutional operations struggle to capture long-range dependencies, which are crucial for accurate segmentation. To overcome this, Vision Transformers (ViTs) have been introduced. SETR [49] employs a ViT backbone for long-range dependency capture, while Segmenter [50] uses a fully Transformer-based encoder–decoder for improved global context modeling. More recently, prompt engineering from large language models has led to the Segment Anything Model (SAM) [51], which offers promptable segmentation. SAM2 [52] enhances this with a data engine and streaming memory for faster video and image segmentation, providing sixfold speed improvements over the original SAM.

2.2. Localization

Localization is the process by which a robot determines its pose based on sensor perceptual information, enabling the safety and reliability of tasks such as planning and navigation. Commonly used sensors include LiDAR, cameras, IMUs, and GPS, among others.

2.2.1. Dead reckoning-based methods

The term "Dead Reckoning Localization(DR-L)" is derived from the nautical technique known as "deductive projection". These methods estimate the robot's pose by using a known initial position and integrating incremental motion measurements from sensors such as encoders, gyroscopes, and accelerometers. Among these, the Inertial Measurement Unit(IMU) integrates accelerometers, gyroscopes, magnetometers, and other sensors. Its advantages—such as high frequency, compact size and weight, and the ability to provide autonomous measurements without relying on external reference—have led to a wide range of applications. For instance, [64] proposes an IMU-based indoor localization system on a microcontroller. Furthermore, there has been a growing research focus on deep learning-based IMU localization systems, such as [65] and [66]. Beyond mobile robots, IMUs are also used in domains like human-footed robots [67] and fire rescue [68], providing critical information for strategic planning and coordination.

Wheel odometry and track odometry are two common techniques based on encoders used for localization and navigation. Wheel odometry estimates the pose of the robot by integrating the wheel's angular and rotational speed, as demonstrated by methods proposed in [69] and [70]. On the other hand, track odometry is designed for tracked robots or agricultural machinery. It utilizes encoders placed on the tracks to monitor the movement and estimate the robot's displacement and rotation [71].

The methods described above have their own advantages and limitations. Encoder-based wheel odometry is simple, easy to implement, and well-suited for planar mobile robots, but it is prone to cumulative errors caused by ground friction. Track-type odometry, on the other hand, is ideal for tracked robots and those operating in complex terrains. However, its accuracy can be compromised by track slip and other environmental factors. The IMU-based method, while useful for providing orientation data, is susceptible to issues such as sensor drift, the accumulation of errors, and high sensitivity to external interference, which can degrade its accuracy over time.

2.2.2. Wireless signal-based methods

Wireless signal-based localization(WS-L) leverages the broadcast characteristics of wireless signals and parameters such as signal strength, arrival time, or phase to estimate location. These methods are widely applied in various scenes. For instance, WiFi signal-based [72] and [73] enable decimeter-level localization, making them suitable for autonomous indoor mobile robots with extensive coverage areas. In more localized indoor environments, Bluetooth-based methods [74] are often preferred. However, limited accuracy and susceptibility to multipath effects and signal attenuation can affect the performance of these methods. Ultra-wideband (UWB)-based methods use wireless technology to transmit narrow, high-speed pulse signals for localization and tracking [75]. With much wider bandwidth compared to traditional methods, UWB signals achieve centimeter-level precision, making them highly effective in challenging environments [76]. However, this method incurs high costs and requires dense deployment of base stations to meet large-scale positioning needs, which can also limit the range of operations.

To enable reliable positioning in large outdoor areas, [77] and [78] proposed a GPS-based localization method, which uses a network of satellites and ground receivers to provide global positioning services. However, GPS signals can be weak or lost when obstructed, necessitating the fusion of additional sensor data to maintain accurate positioning. This sensor fusion approach will be discussed in detail in the fourth section.

2.2.3. Environmental feature-based methods

Environmental feature-based localization(EF-L) uses sensors to extract and analyze special markers or geometric structural features for precise localization. Commonly used sensors include LiDAR and cameras, which provide richer information for localization in terms of scene understanding.

LiDAR is renowned for its excellent ranging performance, capable of generating rich 3D point cloud of *geometry-level* data.

For instance, [79] and [80] present real-time localization methods based on the detection and description of *local features* using 2D LiDAR and 3D LiDAR, respectively, but these methods are prone to cumulative errors. Additionally, *global feature-based* method [81] employs real-time point cloud to match with a prior 3D point cloud map for accurate localization. However, significant changes between the map and the real world can compromise the accuracy and stability of localization. Moreover, [82] demonstrates that LiDAR can identify varying point cloud intensity information reflected from different object materials, providing innovative approaches for solving localization problems. However, LiDAR-based methods are highly dependent on environmental features, and when dealing with large amounts of point cloud data, they may encounter significant challenges related to storage, transmission, and real-time processing.

Visual sensor-based methods utilize image processing algorithms to analyze and extract features for efficient pose estimation of robots. For example, the method proposed in [83] is well-suited for scenes with high real-time requirements. To address challenges posed by extensive datasets, complex scenes, and abstract features, [84] proposed an end-to-end deep learning method to enhance model training and extract comprehensive scene understanding from images. Additionally, [85] demonstrated a method for localization using visual sensors to identify landmarks. However, these methods are easily limited by issues related to illumination and missing texture.

2.2.4. Multi-sensor fusion-based methods

Although each sensor offers unique advantages, standalone usage often suffers from significant cumulative errors and an inability to meet diverse localization needs. Therefore, many approaches enhance algorithm robustness by fusing information from multiple sensors. For instance, the combination of IMU and wheel encoder in [86] and [87] mitigates IMU's accumulation error and the encoder's slip error. To address the challenges in GPS-denied scenes, the strategies that fuse GPS with IMU, LiDAR, and camera data are presented in [88,89] and [90], respectively. In addition, the camera-IMU method proposed in [91] and the LiDAR-IMU proposed in [92] effectively enhance the localization performance in complex environments. To enhance the richness of map information, [93] proposes an Automatic Vision-LiDAR Calibration (AVLC) method designed to minimize errors arising from unforeseen sensor variations. As the number of sensors increases, the computational cost also grows. To mitigate this limitation, [94] proposes a lightweight SLAM method for optical flow tracking, utilizing pyramid IMU prediction to enhance efficiency. On this basis, a centralized multi-robot collaboration system is introduced based on a robot-edge-cloud layered architecture. This system effectively overcomes the challenges of limited onboard computing resources and the low execution efficiency typically associated with a single robot.

Multi-sensor fusion enhances localization by providing complementary and diverse environmental-aware data to correct errors, improving both robustness and accuracy. However, this method also increases the demand for computing resources, necessitating a balance between accuracy and time efficiency in its application. To intuitively compare the performance of all topics, we summarize and analyze them in Table 3.

2.3. Mapping

Mapping is the process of transforming real-world objects, locations, or scenes into digital representations or visual forms. It involves using of sensors, measurement tools, or other technologies to acquire data and convert it into maps, models, images, or other formats for analysis, navigation, visualization, or other applications. In autonomous mobile robots, accurate maps provide prior static information about the environment for localization. As shown in Fig. 3(a), semantic maps are constructed by associating and analyzing environmental information from LiDAR and cameras [95], enabling the robot to achieve more precise localization and navigation [96], as demonstrated in Fig. 3(b). In this section, we discuss mapping in terms of classification based on map representation.

2.3.1. Static environment

Common mapping methods typically assume a static environment, collecting and extracting sensor data to accurately represent the environmental structure. Depending on the application scenario, maps are described in different ways. For instance, the raster map method proposed in [97] effectively mitigates the impact of noise on feature data extraction during localization, making it well-suited for large-scale outdoor localization, as validated in [98]. However, its accuracy is limited by low resolution, making it struggle to provide more detailed localization information. Additionally, vector maps that enable accurate geometric feature matching were proposed in [99]. However, lacking elevation data, this method is only applicable for localization in indoor planar scenes.

3D point cloud map is another form of scene representation that provides valuable prior information for localization. For instance, in [81] and [100], the use of LiDAR and camera matching with prior 3D maps has been proposed to achieve precise localization. However, as the localization area expands, the volume of data stored in 3D point cloud maps increases significantly, leading to higher demands on computing resources for storage, transmission, and algorithm execution.

Furthermore, the localization capabilities of these methods are confined to areas represented in the prior map. Once the robot navigates beyond the known region, localization becomes ineffective. To improve the scalability of maps, researchers have proposed fusion methods for raster maps, vector maps, and point cloud maps, as discussed in [101,102], and [103], respectively.

2.3.2. Dynamic environment

In practice, if the map is not updated promptly when the environment changes, it can result in erroneous information that reduces localization accuracy. Therefore, static assumptions have inherent limitations, particularly in dynamic environments. To address this, [104] proposed an offline map update method using crowdsourced data. However, this approach suffers from high maintenance costs, slow update speeds, and challenges in maintaining data source consistency. Additionally, [105] introduced an incremental map update method that updates local maps by detecting environmental changes in real-time. However, this method still faces issues such as high latency, low accuracy in highly dynamic scenarios, and significant computational resource consumption when dealing with large-scale environments.

To achieve more efficient map updates, [106] proposed a Simultaneous Localization and Mapping (SLAM) method that integrates dynamic object detection, mitigates unstable environmental factors, and uses real-time updated local maps for inter-frame feature matching, without relying on prior maps, iteratively optimizing localization and mapping accuracy.

However, in scenes with sparse geometric features or low texture, SLAM methods that rely on environmental geometric features struggle to capture adequate information, leading to incomplete maps and significant localization errors. To address this issue, [107] and [108] proposed degeneracy detection methods for analyzing and filtering LiDAR and camera data to identify potential issues or anomalies in the data.

Table 3
Summary of localization methods.

Topic	Sensors (References)	Robustness	Accuracy	Real time	Feature
DR-L	IMU [64]-[68]	General	Medium	Great	High frequency, but overly sensitive
	Encoder [69]-[71]	Low	Medium	Good	Easy to implement, but low robustness
WS-L	WiFi [72,73]	General	General	Medium	Only applicable indoors
	Bluetooth [74]	Low	Low	Good	Low accuracy and susceptible to signal interference
	UWB [75,76]	Medium	High	Excellent	High precision, but limited range of activity
	GPS [77,78]	Medium	General	Great	Widely applicable, but easily affected by occlusion
EF-L	LiDAR [79]-[82]	Medium	High	Slow	High precision, but highly rely on geometric features
	Camera [83]-[85]	General	Low	Medium	Rich scene information, but easily affected by lighting
Multi-sensor	IMU, Encoder, GPS, LiDAR, Camera [86]-[94]	High	High	Slow	High precision and robustness, but high computational cost

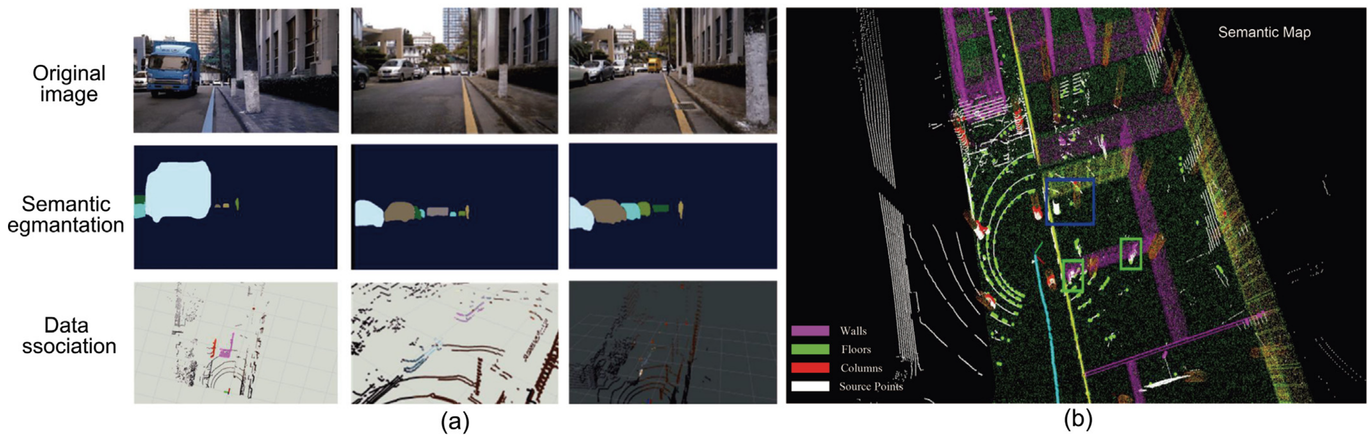

Fig. 3. (a) Data representation and data association. From top to bottom, raw images, images' semantic segmentation, and the semantic point cloud information—obtained by associating LiDAR data with the image's semantic information—are displayed. Copyright©, IEEE ROBIO. (b) Illustrates the semantic map and the localization effect. Copyright©, Automation in Construction.

