智能系统学报
CAAI TRANSACTIONS ON INTELLIGENT SYSTEMS

基于深度强化学习的室内视觉局部路径规划

朱少凯, 孟庆浩, 金晟, 戴旭阳

引用本文:
朱少凯,孟庆浩,金晟,戴旭阳. 基于深度强化学习的室内视觉局部路径规划[J]. 智能系统学报, 2022, 17(5): 908–918.
ZHU Shaokai,MENG Qinghao,JIN Sheng,DAI Xuyang. Indoor visual local path planning based on deep reinforcement learning[J]. *CAAI Transactions on Intelligent Systems*, 2022, 17(5): 908–918.

在线阅读 View online: https://dx.doi.org/10.11992/tis.202107059

DOI: 10.11992/tis.202107059
网络出版地址: https://kns.cnki.net/kcms/detail/23.1538.TP.20220623.1044.004.html

基于深度强化学习的室内视觉局部路径规划

朱少凯, 孟庆浩, 金晟, 戴旭阳
(天津大学 电气自动化与信息工程学院 机器人与自主系统研究所, 天津 300072)

摘 要: 传统的机器人局部路径规划方法多为已有先验地图的情况设计, 导致其在与视觉 (simultaneous localization and mapping, SLAM) 结合的导航中效果不佳。为此传统的机器人局部路径规划方法多为已有先验地图的情况设计, 导致其在与视觉 SLAM 结合的导航中效果不佳。为此, 本文提出一种基于深度强化学习的视觉局部路径规划策略。首先, 基于视觉同时定位与建图 (SLAM) 技术建立周围环境的栅格地图, 并使用 A*算法规划全局路径; 其次, 综合考虑避障、机器人行走效率、位姿跟踪等问题, 构建基于深度强化学习的局部路径规划策略, 设计以前进、左转、右转为基本元素的离散动作空间, 以及基于彩色图、深度图、特征点图等视觉观测的状态空间, 利用近端策略优化 (proximal policy optimization, PPO) 算法学习和探索最佳状态动作映射网络。Habitat 仿真平台运行结果表明, 所提出的局部路径规划策略能够在实时创建的地图上规划出一条最优或次优路径。相比于传统的局部路径规划算法, 平均成功率提高了 53.9%, 位姿跟踪丢失率减小了 66.5%, 碰撞率减小了 30.1%。

关键词: 视觉导航; 深度学习; 强化学习; 局部路径规划; 避障; 视觉 SLAM; 近端策略优化; 移动机器人
中图分类号: TP391 **文献标志码:** A **文章编号:** 1673−4785(2022)05−0908−11

中文引用格式: 朱少凯, 孟庆浩, 金晟, 等. 基于深度强化学习的室内视觉局部路径规划 [J]. 智能系统学报, 2022, 17(5): 908–918.
英文引用格式: ZHU Shaokai, MENG Qinghao, JIN Sheng, et al. Indoor visual local path planning based on deep reinforcement learning[J]. CAAI transactions on intelligent systems, 2022, 17(5): 908–918.

Indoor visual local path planning based on deep reinforcement learning

ZHU Shaokai, MENG Qinghao, JIN Sheng, DAI Xuyang
(Institute of Robotics and Autonomous Systems, School of Electrical and Information Engineering, Tianjin University, 300072, China)

Abstract: Traditional robot local path planning methods are mostly designed for situations with prior maps, thus leading to poor results in navigation when combined with visual simultaneous localization and mapping (SLAM). Therefore, this paper proposes a visual local path planning strategy based on deep reinforcement learning. First, a grid map of the surrounding environment is built based on the visual SLAM technology, and the global path is planned using the A* algorithm. Second, considering the problems of obstacle avoidance, robot walking efficiency, and pose tracking, a local path planning strategy is constructed based on deep reinforcement learning to design the discrete action space with forward movement, left turn, and right turn as the basic elements, as well as the state space based on visual observation maps, such as color, depth, and feature point maps. The proximal policy optimization (PPO) algorithm is used to learn and explore the best state–action mapping network. The running results of the habitat simulation platform show that the proposed local path planning strategy can design an optimal or sub-optimal path on a map generated in real time. Compared with traditional local path planning algorithms, the average success rate of the proposed strategy is increased by 53.9%, and the average tracking failure rate and collision rate are reduced by 66.5% and 30.1%, respectively.