2.3.3. High-level information mapping

High-level information-based methods use abstract representations to describe the environment. By using these semantic-level representations, we can decompose the environment into various semantic units—such as rooms, doors, and furniture—to enhance the comprehensibility and usability of the maps. For example, [109] proposes a method for constructing semantic local maps using image sequences and wheel inertial self-motion to describe environments, providing higher-level matching information for localization. In addition, [110] introduces a topological map based on the spatial relationships between objects and topological structures, allowing for efficient search, localization, and path planning without relying on geometric information.

However, topological maps are limited regarding localization accuracy and flexibility because they lack absolute pose information and are constrained by predefined topologies. For semantic maps, subjectivity and ambiguity can lead to localization errors and higher computational costs, limiting their reliability and effectiveness in practical applications. By combining the advantages of these two methods, [111] proposes a map representation that integrates semantic information with a topological structure to achieve more robust location identification. However, it comes with high computational complexity.

In autonomous robots, mapping plays a crucial role, enabling the robot to understand its surrounding environment and providing valuable information for path planning, obstacle avoidance, localization, and decision-making. However, in practical applications, the choice of mapping method should be adapted to the specific requirements and scenarios of the application. We have summarized all mapping methods in Table 4 for ease of analysis.

3. Planning

In mobile robots, localization and mapping establish the foundation for autonomy. However, achieving autonomous operation in complex environments also requires the critical component of path planning. Path planning allows robots to determine optimal routes based on environmental maps, ensuring efficient navigation and formulating mobility strategies. Based on the evolution of planning algorithms and the complexity of the scenarios, planning methods can be roughly categorized into four types: static environments, dynamic environments, human–robot coexisting environments, and multi-robot environments. In static environments, path planning generally assumes that obstacles and layouts are fixed, focusing on planning efficiency. For dynamic environments, the emphasis is on the ability of the algorithm to adjust paths in real-time to adapt to moving obstacles or changing layouts. Planning needs to consider human safety, behavior, and interaction in human–robot coexisting environments. In multi-robot environments, coordination among multiple robots is necessary to avoid collisions and optimize collaboration.

3.1. Static environment

Path planning algorithms in static environments have reached a high level of maturity, with classical approaches primarily divided into graph-based and sampling-based methods. Graph-based algorithms, such as Dijkstra's [112,113] and A* [114–116], discretize the environment to guarantee optimal solutions. In contrast, sampling-based algorithms, like Rapidly-exploring Random Tree (RRT) [117], excel in high-dimensional, complex environments by sampling in continuous space to efficiently connect the start and goal points. While RRT generates suboptimal paths, its variant, RRT*, achieves asymptotic optimality. Near-asymptotically optimal algorithms balance efficiency and fast path convergence, sacrificing some optimality for improved performance. Next, we introduce RRT-related algorithms based on these three categories.

Table 4
Summary of mapping methods.

Categories	Method	Map topic (References)	Resolution	Accuracy	Comment
Static environment	Discretization	Raster map [97]	Solid	Low	Accuracy is limited by low resolution
	Vector markup	Vector map [99]	High	High	Lack of elevation information, only applied to planar scenes
	Point cloud projected	Point cloud map [81,100]	Very high	Very high	High accuracy, but high data volume for large scenes
	Map fusion	[101]-[103]	–	–	More computing resources are needed
Dynamic environment	Crowdsourcing data updates	[104]	Low	Low	Convenient, but slow update speed and inconsistent data
	Incremental update	[105]	Medium	Medium	The data is consistent, but struggles with dynamics scenes
	SLAM	[106]	High	High	High accuracy, but should be noted the degraded scenes
High-level information	Semantic modeling	Semantic map [109]	Very high	High	More information, but higher maintenance costs
	Spatial analysis	Topological map [110]	Low	Low	Lack of absolute positional information
	Map fusion	[111]	–	–	High computational complexity

3.1.1. Sub-optimal algorithms

Scholars have optimized the road map to enable RRT to quickly find a safe and feasible path, from aspects such as multi-tree structure, goal-biased, obstacle-biased, region-restriction, and road map. Therefore, the sub-optimal algorithms can be divided into the following categories in this section.

Multi-tree Structure: The bidirectional planner (RRT-Connect) [118]. That is, based on RRT, two trees are generated. In each iteration, one tree expands, and another tree tries to connect the nearest vertex of another tree to the new vertex until the two trees are connected, which makes it more efficient. Experiments have shown that the double-tree approach is much more efficient than single-tree. This method also has some drawbacks, such as only considering the possibility of the existence of the homotopy path, which limits the further development of the path quality. In order to solve the problem that the RRT algorithm selects sampling points in the narrow region of high dimension, Wang et al. [119] proposed a triple tree (Triple RRTs) structure.

Goal-biased Sampling: The goal-oriented sampling method in [120] is proposed to reduce the computation time and quickly generate an initial path, which is different from the RRT in the sampling phase. This sampling method limits the sample point's area and changes the sample point's probability. The sampling phase uses pre-sampling to use the target point as the target point, and the probability of being sampled is proportional to the distance from the point to the target point. The sampling points are basically gathered near the target point.

Obstacle-biased Sampling: Many research methods use the information on obstacles in the state space as a basis for biasing, using the information on the local obstacles to guide the sampling by selectively biasing the sampling to the surface of the obstacle and possibly promising areas containing narrow channels. This can improve the problem of the fast search random tree not being sufficiently sampled in a sparse area, such as a narrow channel. In [121], the Dangerzone RRT (DRRT) algorithm was proposed. The algorithm collects sampling points in the vicinity of the danger zone in the spatial search. In a search space consisting of obstacles and hazardous areas, the sampling process uses the triangular surface obtained from the hazardous area and uses it for the sampling process after the initial path is given, causing the tree to grow along the edges of the hazardous area. However, because it focuses on the characteristics of local information (obstacle), it may incompletely explore the space, which may lead to homotopy problems and local minimum problems.

Region-restriction based Sampling: This approach uses the information classification of the search space to define the region that is most promising as part of the optimal path and then uses this region as the next phase of sampling. Motwani et al. [122] proposed Local Principal Component Analysis RRT (LPCA-RRT), based on dynamic system state space modeling cost function. Local planning is improved by discretizing the deviation of the search space samples. In two steps, first, when the system dynamically simulates the space, the direction of propagation of the state space points sampled on the grid is known. Second, the RRT is applied so that the sampling of the RRT is biased toward the point generated by the first step.

Roadmap based Sampling: To address the problem of the RRT algorithm being difficult to quickly plan an effective path in complex scenes such as narrow spaces and mazes, Chi et al. [123] proposed using generalized Voronoi diagram (GVD) to construct a roadmap for static environments, and using the roadmap to generate a heuristic path to guide RRT sampling. Generally, the GVD algorithm can calculate all points on the map that are equidistant from obstacles in a geometric way, which can well represent the collision-free area of the map. The author reduces the redundancy of map feature node representation and subsequent repeated map search and time-consuming obstacle detection in an exponential manner through three steps: feature extraction, feature matrix, and feature node fusion. This roadmap-based sampling method can quickly plan a feasible heuristic path in complex scenes with ms-level time and each node of the heuristic path as the sub-goal of the RRT algorithm to guide planning.

3.1.2. Near asymptotically optimal algorithms

RRT focuses on finding an executable initial path in a simple and fast way. RRT* puts emphasis on improving the quality of the generated path based on RRT. The near-asymptotically optimal planning algorithms have the following characteristics: 1. They need not the BVP solver [124], which is replaced by the forward propagation model without steering function. Similar to RRT, the category only needs to propagate a single time for each iteration. 2. Providing a sparse data structure for answering path queries, which further improves computational performance. The disadvantage of this type of method is that it only guarantees the completeness of the probability, subscribing to the asymptotic optimality of path quality while sacrificing partial high efficiency and fast convergence.

SPARSE-RRT [125] introduces RRT with Best Nearest Strategy and RRT with Drain. BestNearest selects low-cost point expansion. Drain uses the adjustment method to generate a new node. It is better to judge whether other existing nodes are better. Only keep this one node, and everything else is removed so that only the points with good paths will be retained. Unlike RRT*, SPARSE-RRT only needs to maintain a sparse data structure, avoiding BVP problems and being more efficient.

3.1.3. Asymptotically optimal algorithms

As the number of sampling points increases, RRT gradually converges to the optimal path, though the convergence slows down. To improve this, RRG and RRT* were introduced. RRT* operates similarly to RRT but differs in its parent node selection when adding new nodes and reconnecting surrounding nodes.

There are many variant algorithms for RRT*, which generally divide these algorithms into three categories: the single tree, the hybrid trees, and the bidirectional trees.

Single Tree: A typical single-tree RRT* algorithm is called Informed RRT* [126], which increases the speed of asymptotic optimization by limiting the sampling range. As with RRT*, an initial path is generated, after which it is only sampled from a subset of states defined by acceptable heuristics, possibly to improve the solution, with the initial state and the goal state as the elliptical subset, respectively. The two focal points of the region, with the iteration of the algorithm, the quality of the path is gradually improved so that the short axis of the ellipse is also shortened until the best (progressive optimal) path is found. The algorithm balances the problem between exploration and exploitation with falling into homotopy problems. The disadvantage is that in high-dimensional space, the area of the elliptical area will be large, which will not improve the path quality.

Hybrid Trees: Recently, some scholars have combined RRT* with PRM, discrete search, artificial potential field method, machine learning, neural networks, and so on. One typical method is based on neural networks. Qureshi and Yip [127] cite the concept of adaptive sampling called biased-based sampling, where the sampling area (distribution of sampling points) is restricted to a certain range. Although traditional RRT can also be combined with the method of adaptive sampling, it relies too much on the heuristics of artificial construction. Deep sampling-based (Deep SMP) combines deep neural networks with a sampling-based method to train two neural modules offline: (1). Contractive AutoEncoder (Contractive auto-encoders: Explicit invariance during feature extraction) [128] (2). Deep neural network. The previous one is introduced to encode the obs-space point cloud data into a fixed and robust feature space. Dropout-based [129] stochastic Deep Neural Network (DNN), a key part of the samples generator, uses the obs-space encoder to scale the input and output and then generate random samples for RRT* online. Subsequently, the RRT* is implemented to generate a training path to train DeepSMP, minimizing the mean-squared error (MSE) between the predicted state and the real state.

Bidirectional Trees: Chen et al. [130] combined RRT with improved RRT*, the bidirectional tree structure is novel in that it replaces and simultaneously maintains the structure of the two trees with a Hierarchical structure. Firstly, RRT is used to find different homotopy paths, and then improved RRT* is implemented to optimize the path. Shortcutting is used to reduce redundant branches generated by RRT. Gaussian sampling cloud technology is introduced into the RRT for sampling. Since the Euclidean distance does not reflect the differential constraint, the Clothiod path is replaced by a distance metric. Combine the reconnection of the shortcut with the RRT and set the 'Re-SearchParent' procedure instead to save computational effort. In the re-planning phase, only the optimal subtrees in the original tree are extracted.

To sum up, algorithms for static environments focus on finding the most efficient and optimal paths without the need for frequent re-planning. Graph-based algorithms leverage pre-defined nodes and edges, allowing them to compute globally optimal paths since there is no need for continuous updates. RRT (Rapidly-exploring Random Trees) variants, like multi-tree RRT, SPARSE-RRT, and bidirectional trees RRT*, are adapted to static environments by generating paths that might not be optimal but are computationally efficient. The application of these methods in various environments are shown in Fig. 4. Their adaptations include reducing the random samples and balancing path quality with computation time.

3.2. Dynamic environment

Compared with the static environment, the path planning problem of the dynamic environment is undoubtedly much more difficult. On the way to executing, the robot will inevitably encounter many moving obstacles with unknown directions. It is difficult to accurately determine whether these obstacles will block the advancement of the robot. Thus, we should Emphasize the importance of dynamically designing the path to avoid collisions. In a dynamic environment, the following factors can be changed individually or simultaneously: 1. As the robot moves, new obstacles appear in the field of view; 2. Previously observed obstacles may change position; 3. Target points may change.

3.2.1. Emergence of new obstacles

In general, Execution extended RRT (ERRT) [132] is considered to be the first algorithm to be used in a real-time dynamic environment. If the growth process is blocked, the algorithm will abandon the current tree and reselect another new tree growth. The method simultaneously plans paths and executes them, extending the waypoint cache and adaptive cost penalty search on the basis of RRT in order to improve the efficiency and path quality of re-planning. However, the ERRT is only look-ahead path planning, resulting in too long a distance to the goal state.

Dynamic Rapidly-exploring Random Trees (DRRT) [133], unlike ERRT, when an obstacle or tree growth is blocked, the DRRT only trims the unqualified part of the path and then continues to repair the new node on the original tree. In addition, the DRRT sets the root node at the target point, and the failed node will be less. Multipartite RRTs (MPRRT) [134] combines ERRT with DRRT. The algorithm consists of three key techniques: 1. The distribution of biased sample nodes; 2. Re-use previous planning iterations, similar to ERRT, resulting in the regeneration of the tree. 3. Maintain separate detached forests for a limited time based on RRF improvements, reducing the operating time of unpromising areas.

3.2.2. Previously observed obstacles have changed

The D* algorithm, as introduced in [135], represents an evolution of the A* algorithm specifically designed to manage dynamic environmental changes effectively. This adaptation allows for the prompt adjustment of edge weights to accurately reflect real-time obstacles, thereby facilitating efficient path recalculations. By constructing a temporary navigational map, D* efficiently guides robots through the most accessible routes, making it highly suitable for environments where conditions change unpredictably. A novel application of D* is explored in [136], where a multi-objective incremental search algorithm based on D* is presented. By expanding the search strategy to include multiple objectives, this approach provides a more holistic solution to complex path-finding challenges, ensuring that the paths generated not only avoid obstacles but also optimize other important factors such as distance, time, or energy consumption.

Adiyatov and Varol [137] extended RRT*FN to a dynamic environment and proposed RRT*FNDynamic (RRT*FND) for dynamic obstacles. The methodology is similar to DRRT, and the following aspects are improved: (1) Based on RRT*FN, the maximum number of nodes is fixed, so when the path is unreachable due to encountering dynamic obstacles, the invalid nodes can be quickly deleted; (2) is superior to DRRT in the quality of the path; (3) reconnect the re-plan tree to the best path previously generated. Du and Liu [138] applied the Attractive Potential Field as a construction probability map. This probability map contains the prior knowledge of the current shortest path. Then, this probability map is regarded as a heuristic to guide the sampling point to the target point instead of the obstacle. In the post-processing stage, the cubic b-spline is used to post-process the

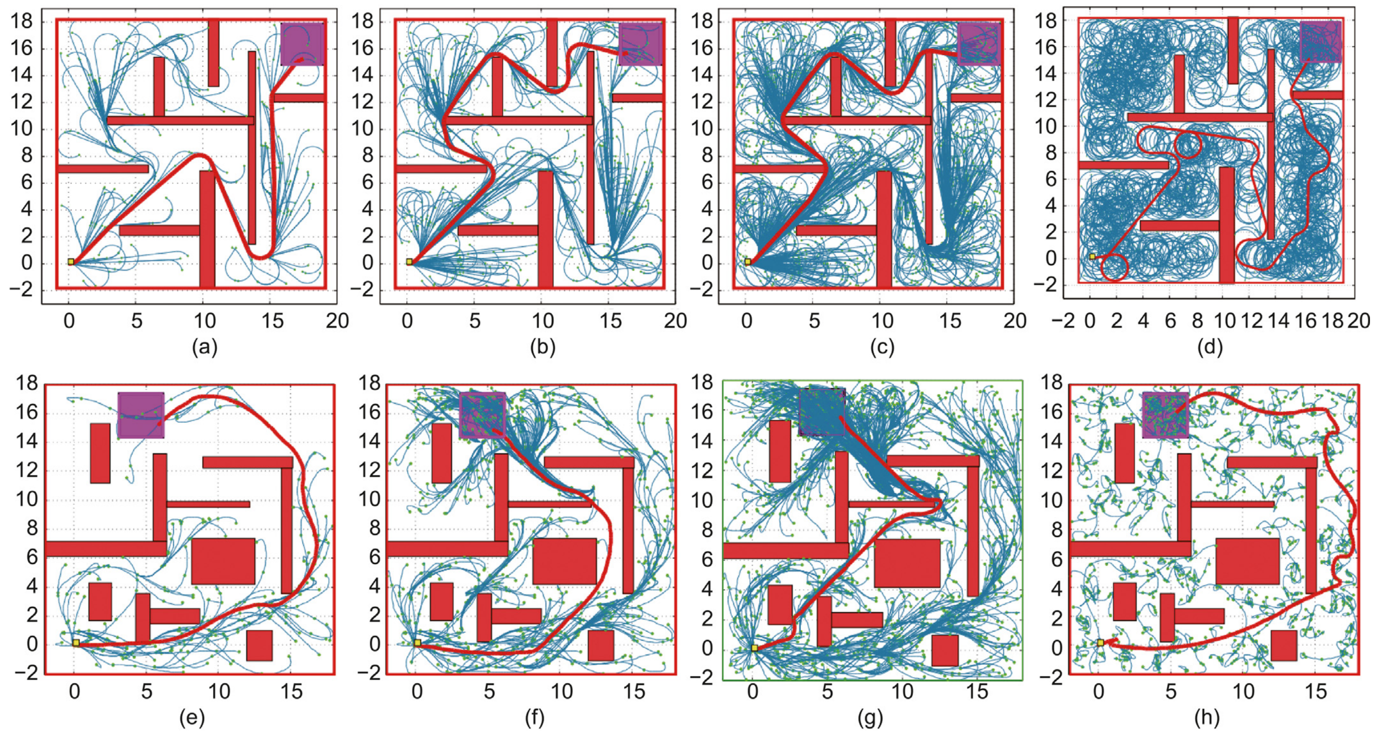

Fig. 4. RRT and RRT* algorithms were run in various environments for a dynamical system with Dubins' vehicle dynamics as well as one with a 2D double integrator dynamics. In Figs. 3(a) and 3(c), the tree maintained by the RRT* algorithm is shown when including around 500 and 6500 vertices, respectively. In Fig. 3(d), the tree maintained by the RRT algorithm is shown when including around 2000 vertices. In Figs. 3(e) and 3(g), the tree maintained by the RRT* algorithm is shown right after 300, and 1500 iterations, respectively. The tree maintained by the RRT algorithm right after 1500 iterations is shown in Fig. 3(h) for comparison [131], ©, 2010 IEEE.

path to increase its smoothness. RRT^X [139] is widely considered to be the first asymptotically optimal re-planning algorithm for sample-based single queries with a guarantee of probability completeness, which aggregates both sample-based and discrete-based searches. This algorithm is mainly designed for unknown dynamic environments without predicting the environment and requiring prior offline calculations. The algorithm sets the root node to the goal point so that the tree does not fail as the environment changes. If the sensor detects an obstacle during the robot's travel when an initial path is planned, it can quickly repair the graph, remodel the shortest path, and quickly avoid it. When obstacle conditions change, a cascading rewire is performed to update the tree's relationship with RRT*. The algorithm guarantees that: (1) each node can maintain expected O(log n) neighbors; (2) the quality of the path is RRT* as the lower limit; (3) the asymptotically optimal.

Assuming that the robot does not have any prior knowledge, Hybrid-RRT* (H-RRT*) [140] is proposed, which can effectively track the previous and current positions of the moving obstacle. When the camera detects moving obstacles, the robot can discard the original target point and set the current position of the moving obstacle to the temporal goal based on the initial global path generated by RRT* so that the robot can move toward this point with the introduction of 'go from behind' strategy.

3.2.3. Dynamic shifts in target points

Artificial Potential Field (APF) methods [141] are favored in path planning due to their low computational demands and straightforward design and are suitable for real-time decision-making scenarios where the target point changes. Such algorithms leverage the concept of potential fields, both attractive and repulsive, for real-time path planning. These methods utilize a potential function within the configuration space (C-space) to differentiate between obstacle-filled and free areas, assigning higher values near obstacles and lower ones at a distance, with a minimum at the goal location. Initially, a potential function is established within the C-space that delineates free from obstacle-laden areas. This function is characterized by high values near obstacles and low values further away, with a minimum set at the goal location. Robots are guided by the gradients of this function, moving from the starting point toward the goal through dynamic updates in their paths. A method introduced in [142] enables robots to follow a predefined path while allowing for stochastic distribution among various paths within distinct topological categories. This approach enhances the adaptability of robotic systems to dynamically changing environments by providing flexible path-planning options. In a novel integration of APF with advanced learning techniques, [143] utilizes APF within an approximate cost function where integral reinforcement learning develops a strategy to minimize time and energy consumption in unknown environments. This method effectively transforms a constrained finite horizon problem into an infinite horizon optimal control problem, optimizing long-term operational efficiency.

Connel and La [144] use an algorithm based on a trigonometric function to determine whether the obstacle interferes with the original path of the robot. If the robot is disturbed, the re-planning temporal goal is chosen as in [140]. The algorithm uses the current position as the parent node, rewiring to generate more nodes and selecting the best sub-path to guide the tree to the temporal goal. When the robot reaches the temporal goal, it continues the global path along the original RRT* plan in a manner similar to [140]. Online RRT* (oRRT*) [145] adds two points based on RRT*: (1) the root node moves with the position of the robot. (2) The density of each grid is pre-defined by the occupancy grid, which provides the foundation for further sampling and subsequent rewiring without adding new nodes. It is divided into three steps: (1) online sampling, adding nodes to the tree within a certain number, and re-optimizing when the robot moves. (2) start-point moving, which adds a node in the new location and rewires neighbor; (3) online pruning. Every time a new node is added, an old node is deleted, and the memory efficiency is maintained.

Based on RRT* and Informed RRT*, Naderi [146] proposed the first real-time RRT-based algorithm while not trimming the tree but maintaining the structure of the entire tree, allowing it to query multiple target points. In the face of dynamic obstacles and dynamic target points that can change their position on the fly, RT-RRT* can be rewired quickly in a good real-time manner to deal with changes in the environment. The real-time capability was achieved by introducing two rewiring methods. One is rewiring starting from the root, which creates a growing circle centered at the agent. The other is using both focused and uniform sampling. When the node of the tree is closest to the goal, uniform samples are achieved randomly in the line between the goal and the node.