Keywords: visual navigation; deep learning; reinforcement learning; local path planning; obstacle avoidance; visual SLAM; proximal policy optimization (PPO); mobile robot

收稿日期: 2021−07−27. **网络出版日期:** 2022−06−24.
基金项目: 中国博士后科学基金项目 (2021M692390); 天津市自然科学基金项目 (20JCZDJC00150, 20JCYBJC00320).
通信作者: 金晟. E-mail: shengjin@tju.edu.cn.

视觉导航是一类新兴的导航技术, 具有使用成本低、获取信息丰富的优点, 成为近些年来机器人领域的研究热点之一[1-4]。路径规划是移动

机器人实现自主视觉导航的关键技术之一, 分为全局路径规划算法和局部路径规划算法[5]。常用的全局路径规划算法有 A*算法[6] 和 D*算法[7] 等。局部路径规划算法根据全局路径和部分环境信息, 输出机器人运动控制指令, 使机器人大致沿着全局路径的轨迹移动。动态窗口法 (dynamic window algorithm, DWA) 作为一种广泛使用的局部路径规划算法, 具有良好的避障能力, 且计算量小、实时性高, 是机器人操作系统 (robot operating system, ROS) 的默认局部路径规划算法[8]。但是, 当环境变得复杂时, DWA 算法容易陷入局部极小值点, 同时无法保证规划出的路径是最优的。时间弹性带 (timed elastic band, TEB) 算法是另一种广泛使用的局部路径规划算法, 采用图优化的方法迭代求解局部路径规划问题, 有较高的操作性[9]。模型预测控制 (model predictive control, MPC) 能根据机器人当前的运动状态预测其未来几个时间步的轨迹, 通过二次规划方法来优化, 求出一个最优局部路径规划解[10]。但是, TEB 和 MPC 算法有非常多的参数需要设置, 所以在复杂的环境应用时需要配备较高性能的计算机和花大量时间手工调参。综上所述, 传统的局部路径规划方法存在参数调整耗时, 缺乏对新环境的泛化能力等问题。此外, 当与视觉 SLAM 协同工作时, 传统局部路径规划算法仅仅考虑机器人运动代价等因素, 没有考虑机器人在视觉 SLAM 过程中易在低纹理区域跟踪丢失的问题, 以上原因导致传统局部路径规划算法应用于基于视觉 SLAM 的导航时表现较差。

深度强化学习自提出以来逐渐得到国内外学者的广泛关注, 其相关理论和应用研究都得到了不同程度的发展[11-12]。由于其“交互式学习”和“试错学习”的特点, 适用于很多问题的决策, 已成为机器人控制领域的研究热点, 其中也包括局部路径规划任务。张福海等[13] 以激光雷达作为环境感知器, 并构造了基于 Q-learning 的强化学习模型, 将其应用在了局部路径规划任务中, 提高了移动机器人对未知环境的适应性。Guldenring 等[14] 利用激光雷达来获取动态环境信息, 并根据环境数据基于 PPO 的强化学习算法进行局部路径规划。Balakrishnan 等[15] 在 A*全局路径规划算法基础上, 利用深度强化学习训练了一种局部路径规划策略, 以到达局部目标点。然而, 该方法依赖于真实先验地图。Chaplot 等[16] 同样训练了一种基于深度强化学习的局部路径规划策略, 并与全局策略相结合, 以完成视觉探索任务。但是该方法依赖于真实先验位姿。在实际的视觉导航任务中, 机器人需要依靠视觉传感器来获取位姿和环境地图。因此, 如何合理地设计局部路径规划机制, 使局部路径规划能够和视觉 SLAM 方法更好地配合, 最大限度地避免碰撞和位姿跟踪丢失、提升导航成功率是这类方法的关键, 也是至今仍未被探索的问题。