In summary, algorithms for dynamic environments need to handle moving obstacles and changing layouts, requiring real-time reactivity and path adjustments. D*, a dynamic version of A*, updates only the affected sections of the path rather than re-planning from scratch, making it well-suited for environments where some parts of the layout change frequently. Dynamic RRT (DRRT) and RRT*FND adjust paths based on new information, such as newly detected obstacles or altered goals. DRRT, for example, recalculates paths by pruning and regrowing parts of the tree, which adapts the algorithm for dynamic updates. Artificial Potential Fields (APF) are particularly reactive to nearby obstacles, enabling quick adjustments, though they can struggle with local minima. This high sensitivity makes APF responsive to dynamic changes but limits its suitability in highly complex environments.

3.3. Human–robot coexisting environment

In recent years, with the development of robot technology, various security robots and service robots have gradually entered our lives. This trend has made human life more convenient. Conversely, it has also made the working environment of mobile robots more complicated. The difficulties in dynamic planning under the human–machine coexistence environment are as follows: monitoring and tracking mobile obstacles; prediction of future poses; online planning and navigation [147]. The principle problem is how to find a feasible and optimal trajectory in a complex, dynamic human–machine coexistence environment. Conditions to be met: (1) efficient; (2) low storage requirement. The vast majority of sampling-based planning algorithms are all static, whereas only a small fraction can be applied to dynamic environments quickly and efficiently [148].

3.3.1. Planning with motion prediction of the obstacles

Some planning methods predict the dynamic obstacle or the next position of the pedestrian with the guidance of machine learning, which can be classified as a motion predictor. [149] uses a Model Predictive Control (MPC) method, making it possible to execute the planned trajectory and plan simultaneously. [150] proposes an occlusion-aware motion planning control (OA-MPC) framework. It proposes a safe navigation method for robots operating in dynamic and uncertain environments where occlusions could hide moving obstacles. The solution addresses the issue of predicting the trajectories of hidden dynamic agents (such as pedestrians) using forward reachability analysis, which calculates potential reachable zones for these agents based on worst-case assumptions. Once the MPC is chosen, only a limited segment of the planned trajectories is executed; at the same time, a new trajectory is planned online. When the Execution is over, a new planned trajectory is ready for Execution. The risk-RRT combines the probabilistic collision risk function linking planning and navigation methods with the perception and the prediction of the moving pedestrians [147,151,152]. Assuming that the typical motion pattern in the observed environment has been learned, linear and continuous models are produced. [153] uses the Gaussian process [154] characterizes the typical trajectory, and cooperates with other probabilistic frameworks to complete the prediction of the motion model of dynamic obstacles using Gaussian Processes. The uncertainties in the static environment are characterized by the occupancy grid, and the uncertainties in the dynamic environment are characterized by a mixed Gauss of each timestamp. The likelihood of the future trajectory of the obstacle and the probability of the occupancy is used to calculate the risk of the collision, and the calculated risk of each different timestep is recorded into the Occupancy Grid Map, which guides the node with the least risk. Combine time with re-planning online to update the path in real-time. However, risk-RRT does not care about path quality, and the generated path is suboptimal. In addition, simply considering people as mobile obstacles with a typical motion model while ignoring the comfort of pedestrians and social adaptive conventions cannot meet the social requirements for robots in the human–computer coexistence environment.

Compared to [152,155] not only deals with the notion of risk of collision but also takes account of social conventions in the navigation process based on risk-RRT, which represents the notion of risk of disturbance, equipping the robot with the features to respect human interacting and proximity constraints such as personal space and o-space. Robots can be competent in carrying out 'joining a group' tasks, which are based on socially adapted behavior. Risk-RRT* [148] combines RRT* with comfort and collision risk (CCR) map [156]. The human comfort model is used together with collision risk to calculate the occupancy grid map. Combined with RRT*, the efficiency of the algorithm and the comfort of the person are effectively improved. In spite of the quality of the path being solved, it takes a long time to obtain this effect. Risk-based Dual-Tree Rapidly exploring Random Tree (Risk-DTRRT) [157] is proposed to improve the quality of the path and find the optimal homotopy path in the heuristic path. The paper points out the problems of reconnecting on the original tree: firstly, recalculating risk and input according to the timestamp will generate an inverse dynamic problem, and the calculation amount will increase. Secondly, it will waste the effective information of the original tree. The Dual-Tree planning framework is proposed, including the initial heuristic tree and the rewired tree. First, a time-based initial heuristic path is planned using Risk-based RRT. Instead of directly reconnecting the initial path, the rewired tree is established, and the line-of-sight control algorithm is introduced to find the possibility of minimum-cost rewiring.

3.3.2. Learning from demonstration

Learning from Demonstration (LfD) is a promising technology that can release robot prototypes from research laboratories. Based on LfD, robots can successfully work in the real world, even realizing human–robot collaboration (HRC) without any prior knowledge of robotics technology in order to teach robots new tasks. Researchers should not only consider robot learning algorithms or techniques but also take into account many human-centric issues, such as the human partner's feelings and intentions during collaboration phases [158]. [159] proposes CAM-RL, a crowd-aware memory-based reinforcement learning method that uses gated recurrent units (GRU) to model human dependencies and enhance robot navigation in crowded environments. By incorporating multi-layer perceptron (MLP) and attention mechanisms, CAM-RL improves human–robot interaction. In [160], a socially aware mapless navigation algorithm combines safe reinforcement learning with societal traffic norms to improve cooperative avoidance and navigation success in complex environments. This novel approach highlights the importance of incorporating social behaviors and norms into navigation algorithms to enhance interaction dynamics among robots.

The field of robot programming through demonstration or simulation involves the transfer of effective behavior from human demonstration to observing the robot. Learning from demonstration avoids hard programming in complex environments. There are several methods for accurately introducing human navigation features into the navigation path of the robot, which the first is to adjust the behavior of the robot by adding the constraints of human society to the robot's path planning program through the hard-coding based method [161]. Compared to abstract mathematical formulas that are hardly defined in the form of coding, there is no doubt that demonstration can more vividly describe social navigation. [162] combines Fully Convolutional Neural Networks (FCNs) and RRT*, applying fully convolutional networks to learn from demonstrations representing human-aware navigation. [163] gives a series of expert examples for extracting the reward function and maximizing or minimizing the reward function in the process of iteratively. Purely using IRL on the basis of the current cost function at each iteration, planning forward, which is not suitable in a complex environment. IRL, as an auxiliary tool of RRT* can guide the sampling process of RRT*, making the planning more efficient. [164,165] learn social navigation behaviors from the data sets of expert demonstrations obtained. The IRL of Maximum entropy-based [166] is used to continuously update the weights of each feature for social navigation to guide the next RRT* growth. [167] proposes Raiding Rapidly Exploring Learning Trees (RLT*), applying IRL to learn the cost function used by RRT* from the expert example. [167] also proposes Approximate Maximum Margin Planning (AMMP), which is extended by the maximum margin (MMP). The algorithm uses the caching scheme to improve its performance and reduce computational costs. Why is AMMP introduced in RRT*: noisy gradients can be produced when a separate set of points is sampled at every iteration, which can affect convergence. Only when the cost function is updated is a rewire needed. IRL does not require modeling of system dynamics.

Briefly, in human–robot coexisting environments, algorithms need to prioritize safety, predict human behavior, and create comfortable, collision-free paths. MPC (Model Predictive Control) incorporates predictive models to anticipate human movements, adapting its path to ensure safe distances are maintained. By constantly updating predictions, MPC can generate smooth paths that adapt to human presence in real-time. Risk-aware algorithms, like Risk-RRT* and Risk-DTRRT, integrate risk assessment into path planning, prioritizing routes that minimize the potential for risky interactions. This adaptation involves assigning higher costs to areas with a high likelihood of human presence, allowing the robot to proactively avoid these areas. RLT* combines reinforcement learning with RRT* to adapt its planning based on prior human–robot interactions, gradually optimizing paths for smooth interactions while remaining responsive to changing human behaviors.

3.4. Multi-robot environment

While human–robot interaction focuses on the dynamics between robots and individual humans, multi-robot collaboration involves coordinating multiple autonomous agents simultaneously, introducing distinct technical challenges. Multi-robot systems require advanced algorithms for seamless inter-robot communication, task allocation, and conflict resolution to ensure cohesive and efficient operations. Unlike human–robot interactions, which primarily address adapting to human behavior, multi-robot collaboration demands synchronized actions and consistent information sharing among all robots to achieve collective objectives.

Additionally, scalability and environmental adaptability are critical hurdles unique to multi-robot systems. As the number of robots increases, the computational and logistical demands escalate, necessitating scalable control solutions that can handle exponential data processing and communication overhead. Furthermore, these systems must navigate diverse and dynamically changing environments while maintaining tight coordination, requiring highly adaptive and resilient navigation algorithms. Robust communication protocols and decentralized control strategies are essential to managing real-time data exchange and autonomous decision-making, distinguishing multi-robot collaboration from the typically simpler communication needs in human–robot interaction.

3.4.1. Bio-inspired methods

Bio-inspired algorithms for multi-robot navigation harness the principles of natural systems to navigate complex environments efficiently and adaptively. Unlike traditional mathematical modeling techniques that focus on explicit environmental modeling, these methods emulate biological processes, fostering decentralized decision-making akin to individual organisms within a collective. This approach not only scales well with environmental changes but also avoids common computational pitfalls such as entrapment in local minima and the need to solve complex objective functions directly.

As an important category of bio-inspired approach, Particle Swarm Optimization (PSO) has emerged as a robust tool for addressing NP-hard challenges in multi-robot path planning, favored for its straightforward implementation and quick convergence properties. Researchers have developed an interval multi-objective PSO, documented in [168], which innovatively enhances the updating mechanism for global best positions and adjusts crowding distances. Furthermore, an innovative combination of PSO with differentially perturbed velocities, explored in [169], effectively minimizes path lengths and arrival times. This approach is complemented by integrating a time-stamp segmentation model designed to handle coordination costs more efficiently, thus optimizing the overall time efficiency of the robotic swarm. In another contribution, PSO is employed alongside coevolutionary strategies and evolutionary game theory, as documented in [170]. Additionally, a novel hybrid approach that combines democratic robotics-oriented PSO with improved Q-learning is introduced in [171]. This hybridization aims to facilitate fast, real-time path planning. Collectively, these studies enhance the application spectrum of PSO in complex multi-robot path planning scenarios. They showcase a broad array of enhancements, from theoretical refinements that improve algorithmic efficiency and adaptability to practical implementations that address real-world operational challenges.

On the other hand, genetic algorithms are highly effective for multi-robot navigation, applying principles of natural selection to evolve optimal paths in complex environments. For instance, in [172], genetic algorithms are employed to solve the Multiple Traveling Sales Person problem, utilizing Euclidean distances and Dubins curves to ensure path continuity. Moreover, in [116], genetic algorithms are deployed alongside area sensors, which significantly improve the management of persistent cooperative coverage tasks, showcasing their utility in environments

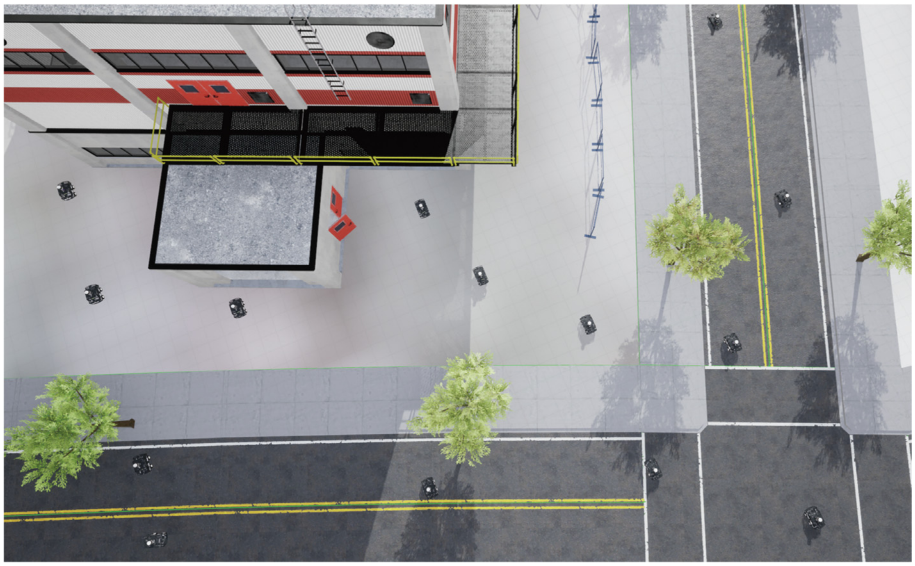

Fig. 5. Multi-robot navigation in a complex environment.

requiring continuous operational efficiency. In [173], genetic algorithms are specifically tailored for path optimization in static environments, cleverly differentiating between offline and online scheduling strategies to effectively mitigate conflicts among robots. Furthermore, in [174], genetic algorithms are innovatively adapted for real-time, cooperative path planning. The authors redefine genetic operators to enhance operational responsiveness, enabling more agile and effective responses in dynamic settings. These varied applications underscore the effectiveness of genetic algorithms in navigating the complexities of multi-robot path planning. By leveraging their robust genetic operators and evolutionary principles, these algorithms not only tackle high-dimensional search spaces but also adapt to the evolving dynamics of multi-robot systems.