针对以上问题, 本文在视觉 SLAM 和全局路径规划算法的基础上, 提出一种基于深度强化学习的室内视觉局部路径规划策略。该策略在强化学习 PPO[17](proximal policy optimization) 算法的基础上, 充分考虑了机器人避障、防止视觉 SLAM 跟踪丢失以及机器人行走效率等多方面因素, 设计奖励函数和网络结构, 在大量的场景下学习最佳状态–动作映射网络, 提高移动机器人导航成功率。既避免了部分传统路径规划算法调参复杂的问题, 又具有很好的泛化性, 且与视觉 SLAM 模块契合。最终, 在三维物理仿真平台 Habitat[18] 中利用机器人对该局部路径规划策略进行相关仿真分析, 证实了所提出策略的有效性。

本文的创新点主要包括: 1)提出了一种基于深度强化学习的移动机器人室内视觉局部路径规划算法, 合理地设计了环境交互机制与观测的状态空间; 2)研究了多样的奖励函数, 加快了算法的收敛速度, 提高了模型的性能, 最大限度地避免了碰撞和位姿跟踪的丢失、可尽快到达局部目标点。3)将局部路径规划模型融入总体导航框架, 与视觉 SLAM 模块、全局路径规划、仿真平台相互配合, 有助于长距离室内复杂场景下的点导航。

1　问题描述

机器人在室内导航的过程中, 在低纹理区域易发生视觉 SLAM 跟踪失败现象。因此, 考虑机器人快速接近局部目标点的同时, 还要兼顾低纹理区域、障碍物等诸多不利因素对于视觉导航任务造成的影响。本文设计的局部目标点导航策略, 可以实现规避障碍物、保证跟踪稳定性以避免视觉 SLAM 失败、成功到达局部目标点的目的。

局部路径规划策略是与视觉 SLAM 模块、全局路径规划、仿真平台相互配合的, 它们的关系如图 1 所示。首先, 选用 Habitat 仿真平台, 机器人在该平台中能以实体的形式存在。该平台能实时地提供机器人在当前位置所采集到的彩色图、深度图, 并实时检测机器人是否发生碰撞等。对于每一个导航任务, 仿真平台会给定机器人的初始位置和机器人距离全局目标点的相对位置。其

次，机器人通过视觉 SLAM 模块与三维重建技术[19-21]得到室内三维环境的稠密点云地图，然后将一定高度范围内的点云投影之后，获得用于导航的二维栅格地图。此外，视觉 SLAM 模块还能实时提供机器人在当前位置所采集到的 ORB (oriented FAST and rotated BRIEF) 特征点图[22]并实时监测机器人当前的位姿跟踪状态（跟踪正常/跟踪丢失）。然后，在得到的二维栅格地图基础上，使用 A*算法规划全局路径。局部目标点位置的确定依赖于全局路径，如图 2 所示，以机器人当前位置为圆心，半径为 r 的圆与全局路径的交点即为局部路径规划目标点。最后，局部路径规划策略的目标是使机器人从起点成功移动到局部目标点的位置，同时兼顾避障、避免跟踪失败的目的。在本文中，局部路径规划策略的输入是当前时刻的观测（包括：彩色图、深度图、特征点图、当前位置与局部目标点之间的距离和方向），输出是机器人当前时刻所采取的动作。训练目标是避免发生碰撞和视觉 SLAM 跟踪丢失的同时，采取尽量少的动作运动到局部目标点。当机器人到达局部目标点或机器人行动步数达到步数限制时，则开始规划下一个局部目标点，以此循环，直到机器人到达全局目标点。当机器人在行走过程中发生碰撞或视觉 SLAM 跟踪丢失时，则认为导航任务失败。

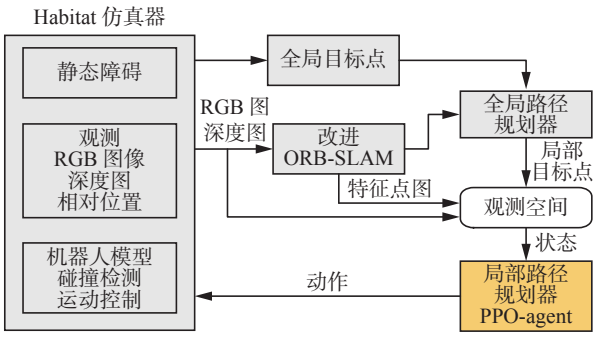


图 1　总体导航架构

Fig. 1　Overall navigation framework

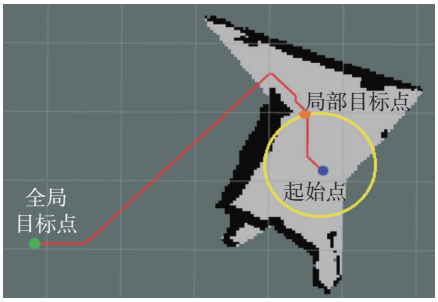


图 2　局部导航示意图

Fig. 2　Schematic diagram of the local navigation

2　基于深度强化学习的路径规划

2.1　模型建立

大多数强化学习可以用部分可观测马尔科夫决策过程 (POMDP) 描述，POMDP 用一个五元组 (S,A,P,R,γ)来表示。其中 S 为机器人状态空间，A 为机器人动作空间，P 为从一个状态到另一个状态的转换概率，R 为奖励，γ表示折扣因子。常用演员–评论家 (actor-critic, AC) 算法来解决该部分可观测的马尔科夫决策问题。AC 算法基于策略梯度方法，并通过最大化平均回报来直接更新策略。在执行策略参数π时，累积每个时间步动作的预期奖励来计算价值函数，如式 (1) 所示。随后，按式 (2) 的贝尔曼方程迭代直至策略参数收敛至最优。

$$V^*(s_t)=\mathrm{E}[\Sigma\gamma R(s_t,\pi(s_t))] \tag{1}$$

$$\pi^*(s_{t+1})=\mathrm{argmax}\left(R(s_t,\pi(s_t))+\int_{x_{t=1}}P(s_t,s_{t+1}\mid\pi(s_t))\cdot V^*(s_{t+1})\mathrm{d}s_{t+1}\right) \tag{2}$$

式 (1) 和 (2) 中，E 为求期望函数，s_t 代表 t 时刻机器人观测状态，$R(s_t,\pi(s_t))$ 为当前策略在状态 s_t 下的奖励值，$V^*(s_t)$为最优状态值函数，γ 为折扣因子。$P(s_t,s_{t+1}\mid\pi(s_t))$为状态转移概率。对于从状态 s_t 可以到达的任何状态 s_{t+1}，$\pi'(s_{t+1})$为能够使得状态 s_{t+1} 获得最优价值的策略。

2.2　框架设计

深度强化学习中，智能体 (Agent) 负责与环境交互，是算法的应用对象。外部环境 (Env) 是客观存在，用于制定奖惩规则和交互机制。本文将局部路径规划问题转换到深度强化学习的框架中进行求解。在交互过程中，需要定义相应的接口标准，主要有 step、reset、render 3 个函数。

1) step 函数：负责制定 Agent 与 Env 的交互机制和奖惩规则，在基于深度强化学习的局部路径规划策略根据当前状态s_i求解出动作a_1后，该函数能够返回执行a_1后的奖励r_i，以及是否结束的标志位 done。此外，该函数能返回下一时刻的状态 s_{t+1}。该函数是将局部路径规划问题转换到深度强化学习问题的关键，是算法设计中最重要的函数，$(s_i,a_t,r_t,s_{i+1},\mathrm{done})$构成了局部路径规划策略训练所需要的基本数据单元。