3.4.2. Learning-based methods

Learning-based navigation strategies employ sophisticated machine learning techniques, notably RL, to empower multi-robot systems with advanced path planning. These strategies harness extensive environmental interaction data to train robust algorithms, enabling robotic systems to autonomously develop and refine effective navigational and cooperative behaviors. The principal strength of these approaches lies in their adaptability; robots dynamically enhance their decision-making by integrating real-time data feedback and ongoing model optimization, effectively adapting to evolving environmental conditions. This adaptability, combined with scalability that accommodates varying swarm sizes, robustness against sensor inaccuracies, and heightened autonomy that minimizes human intervention, significantly bolsters the efficiency and effectiveness of multi-robot collaboration.

Recent advancements in learning-based multi-robot path planning have showcased the profound impact of neural network models and sophisticated decision-making frameworks on enhancing autonomous navigation capabilities across various robotic platforms. A prime example of this integration is the Cooperative Autonomous Distributed Robotic Laboratory (CADRL) algorithm, introduced in [175], which employs deep reinforcement learning to develop cooperative navigation policies tailored for navigating complex environments. Additionally, the study presented in [176] explores a novel reinforcement learning policy specifically crafted for distributed agent-level obstacle avoidance. This policy achieves faster convergence rates compared to conventional methods, highlighting the effectiveness of specialized learning strategies in improving the autonomy and responsiveness of individual robots within a multi-robot system. Building upon these developments, the research in [177] utilizes the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm to address both goal assignment and path planning simultaneously. This method has shown significant improvements in real-time performance for unmanned aerial vehicles (UAVs) operating in dynamic environments.

The rapid evolution in learning-based multi-robot navigation is further exemplified by the work in [178], which introduces a hierarchical framework adept at integrating sensor and agent data to significantly enhance collaborative navigation in environments where robot communication is constrained. This methodology not only enhances data utilization but also ensures more coherent and unified decision-making processes among the robots. Additionally, the study presented in [160] introduces a socially aware mapless navigation algorithm that ingeniously blends safe reinforcement learning with societal traffic norms. By embedding social behaviors and norms into the navigation algorithms, this strategy enriches the interaction dynamics among robots, allowing for more natural and intuitive robotic behaviors in environments typically navigated by humans, thereby facilitating smoother integration and interaction. Expanding upon these innovations, [179] explores a learning-based method specifically designed to tackle the challenges of multi-agent navigation to multiple destinations amidst a mix of static and dynamic obstacles. Collectively, the above studies mark significant strides in advancing the capabilities of multi-robot systems. They address the varied of difficult challenges encountered navigation in complex environments (as shown in Fig. 5) through continuous algorithmic innovation and adaptation. These research efforts underscore a notable shift towards developing more intelligent, autonomous, and collaborative robotic systems.

3.5. Discussion

As shown in Table 5, we selected four representative algorithms from four environments for comparative analysis. The parameters involved in the table are defined as follows. V is the number of nodes, and E is the number of edges of a graph. n is the

Table 5
Summary of planning methods.

Categories	Method	Path quality	Convergence speed	Scalability	Computational complexity
Static environment	Graph-based [112], [114]	Optimal	Medium	Medium	$\mathcal{O}((V+E)\cdot\log V)$
	Multi-tree RRT [118], [119]	Sub-optimal	High	High	$\mathcal{O}(n\log n)$
	SPARSE-RRT [125]	Near Asymptotically optimal	Medium	High	$\mathcal{O}(n\log n)$
	Bidirectional trees RRT* [130]	Asymptotically optimal	Low	High	$\mathcal{O}(n^2)$
Dynamic environment	D* [135]	Optimal	High	Medium	$\mathcal{O}((V+E)\cdot\log V)$
	DRRT [133]	Sub-optimal	High	High	$\mathcal{O}(n\log n)$
	RRT*FND [137]	Asymptotically optimal	Low	High	$\mathcal{O}(n\log n+k)$
	APF [141]	Sub-optimal or no solution	High	Low	$\mathcal{O}(m)$
Human-Robot coexisting environment	MPC [149]	Optimal	Medium	Low	$\mathcal{O}(N^3)$
	Risk-RRT* [148]	Asymptotically optimal	High	High	$\mathcal{O}(n^2)$
	Risk-DTRRT [157]	Asymptotically optimal	High	High	$\mathcal{O}(n^2)$
	RLT* [167]	Asymptotically optimal	Low	High	$\mathcal{O}(n\log n)$
Multi-Robot environment	PSO [168], [171]	Sub-optimal	Low	Medium	$\mathcal{O}(A*N)$
	Genetic Algorithms [172], [116]	Sub-optimal	Low	Medium	$\mathcal{O}(A*N)$
	RL [175]	Sub-optimal	Medium	Low	$\mathcal{O}(A*S)$
	Multi-agent RL [177], [178]	Sub-optimal	Medium	Medium	$\mathcal{O}(A^2*S)$

number of sampling nodes of a tree. m is the number of obstacles in the environment. k is the number of dynamic environment updates. N is the number of time steps in the prediction. In static environments, algorithms can assume that obstacles and the environment layout remain constant, which allows for precomputed or gradually optimized paths. Graph-based algorithms provide optimal paths by searching through pre-defined nodes and edges, but their computational complexity is high. Multi-tree RRT can explore large spaces faster by creating multiple trees, though the path quality is typically sub-optimal. SPARSE-RRT introduces sparsity to reduce node density and improve search speed, producing paths close to optimal. Bidirectional trees RRT* works with two expanding trees from the start and goal, aiming for a high-quality, asymptotically optimal path but at a slower convergence rate. In dynamic environments, obstacles may move or appear unpredictably, requiring algorithms that can adapt paths in real-time. D* algorithm efficiently re-plans paths by updating only the affected parts, ensuring an optimal path but at the cost of higher computation. DRRT extends RRT by allowing dynamic replanning, providing sub-optimal but feasible paths quickly. RRT*FND builds on RRT with frequent updates, which provides asymptotically optimal paths in a dynamic environment but with significant computational cost. APF (Artificial Potential Fields) is faster but often struggles with local minima and may fail in complex environments. In environments where robots coexist with humans, path planning must account for human behaviors and interactions. MPC (Model Predictive Control) produces optimal paths by continuously predicting future positions, but it requires high computation and may not be suitable for real-time applications. Risk-RRT* and Risk-DTRRT incorporate risk assessments, making them suitable for environments where safety is a priority, but this increases computational complexity. RLT* combines reinforcement learning with RRT*, gradually optimizing paths in real-time, though the convergence speed remains low.

Table 5 also illustrates the characteristics of various multi-robot navigation methods. A is the number of robots, while S is the size of observation space. Both PSO and genetic algorithms exhibit low convergence speeds and medium scalability, suggesting that while they can manage a moderate number of robots, their efficiency diminishes as the system grows. In contrast, RL methods achieve medium convergence speeds but suffer from low scalability, limiting their applicability in larger multi-robot systems. Multi-agent RL strikes a balance with medium scalability and medium convergence speeds; however, it incurs higher computational complexity due to the increased interactions between agents. Overall, the analysis underscores the trade-offs between convergence speed, scalability, and computational demands inherent in each method.

4. Challenge and direction

4.1. Perception

One of the most significant issues is occlusion observation [180,181]. Achieving robust perception with partial observation is still an open problem. Learning-based methods benefit significantly from large-scale data and deep neural networks. However, when deployed in robotic operation environments, these methods often suffer from domain shift, where the learned networks fail to generalize to conditions that differ from the training data. To address this challenge, online continual learning (OCL) methods [182,183] can be employed to fine-tune the network with newly acquired data in real-time. However, OCL still faces issues like slow convergence and unstable fine-tuning when deployed in real robotic scenarios.

As for localization and mapping, they are affected by several factors, including sensor noise, external interference, and environmental conditions such as weather. In low-texture, dynamic, or complex environments, localization accuracy tends to decline, posing potential safety risks. To address these challenges, multi-sensor fusion is key, combining data from various sensors while managing issues like data inconsistency and time delays. Additionally, achieving high-precision localization requires processing large amounts of perceptual information in real-time, necessitating a balance between complexity and performance. In dynamic environments, real-time map updates are critical, requiring efficient data processing to ensure that maps reflect changes accurately. Large-scale mapping adds further complexity due to the vast amounts of data involved, necessitating efficient memory management and processing techniques. Methods must also ensure consistency and comparability of data across varying regions and applications to support reliable decision-making in diverse scenarios.

4.2. Planning and collaboration

Currently, the challenges of path planning algorithms mainly focus on real-time performance and adaptability in complex dynamic environments. With the increasing diversity of application scenarios, ensuring the safety and efficiency of paths in crowded or uncertain environments has become a key challenge. Additionally, computational efficiency and energy consumption are gaining more attention in industrial and commercial applications. To address these challenges, research is shifting towards the incorporation of intelligent algorithms such as deep learning and reinforcement learning to enhance robustness and adaptability in path planning. Furthermore, human–robot interaction comfort and multi-robot collaboration are also key areas of focus. In the future, path planning algorithms combined with predictive models and learning methods will be better equipped to handle uncertainty and dynamic changes in complex environments, achieving more intelligent autonomous navigation.

Besides, in the realm of multi-robot path planning, several core challenges hinder the development of efficient, robust systems. First, scalability remains a pivotal issue as increasing the number of robots in a system introduces significant computational and coordination complexities. Additionally, the heterogeneity of robot capabilities presents another layer of complexity. Moreover, local optima poses a substantial challenge in optimization-based navigation strategies. To overcome these challenges, future research directions could focus on developing adaptive, learning-based algorithms that enhance scalability and flexibility in system design. Leveraging artificial intelligence and machine learning could enable more dynamic decision-making processes, allowing robots to autonomously adjust their strategies based on real-time data. Furthermore, the exploration of decentralized decision-making models, where robots operate semi-independently while adhering to a set of global objectives, may provide a balance between autonomy and coordination, effectively avoiding local optima and improving overall system performance.

5. Conclusion

Our review has highlighted the importance of perception in replicating human-like comprehension and interpretation of the environment. Despite significant advancements in recent years, numerous challenges persist. Condition sensitivity in appearance features, occlusion problems, and domain drift in learning-based methods represent ongoing obstacles. Also, the degenerate scenes, the complexities of large-scale data processing, and the noise sensitivity are still open problems.

In terms of path planning for single-robot systems, we review the development of graph-based and sampling-based path planning algorithms. With the expansion of application scenarios to industrial and commercial domains, researchers also began considering the impact of dynamic obstacles. However, simply treating pedestrians as obstacles may disrupt their normal activities, causing congestion. Therefore, achieving comfortable human–robot interaction has become a critical issue for path-planning algorithms. The use of predictive and learning-based approaches is now the mainstream direction in both practical applications and research for robotics.

For multi-robot systems, this review has systematically explored a broad spectrum of multi-robot navigation strategies. It provided an in-depth analysis of each method, highlighting their unique features and key contributions. Despite notable advancements, persistent challenges such as scalability, heterogeneity of capabilities among robots, and susceptibility to local optima continue to constrain the efficacy of these systems. Promising research directions include the integration of learning-based with bio-inspired models to enhance adaptability and decision-making capabilities. Moreover, advancements in communication technologies and the development of decentralized algorithms promise to address many of the current challenges.

The manuscript concludes with a discussion of future research directions. We believe that this review can promote the development of autonomous robots and their applications.

CRediT authorship contribution statement

Weinan Chen: Writing – original draft, Investigation. **Wenzheng Chi:** Writing – original draft, Investigation. **Sehua Ji:** Writing – original draft, Investigation. **Hanjing Ye:** Writing – original draft, Investigation. **Jie Liu:** Writing – original draft, Investigation. **Yunjie Jia:** Writing – original draft, Investigation. **Jiajie Yu:** Writing – original draft, Investigation. **Jiyu Cheng:** Writing – original draft, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The work in this paper is supported by the National Natural Science Foundation of China (62103179, 62273246, and U23A20339).

References

[1] J. Ni, Y. Chen, G. Tang, J. Shi, W. Cao, P. Shi, Deep learning-based scene understanding for autonomous robots: a survey, Intell. Robot. (2023).
[2] M. Kim, M. Zhou, S. Lee, H. Lee, Development of an autonomous mobile robot in the outdoor environments with a comparative survey of LiDAR SLAM, in: 2022 22nd International Conference on Control, Automation and Systems, ICCAS, 2022, pp. 1990–1995.
[3] R.K. Raj, A. Kos, A comprehensive study of mobile robot: History, developments, applications, and future research perspectives, Appl. Sci. (2022).
[4] L. Antonyshyn, J. Silveira, S.N. Givigi, J.A. Marshall, Multiple mobile robot task and motion planning: A survey, ACM Comput. Surv. 55 (2022) 1–35.
[5] S. Wang, Y. Wang, D. Li, Q. Zhao, Distributed relative localization algorithms for multi-robot networks: A survey, Sensors (Basel, Switzerland) 23 (2023).
[6] B. Wu, C.S. Suh, State-of-the-art in robot learning for multi-robot collaboration: A comprehensive survey, 2024, ArXiv, arXiv:2408.11822.
[7] J. Orr, A. Dutta, Multi-agent deep reinforcement learning for multi-robot applications: A survey, Sensors (Basel, Switzerland) 23 (2023).
[8] E. Rosten, T. Drummond, Machine learning for high-speed corner detection, in: Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9, Springer, 2006, pp. 430–443.
[9] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, P. Fua, BRIEF: Computing a local binary descriptor very fast, IEEE Trans. Pattern Anal. Mach. Intell. 34 (7) (2011) 1281–1298.
[10] C. Mei, G. Sibley, M. Cummins, P. Newman, I. Reid, A constant-time efficient stereo slam system, in: Proceedings of the British Machine Vision Conference, vol. 1, (no. 2009) BMVA Press, 2009.
[11] D.G. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, Ieee, 1999, pp. 1150–1157.
[12] D. DeTone, T. Malisiewicz, A. Rabinovich, Superpoint: Self-supervised interest point detection and description, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 224–236.
[13] M. Tyszkiewicz, P. Fua, E. Trulls, DISK: Learning Local Features with Policy Gradient, vol. 33, 2020, pp. 14254–14265.
[14] P.-E. Sarlin, D. DeTone, T. Malisiewicz, A. Rabinovich, Superglue: Learning feature matching with graph neural networks, 2020, pp. 4938–4947.
[15] J. Sun, Z. Shen, Y. Wang, H. Bao, X. Zhou, LoFTR: Detector-free local feature matching with transformers, 2021, pp. 8922–8931.
[16] J. Edstedt, I. Athanasiadis, M.a. Wadenbäck, M. Felsberg, DKM: Dense kernelized feature matching for geometry estimation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2023.
[17] Y. Zhang, X. Zhao, MESA: Matching everything by segmenting anything, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 20217–20226.
[18] H. Jégou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3304–3311.
[19] R. Arandjelovic, A. Zisserman, All about VLAD, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 1578–1585.
[20] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, C. Schmid, Aggregating local image descriptors into compact codes, IEEE Trans. Pattern Anal. Mach. Intell. 34 (9) (2011) 1704–1716.

[21] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, J. Sivic, NetVLAD: CNN architecture for weakly supervised place recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5297–5307.
[22] W. Chen, H. Ye, L. Zhu, C. Tang, C. Fu, Y. Chen, H. Zhang, Keyframe selection with information occupancy grid model for long-term data association, in: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2022, pp. 2786–2793.
[23] H. Ye, W. Chen, J. Yu, L. He, Y. Guan, H. Zhang, Condition-invariant and compact visual place description by convolutional autoencoder, Robotica 41 (6) (2023) 1718–1732.
[24] Y. Ge, H. Wang, F. Zhu, R. Zhao, H. Li, Self-supervising fine-grained region similarities for large-scale image localization, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16, Springer, 2020, pp. 369–386.
[25] F. Lu, X. Lan, L. Zhang, D. Jiang, Y. Wang, C. Yuan, CricaVPR: Cross-image correlation-aware representation learning for visual place recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 16772–16782.
[26] C. Godard, O. Mac Aodha, G.J. Brostow, Unsupervised monocular depth estimation with left-right consistency, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 270–279.
[27] C. Godard, O. Mac Aodha, M. Firman, G.J. Brostow, Digging into self-supervised monocular depth estimation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3828–3838.
[28] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, V. Koltun, Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer, IEEE Trans. Pattern Anal. Mach. Intell. 44 (3) (2020) 1623–1637.
[29] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, H. Zhao, Depth anything: Unleashing the power of large-scale unlabeled data, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 10371–10381.
[30] L. Piccinelli, Y.-H. Yang, C. Sakaridis, M. Segu, S. Li, L. Van Gool, F. Yu, UniDepth: Universal monocular metric depth estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 10106–10116.
[31] J.R. Uijlings, K.E. Van De Sande, T. Gevers, A.W. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. 104 (2013) 154–171.
[32] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
[33] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, vol. 28, 2015.
[34] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.
[35] J. Redmon, A. Farhadi, YOLO9000: better, faster, stronger, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7263–7271.
[36] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, 2018, arXiv preprint arXiv:1804.02767.
[37] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: European Conference on Computer Vision, Springer, 2020, pp. 213–229.
[38] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai, Deformable DETR: Deformable transformers for end-to-end object detection, in: International Conference on Learning Representations, 2021.
[39] L.H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, et al., Grounded language-image pre-training, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10965–10975.
[40] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, et al., Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2023, arXiv preprint arXiv:2303.05499.
[41] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, Y. Shan, Yolo-world: Real-time open-vocabulary object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 16901–16911.
[42] A. Bewley, Z. Ge, L. Ott, F. Ramos, B. Upcroft, Simple online and realtime tracking, in: 2016 IEEE International Conference on Image Processing, ICIP, IEEE, 2016, pp. 3464–3468.
[43] N. Wojke, A. Bewley, D. Paulus, Simple online and realtime tracking with a deep association metric, in: 2017 IEEE International Conference on Image Processing, ICIP, IEEE, 2017, pp. 3645–3649.
[44] Y. Zhang, C. Wang, X. Wang, W. Zeng, W. Liu, Fairmot: On the fairness of detection and re-identification in multiple object tracking, Int. J. Comput. Vis. 129 (2021) 3069–3087.
[45] J. Pang, L. Qiu, X. Li, H. Chen, Q. Li, T. Darrell, F. Yu, Quasi-dense similarity learning for multiple object tracking, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 164–173.
[46] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, X. Wang, Bytetrack: Multi-object tracking by associating every detection box, in: European Conference on Computer Vision, Springer, 2022, pp. 1–21.
[47] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, Springer, 2015, pp. 234–241.
[48] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 801–818.
[49] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P.H. Torr, L. Zhang, Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers, in: CVPR, 2021.
[50] R. Strudel, R. Garcia, I. Laptev, C. Schmid, Segmenter: Transformer for semantic segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 7262–7272.
[51] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A.C. Berg, W.-Y. Lo, et al., Segment anything, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 4015–4026.
[52] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K.V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, C. Feichtenhofer, SAM 2: Segment anything in images and videos, 2024, arXiv preprint.
[53] C. Campos, R. Elvira, J.J.G. Rodríguez, J.M. M. Montiel, J. D. Tardós, ORB-SLAM3: An accurate open-source library for visual, visual–Inertial, and multimap SLAM, IEEE Trans. Robot. 37 (6) (2021) 1874–1890.
[54] S. Xu, S. Chen, R. Xu, C. Wang, P. Lu, L. Guo, Local feature matching using deep learning: A survey, Inf. Fusion 107 (2024) 102344.
[55] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft COCO: Common objects in context, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, Springer International Publishing, Cham, 2014, pp. 740–755.
[56] Z. Li, N. Snavely, Megadepth: Learning single-view depth prediction from internet photos, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2041–2050.
[57] A. Dai, A.X. Chang, M. Savva, M. Halber, T. Funkhouser, M. Nießner, Scannet: Richly-annotated 3d reconstructions of indoor scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5828–5839.
[58] S. Lowry, N. Sünderhauf, P. Newman, J.J. Leonard, D. Cox, P. Corke, M.J. Milford, Visual place recognition: A survey, ieee Trans. Robot. 32 (1) (2015) 1–19.
[59] P. Yin, J. Jiao, S. Zhao, L. Xu, G. Huang, H. Choset, S. Scherer, J. Han, General place recognition survey: Towards real-world autonomy, 2024, arXiv preprint arXiv:2405.04812.
[60] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixe, B. Leibe, HOTA: A higher order metric for evaluating multi-object tracking, Int. J. Comput. Vis. (IJCV) (2020).
[61] M. Oquab, T. Darcet, T. Moutakanni, H.V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, P. Bojanowski, DINOv2: Learning robust visual features without supervision, 2023, arXiv:2304.07193.
[62] X. Lin, J. Ruan, Y. Yang, L. He, Y. Guan, H. Zhang, Robust data association against detection deficiency for semantic SLAM, IEEE Trans. Autom. Sci. Eng. 21 (1) (2023) 868–880.
[63] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International Conference on Machine Learning, PMLR, 2021, pp. 8748–8763.
[64] M. Ibrahim, O. Moselhi, IMU-based indoor localization for construction applications, in: ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction, vol. 32, 2015, p. 1.
[65] Y. Wang, H. Cheng, M.Q.-H. Meng, Spatiotemporal co-attention hybrid neural network for pedestrian localization based on 6D IMU, IEEE Trans. Autom. Sci. Eng. 20 (2023) 636–648.
[66] M. Brossard, A. Barrau, S. Bonnabel, AI-IMU dead-reckoning, IEEE Trans. Intell. Veh. 5 (2019) 585–595.
[67] L.V. Nguyen, H.M. La, A human foot motion localization algorithm using IMU, in: 2016 American Control Conference, ACC, 2016, pp. 4379–4384.
[68] T.-N. Do, R. Liu, C. Yuen, U.-X. Tan, Design of an infrastructureless indoor localization device using an IMU sensor, in: 2015 IEEE International Conference on Robotics and Biomimetics, ROBIO, 2015, pp. 2115–2120.