2) reset 函数：定义一个回合 (episode) 为 Agent 与 Env 的一次完整交互。Agent 通过与 Env 多次交互积累经验，从而学到好的局部路径规划策略。回合结束的判断标志位为 done，当其为 True 时，回合结束。此时，Agent 和 Env 需要通过该 re-

set 函数重新初始化, 开始新一轮的交互。为了保证 Agent 和 Env 的可持续性交互, 回合结束在机器人到达局部目标点、发生碰撞、视觉 SLAM 跟踪失败、到达每回合的最大步数限制时触发。

3) render 函数: 局部路径规划策略需要与视觉 SLAM 模块、全局路径规划、仿真平台相互配合。为了便于算法调试, 使用 render 函数来输出可视化窗口显示机器人当前的状态及所处环境, 如彩色图 (RGB)、深度图 (Depth)、全局地图、路径规划等。

step 函数、reset 函数、render 函数构成了 Env 部分。step 函数负责 Agent 与 Env 之间的交互, 其中嵌套了负责重置的 reset 函数和负责显示功能的 render 函数。

2.3　可观测状态与奖励函数设计

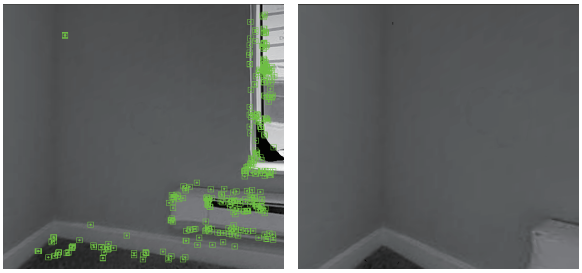
step 函数是基于深度强化学习的局部路径规划策略实施的重点, 其中涉及到两个关键问题: 一是如何描述可观测状态空间, 另一个是如何设计奖励函数。前者反映了机器人在局部路径规划的实施过程中需要注意的环境信息, 后者能指导局部路径规划策略向目标方向更新。

在机器人的局部路径规划任务中, Agent 所能观测到的状态 S 来源于自身所装备的 RGB-D 相机以及视觉 SLAM 模块, 具体可用一个五元组 (s_{rgb}, s_{depth}, s_{dis}, s_{angle}, s_{mpt}) 来表示。其中 s_{rgb} 和 s_{depth} 来源于 RGB-D 相机, 分别表示彩色图和深度图; s_{dis} 表示机器人与局部目标点的相对距离, 计算方法如式 (3) 所示; s_{angle} 为机器人当前朝向角与机器人和局部目标点连线角度的差值, 设机器人在二维栅格地图上的坐标为 (x_1, y_1), 朝向角为 β, 局部目标点坐标为 (x_2, y_2), s_{angle} 计算方式如式 (4):

$$s_{\text{dis}} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3}$$

$$\left| s_{\text{angle}} \right| = \arccos\left(\frac{\cos(\beta)(y_2 - y_1) + \sin(\beta)(x_2 - x_1)}{s_{\text{dis}}} \right) \tag{4}$$

s_{mpt} 为采样时刻当前帧的特征点图的位置矩阵, 包含了特征点在图像帧上的位置和数量信息。在 ORB-SLAM2[23] 算法中, ORB 特征点的匹配是实现前端视觉里程计的基础, 特征点在图像中的位置和数量包含了视觉 SLAM 跟踪稳定的信息。如图 3 所示, 图 3 (b) 由图 (a) 采样位置向左旋转 30°后得到。在图 3(a) 所示采样位置视觉 SLAM 跟踪成功, 在图 3(b) 所示采样位置则跟踪丢失。由图 3(a) 可知, 特征点集中于右半部分, 左半部分特征较为稀疏, 相机此时左转视觉 SLAM 跟踪失败的风险较高。定义动作空间 $A(a_l, a_f, a_r)$, 其 3 个元素分别代表机器人左转、前行和右转 3 个动作, 每个动作执行时间为 1 s, 左转和右转的角速度为 0.3 rad/s, 前进速度为 0.1 m/s。