[69] A. Mandow, J.L. Martínez, J. Morales, J.-L. Blanco, A.J. García-Cerezo, J. González, Experimental kinematics for wheeled skid-steer mobile robots, in: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2007, pp. 1222–1227.
[70] Y. Wu, T. Wang, J. Liang, J. Chen, Q. Zhao, X. Yang, C. Han, Experimental kinematics modeling estimation for wheeled skid-steering mobile robots, in: 2013 IEEE International Conference on Robotics and Biomimetics, ROBIO, 2013, pp. 268–273.
[71] J.L. Martínez, A. Mandow, J. Morales, S. Pedraza, A.J. García-Cerezo, Approximating kinematics for tracked mobile robots, Int. J. Robot. Res. 24 (2005) 867–878.
[72] M. Kotaru, K. Joshi, D. Bharadia, S. Katti, Spotfi: Decimeter level localization using wifi, in: Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, 2015, pp. 269–282.
[73] J. Biswas, M.M. Veloso, WiFi localization and navigation for autonomous indoor mobile robots, in: 2010 IEEE International Conference on Robotics and Automation, ICRA, 2010, pp. 4379–4384.
[74] P. Kriz, F. Maly, T. Kozel, Improving indoor localization using bluetooth low energy beacons, Mob. Inform. Syst. 2016 (1) (2016) 2083094.
[75] M. Ridolfi, A. Kaya, R. Berkvens, M. Weyn, W. Joseph, E.D. Poorter, Self-calibration and collaborative localization for UWB positioning systems: A survey and future research directions, ACM Comput. Surv. 54 (4) (2021) 1–27.
[76] K. Yu, K. Wen, Y. Li, S. Zhang, K. Zhang, A novel NLOS mitigation algorithm for UWB localization in harsh indoor environments, IEEE Trans. Veh. Technol. 68 (1) (2018) 686–699.
[77] N.M. Drawil, H.M. Amar, O.A. Basir, GPS localization accuracy classification: A context-based approach, IEEE Trans. Intell. Transp. Syst. 14 (2013) 262–273.
[78] E. Zhang, N. Masoud, Increasing GPS localization accuracy with reinforcement learning, IEEE Trans. Intell. Transp. Syst. 22 (2020) 2615–2626.
[79] L. Douadi, Y. Dupuis, P. Vasseur, Stable keypoints selection for 2D LiDAR based place recognition with map data reduction, Robotica 40 (11) (2022) 3786–3810.
[80] J. Zhang, S. Singh, Low-drift and real-time lidar odometry and mapping, Auton. Robots 41 (2017) 401–416.
[81] A. Pfrunder, P.V. Borges, A.R. Romero, G. Catt, A. Elfes, Real-time autonomous ground vehicle navigation in heterogeneous environments using a 3D lidar, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2017, pp. 2601–2608.
[82] L. He, W. Li, Y. Guan, H. Zhang, IGICP: Intensity and geometry enhanced LiDAR odometry, IEEE Trans. Intell. Veh. 9 (2024) 541–554.
[83] Y. Wu, F. Tang, H. Li, Image-based camera localization: an overview, Vis. Comput. Ind. Biomed. Art 1 (2018) 1–13.
[84] E. Brachmann, C. Rother, Learning less is more - 6D camera localization via 3D surface regression, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2017, pp. 4654–4662.
[85] K. Guan, L. Ma, X. Tan, S. Guo, Vision-based indoor localization approach based on SURF and landmark, in: 2016 International Wireless Communications and Mobile Computing Conference, IWCMC, 2016, pp. 655–659.
[86] Y. Wu, J. Kuang, X. Niu, Wheel-INS2: Multiple MEMS IMU-based dead reckoning system with different configurations for wheeled robots, IEEE Trans. Intell. Transp. Syst. 24 (2020) 3064–3077.
[87] Y. Wu, X. Niu, J. Kuang, A comparison of three measurement models for the wheel-mounted MEMS IMU-based dead reckoning system, IEEE Trans. Veh. Technol. 70 (2020) 11193–11203.
[88] S. Zhao, Y. Chen, J.A. Farrell, High-precision vehicle navigation in urban environments using an MEM's IMU and single-frequency GPS receiver, IEEE Trans. Intell. Transp. Syst. 17 (2016) 2854–2867.
[89] J.J. Patoliya, H.K. Mewada, M. Hassaballah, M.A. Khan, S. Kadry, A robust autonomous navigation and mapping system based on GPS and LiDAR data for unconstraint environment, Earth Sci. Inform. 15 (2022) 2703–2715.
[90] H. Li, F. Nashashibi, G. Toulminet, Localization for intelligent vehicle by fusing mono-camera, low-cost GPS and map data, in: 13th International IEEE Conference on Intelligent Transportation Systems, 2010, pp. 1657–1662.
[91] J.A. Hesch, D.G. Kottas, S.L. Bowman, S.I. Roumeliotis, Camera-IMU-based localization: Observability analysis and consistency improvement, Int. J. Robot. Res. 33 (2014) 182–201.
[92] K. Li, Z. Ouyang, L. Hu, D. Hao, L. Kneip, Robust SRIF-based LiDAR-IMU localization for autonomous vehicles, in: 2021 IEEE International Conference on Robotics and Automation, ICRA, 2021, pp. 5381–5387.
[93] X. Liu, S. Wen, Z. Jiang, W. Tian, T.Z. Qiu, K.M. Othman, A multisensor fusion with automatic vision–LiDAR calibration based on factor graph joint optimization for SLAM, IEEE Trans. Instrum. Meas. 72 (2023) 1–9.
[94] X. Liu, S. Wen, J. Zhao, T.Z. Qiu, H. Zhang, Edge-assisted multi-robot visual-inertial SLAM with efficient communication, IEEE Trans. Autom. Sci. Eng. (2024).
[95] Z. Zhu, X. Lin, Z. Su, S. Mao, H. Zhu, X. Zhou, Disturbance-resistant camera-LiDAR fusion for robust three-dimentional object detection, in: IEEE International Conference on Robotics and Biomimetics, ROBIO, 2022, pp. 705–710.
[96] H. Yin, Z. Lin, J.K. Yeoh, Semantic localization on BIM-generated maps using a 3D LiDAR sensor, Autom. Constr. 146 (2023) 104641.
[97] Y. Shen, Z. Jiao, A novel self-positioning based on feature map creation and laser location method for RBPF-slam, J. Robot. 2021 (2021) 9988916:1–9988916:11.
[98] A. Das, S.L. Waslander, Scan registration using segmented region growing NDT, Int. J. Robot. Res. 33 (2014) 1645–1663.
[99] J. Gaffuri, Toward web mapping with vector data, in: International Conference Geographic Information Science, 2012, pp. 87–101.
[100] L. Liu, H. Li, Y. Dai, Efficient global 2D-3D matching for camera localization in a large-scale 3D map, in: 2017 IEEE International Conference on Computer Vision, ICCV, 2017, pp. 2391–2400.
[101] M. Wang, M. Cong, Y. Du, D. Liu, X. Tian, Multi-robot raster map fusion without initial relative position, Robot. Intell. Autom. 43 (5) (2023) 498–508.
[102] J. Chen, Y. Wu, J. Tan, H. Ma, Y. Furukawa, MapTracker: Tracking with strided memory fusion for consistent vector HD mapping, 2024, ArXiv, arXiv:2403.15951.
[103] Z. Wang, W. Zhan, M. Tomizuka, Fusing bird's eye view LIDAR point cloud and front view camera image for 3D object detection, in: 2018 IEEE Intelligent Vehicles Symposium, IV, 2018, pp. 1–6.
[104] K. Kim, S. Cho, W. Chung, HD map update for autonomous driving with crowdsourced data, IEEE Robot. Autom. Lett. 6 (2021) 1895–1901.
[105] Q. Zou, M. Sester, Incremental map refinement of building information using LiDAR point clouds, Int. Arch. Photogramm. Rem. Sens. Spatial Inform. Sci. 43 (2021) 277–282.
[106] S. Song, H. Lim, A.J. Lee, H. Myung, DynaVINS: A visual-inertial SLAM for dynamic environments, IEEE Robot. Autom. Lett. 7 (2022) 11523–11530.
[107] S. Ji, W. Chen, Z. Su, Y. Guan, J. Li, H. Zhang, H. Zhu, A point-to-distribution degeneracy detection factor for LiDAR SLAM using local geometric models, in: IEEE International Conference on Robotics and Automation, ICRA, 2024, pp. 12283–12289.
[108] H. Cho, S. Yeon, H. Choi, N.L. Doh, Detection and compensation of degeneracy cases for IMU-kinect integrated continuous SLAM with plane features †, Sensors (Basel, Switzerland) 18 (2018).
[109] Z. Zhang, J. Zhao, C. Huang, L. Li, Learning visual semantic map-matching for loosely multi-sensor fusion localization of autonomous vehicles, IEEE Trans. Intell. Veh. 8 (2023) 358–367.
[110] Z. Zhang, J. Yu, J. Tang, Y. Xu, Y. Wang, MR-TopoMap: Multi-robot exploration based on topological map in communication restricted environment, IEEE Robot. Autom. Lett. 7 (2022) 10794–10801.
[111] G. He, Q. Zhang, Y. Zhuang, Online semantic-assisted topological map building with LiDAR in large-scale outdoor environments: Toward robust place recognition, IEEE Trans. Instrum. Meas. 71 (2022) 1–12.
[112] H.C. Thomas, E.L. Charles, L.R. Ronald, S. Clifford, Section 24.3: Dijkstra's algorithm, Introd. Algorithms (2001) 595–601.
[113] X. Bai, W. Yan, M. Cao, D. Xue, Distributed multi-vehicle task assignment in a time-invariant drift field with obstacles, IET Control Theory Appl. 13 (17) (2019) 2886–2893.
[114] S.M. LaValle, Planning Algorithms, Cambridge University Press, 2006.
[115] A. Erokhin, V. Erokhin, S. Sotnikov, A. Gogolevsky, Optimal multi-robot path finding algorithm based on a, in: Intelligent Systems in Cybernetics and Automation Control Theory 2, Springer, 2019, pp. 172–182.
[116] G. Sun, R. Zhou, B. Di, Z. Dong, Y. Wang, A novel cooperative path planning for multi-robot persistent coverage with obstacles and coverage period constraints, Sensors 19 (9) (2019) 1994.
[117] S.M. LaValle, Rapidly-exploring random trees : a new tool for path planning, Ann. Res. Rep. (1998) URL https://api.semanticscholar.org/CorpusID:14744621.
[118] S. Lavalle, J. Kuffner, Rapidly-exploring random trees: Progress and prospects, in: Algorithmic and computational robotics: New directions, 2000.
[119] W. Wang, X. Xu, Y. Li, J. Song, H. He, Triple RRTs: An effective method for path planning in narrow passages, Adv. Robot. 24 (7) (2010) 943–962.
[120] G. Kang, Y.B. Kim, W.S. You, Y.H. Lee, H.S. Oh, H. Moon, H.R. Choi, Sampling-based path planning with goal oriented sampling, in: 2016 IEEE International Conference on Advanced Intelligent Mechatronics, AIM, 2016, pp. 1285–1290.
[121] F. Peng, Y. Zhao, Random triangle sampling path planning of assembly/disassembly in environment with dangerzones, in: 2010 International Conference on Measuring Technology and Mechatronics Automation, 2010, pp. 972–976.
[122] R. Motwani, M. Motwani, F.C. Harris, Uniform and efficient exploration of state space using kinodynamic sampling-based planners, in: Computational Kinematics, 2014, pp. 67–74.

[123] W. Chi, Z. Ding, J. Wang, G. Chen, L. Sun, A generalized voronoi diagram-based efficient heuristic path planning method for RRTs in mobile robots, IEEE Trans. Ind. Electron. 69 (5) (2022) 4926–4937.
[124] P. Wan, J. Wen, C. Wu, A discriminating method of driving anger based on sample entropy of eeg and BVP, in: 2015 International Conference on Transportation Information and Safety, ICTIS, 2015, pp. 156–161.
[125] Z. Littlefield, Y. Li, K.E. Bekris, Efficient sampling-based motion planning with asymptotic near-optimality guarantees for systems with dynamics, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2013, pp. 1779–1785.
[126] J.D. Gammell, S.S. Srinivasa, T.D. Barfoot, Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic, in: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014, pp. 2997–3004.
[127] A.H. Qureshi, M.C. Yip, Deeply informed neural sampling for robot motion planning, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2018, pp. 6582–6588.
[128] S. Rifai, P. Vincent, X. Muller, X. Glorot, Y. Bengio, Contractive auto-encoders: explicit invariance during feature extraction, in: International Conference on Machine Learning, ICML, 2011, pp. 833–840.
[129] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.
[130] L. Chen, Y. Shan, W. Tian, B. Li, D. Cao, A fast and efficient double-tree RRT*-like sampling-based planner applying on mobile robotic systems, IEEE/ASME Trans. Mechatronics 23 (6) (2018) 2568–2578.
[131] S. Karaman, E. Frazzoli, Optimal kinodynamic motion planning using incremental sampling-based methods, in: IEEE Conference on Decision and Control, 2010, pp. 7681–7687.
[132] J. Bruce, M. Veloso, Real-time randomized path planning for robot navigation, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002, pp. 2383–2388.
[133] D. Ferguson, N. Kalra, A. Stentz, Replanning with RRTs, in: IEEE International Conference on Robotics and Automation, 2006, pp. 1243–1248.
[134] M. Zucker, J. Kuffner, M. Branicky, Multipartite RRTs for rapid replanning in dynamic environments, in: IEEE International Conference on Robotics and Automation, 2007, pp. 1603–1609.
[135] A. Stentz, Optimal and efficient path planning for partially-known environments, in: Proceedings of the 1994 IEEE International Conference on Robotics and Automation, IEEE, 1994, pp. 3310–3317.
[136] Z. Ren, S. Rathinam, M. Likhachev, H. Choset, Multi-objective path-based d* lite, IEEE Robot. Autom. Lett. 7 (2) (2022) 3318–3325.
[137] O. Adiyatov, H.A. Varol, A novel RRT*-based algorithm for motion planning in dynamic environments, in: IEEE International Conference on Mechatronics and Automation, 2017, pp. 1416–1421.
[138] Z. Du, S. Liu, Asymptotical RRT-based path planning for mobile robots in dynamic environments, in: 2018 37th Chinese Control Conference, CCC, 2018, pp. 5281–5286.
[139] M. Otte, E. Frazzoli, RRT^X: Asymptotically optimal single-query sampling-based motion planning with quick replanning, Int. J. Robot. Res. 35 (7) (2015) 797–822.
[140] A.H. Qureshi, S. Mumtaz, W. Khan, A.A.A. Sheikh, K.F. Iqbal, Y. Ayaz, O. Hasan, Augmenting RRT*-planner with local trees for motion planning in complex dynamic environments, in: 2014 19th International Conference on Methods and Models in Automation and Robotics, MMAR, 2014, pp. 657–662.
[141] Y. Chen, J. Yu, X. Su, G. Luo, Path planning for multi-UAV formation, J. Intell. Robot. Syst. 77 (2015) 229–246.
[142] X. Wang, A. Sahin, S. Bhattacharya, Coordination-free multi-robot path planning for congestion reduction using topological reasoning, J. Intell. Robot. Syst. 108 (3) (2023) 50.
[143] C. He, Y. Wan, Y. Gu, F.L. Lewis, Integral reinforcement learning-based multi-robot minimum time-energy path planning subject to collision avoidance and unknown environmental disturbances, IEEE Control Syst. Lett. 5 (3) (2020) 983–988.
[144] D. Connell, H.M. La, Dynamic path planning and replanning for mobile robots using RRT, in: 2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC, 2017, pp. 1429–1434.
[145] B. Chandler, M.A. Goodrich, Online RRT* and online FMT*: Rapid replanning with dynamic cost, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2017, pp. 6313–6318.
[146] K. Naderi, RT-RRT*:a real-time path planning algorithm based on RRT*, in: Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games, 2015, pp. 113–118.
[147] C. Fulgenzi, C. Tay, A. Spalanzani, C. Laugier, Probabilistic navigation in dynamic environment using rapidly-exploring random trees and Gaussian processes, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, pp. 1056–1062.
[148] W. Chi, M.Q.-H. Meng, Risk-RRT*: A robot motion planning algorithm for the human robot coexisting environment, in: 2017 18th International Conference on Advanced Robotics, 2017, pp. 583–588.
[149] W. Mechlih, Trajectory planning for mobile robots in a dynamic environment, in: Proceedings of VNIS '93 - Vehicle Navigation and Information Systems Conference, 1993, pp. 551–554.
[150] R. Firoozi, A. Mir, G.S. Camps, M. Schwager, Occlusion-aware mpc for guaranteed safe robot navigation with unseen dynamic obstacles, ArXiv abs/2211.09156 (2022) http://dx.doi.org/10.48550/arXiv.2211.09156.
[151] M. Garzón, E.P. Fotiadis, A. Barrientos, A. Spalanzani, RiskRRT-based planning for interception of moving objects in complex environments, in: ROBOT2013: First Iberian Robotics Conference, 2014, pp. 489–503.
[152] C. Fulgenzi, A. Spalanzani, C. Laugier, C. Tay, Risk based motion planning and navigation in uncertain dynamic environment, Res. Rep. (2010) 14.
[153] W. Chi, H. Kono, Y. Tamura, A. Yamashita, H. Asama, M.Q.-H. Meng, A human-friendly robot navigation algorithm using the risk-RRT approach, in: 2016 IEEE International Conference on Real-Time Computing and Robotics, RCAR, 2016, pp. 227–232.
[154] C. Tay, C. Laugier, Modelling smooth paths using Gaussian processes, Springer Tracts Adv. Robot. 42 (2007) 381–390.
[155] J. Rios-Martinez, A. Spalanzani, C. Laugier, Understanding human interaction for probabilistic autonomous navigation using risk-RRT approach, in: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011, pp. 2014–2019.
[156] W. Chi, H. Kono, Y. Tamura, A. Yamashita, A. Hajime, M.Q.-H. Meng, A human-friendly robot navigation algorithm using the risk-RRT approach, in: 2016 IEEE International Conference on Real-Time Computing and Robotics, RCAR, 2016, pp. 227–232.
[157] W. Chi, C. Wang, J. Wang, M.Q.-H. Meng, Risk-DTRRT-based optimal motion planning algorithm for mobile robots, IEEE Trans. Autom. Sci. Eng. 16 (3) (2019) 1271–1288.
[158] B.D. Argall, S. Chernova, M. Veloso, B. Browning, A survey of robot learning from demonstration, Robot. Auton. Syst. 57 (5) (2009) 469–483.
[159] S.S. Samsani, H. Mutahira, M.S. Muhammad, Memory-based crowd-aware robot navigation using deep reinforcement learning, Complex Intell. Syst. 9 (2) (2023) 2147–2158.
[160] J. Qin, J. Qin, J. Qiu, Q. Liu, M. Li, Q. Ma, SRL-ORCA: A socially aware multi-agent mapless navigation algorithm in complex dynamic scenes, IEEE Robot. Autom. Lett. 9 (1) (2023) 143–150.
[161] E. Sisbot, L.F. Marin-Urias, R. Alami, T. Siméon, A human aware mobile robot motion planner, IEEE Trans. Robot. 23 (5) (2007) 874–883.
[162] N. Pérez-Higueras, F. Caballero, M. Luis, Learning human-aware path planning with fully convolutional networks, in: 2018 IEEE International Conference on Robotics and Automation, ICRA, 2018, pp. 5897–5902.
[163] A.Y. Ng, S. Russell, Algorithms for inverse reinforcement learning, in: International Conference on Machine Learning, 2000, pp. 663–670.
[164] N. Pérez-Higueras, F. Caballero, M. Luis, Teaching robot navigation behaviors to optimal RRT planners, Int. J. Soc. Robot. 10 (2018) 235–249.
[165] N. Pérez-Higueras, F. Caballero, L. Merino, Learning robot navigation behaviors by demonstration using a RRT* planner, in: International Conference on Social Robotics, 2016, pp. 1–10.
[166] B.D. Ziebart, A.L. Maas, J.A. Bagnell, A.K. Dey, Maximum entropy inverse reinforcement learning, in: Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, 2008, pp. 1433–1438.
[167] K. Shiarlis, J. Messias, S. Whiteson, Rapidly exploring learning trees, in: 2017 IEEE International Conference on Robotics and Automation, ICRA, 2017.
[168] Z. Chen, H. Wu, Y. Chen, L. Cheng, B. Zhang, Patrol robot path planning in nuclear power plant using an interval multi-objective particle swarm optimization algorithm, Appl. Soft Comput. 116 (2022) 108192.
[169] Y. Chen, S. Ren, Z. Chen, M. Chen, H. Wu, Path planning for vehicle-borne system consisting of multi air–ground robots, Robotica 38 (3) (2020) 493–511.
[170] B. Tang, K. Xiang, M. Pang, Z. Zhanxia, Multi-robot path planning using an improved self-adaptive particle swarm optimization, Int. J. Adv. Robot. Syst. 17 (5) (2020) 1729881420936154.
[171] B. Sahu, P.K. Das, M. ranjan Kabat, Multi-robot cooperation and path planning for stick transporting using improved Q-learning and democratic robotics PSO, J. Comput. Sci. 60 (2022) 101637.
[172] R.A. Saeed, D. Reforgiato Recupero, P. Remagnino, The boundary node method for multi-robot multi-goal path planning problems, Expert Syst. 38 (6) (2021) e12691.
[173] W. Xu, Q. Wang, M. Yu, D. Zhao, Path planning for multi-AGV systems based on two-stage scheduling, Int. J. Performabil. Eng. 13 (8) (2017) 1347.
[174] H. Huang, T. Zhuo, Multi-model cooperative task assignment and path planning of multiple UCAV formation, Multimedia Tools Appl. 78 (2019) 415–436.

[175] Y.F. Chen, M. Liu, M. Everett, J.P. How, Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning, in: 2017 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2017, pp. 285–292.
[176] D. Wang, T. Fan, T. Han, J. Pan, A two-stage reinforcement learning approach for multi-UAV collision avoidance under imperfect sensing, IEEE Robot. Autom. Lett. 5 (2) (2020) 3098–3105.
[177] H. Qie, D. Shi, T. Shen, X. Xu, Y. Li, L. Wang, Joint optimization of multi-UAV target assignment and path planning based on multi-agent reinforcement learning, IEEE Access 7 (2019) 146264–146272.
[178] Y. Jia, Y. Song, B. Xiong, J. Cheng, W. Zhang, S.X. Yang, S. Kwong, Hierarchical perception-improving for decentralized multi-robot motion planning in complex scenarios, IEEE Trans. Intell. Transp. Syst. 25 (7) (2024) 6486–6500.
[179] W. Ou, B. Luo, X. Xu, Y. Feng, Y. Zhao, Reinforcement learned multi–agent cooperative navigation in hybrid environment with relational graph learning, IEEE Trans. Artif. Intell. (2024).
[180] J. Zhao, H. Ye, Y. Zhan, H. Zhang, Human orientation estimation under partial occlusion, 2024, arXiv preprint arXiv:2404.14139.
[181] H. Ye, J. Zhao, Y. Pan, W. Cherr, L. He, H. Zhang, Robot person following under partial occlusion, in: 2023 IEEE International Conference on Robotics and Automation, ICRA, 2023, pp. 7591–7597.
[182] Z. Mai, R. Li, J. Jeong, D. Quispe, H. Kim, S. Sanner, Online continual learning in image classification: An empirical survey, Neurocomputing 469 (2022) 28–51.
[183] H. Ye, J. Zhao, Y. Zhan, W. Chen, L. He, H. Zhang, Person re-identification for robot person following with online continual learning, IEEE Robot. Autom. Lett. (2024).