(a) 跟踪正常　　(b) 跟踪失败

图 3　ORB 特征点跟踪图

Fig. 3　Pictures of ORB feature point tracking

充分考虑了机器人避障、防止视觉 SLAM 在低纹理区域跟踪丢失以及机器人行走效率等多方面因素, 把奖励函数分为 6 个部分。其中, 避障、距离、视觉 SLAM 的奖励函数部分分别用 r_{obv}、r_{dis}、r_{slam} 表示, λ_1、λ_2、λ_3 为相应部分奖励的系数。该 3 部分奖励函数已能够满足点导航的基本需求。然而, 由于算法初期网络训练不充分, 机器人在面对障碍和低纹理区域时会选择错误的动作导致碰撞、跟踪失败。在很多次试错之后, PPO 算法的 Actor 网络才会在面对障碍时选择正确的动作, 此时算法才开始收敛。所以仅依靠上述 3 部分的奖励函数训练局部路径规划算法会存在训练时间长、算法收敛速度慢的缺点。为了加快算法收敛速度, 提高模型性能, 本文又设计了角度、特征点数及到达局部目标点的奖励, 分别使用 r_{angle}、r_{mmpt}、r_{bonus} 表示, λ_4、λ_5、λ_6 为系数。为方便表示, s_{sts} 表示机器人运行状态, 其值域为{0,1,2,3}, 分别表示机器人正常运行、碰撞、视觉 SLAM 跟踪丢失和到达局部目标点。

1) 避障

避障是导航的基本要求, 机器人撞到障碍物就意味着本次导航任务的失败, 机器人运动过程发生碰撞, 会产生一个较大的负值奖励。具体设计如式 (5):

$$r_{\text{asv}} = \begin{cases} \lambda_1, & s_{\text{sts}} = 1 \\ 0, & \text{其他} \end{cases} \tag{5}$$

2) 距离

导航过程中需要逐步减小与局部目标点的距离, 故根据执行当前策略动作后与局部目标点距离的变化量设计相应奖励函数。设当前时刻机器人坐标为 (x_t, y_t), 执行当前策略得出的动作 a_t 后坐标为 (x_{t+1}, y_{t+1}), 局部目标点坐标为 (x_d, y_d)。可以简单在笛卡尔坐标系上计算得到执行动作后, 距目标点的距离变化量为

$$\Delta d = \sqrt{(x_t - x_d)^2 + (y_t - y_d)^2} - \sqrt{(x_{t+1} - x_d)^2 + (y_{t+1} - y_d)^2} \tag{6}$$

对应的奖励函数为

$$r_{\text{dis}} = \lambda_2 \Delta d \tag{7}$$

3）视觉 SLAM 跟踪

在训练过程中，若机器人视觉 SLAM 跟踪稳定运行，执行某一动作后，若视觉 SLAM 跟踪能保持运行，不给予奖励，若执行动作后跟踪丢失，则导航失败，给予一个较大的负值奖励，即如式 (8) 所示：

$$r_{\text{slam}} = \begin{cases} -\lambda_3, & s_{\text{sts}} = 2 \\ 0, & \text{其他} \end{cases} \tag{8}$$

4）角度

正确的导航方向是机器人能到达目标点的前提，为防止机器人在调整方向时出现奖励稀疏的问题，根据机器人动作执行前后其与局部目标点的相对角度的变化量给予奖励。设当前时刻机器人与局部目标点的相对角度为 α_t，执行当前策略得出的动作后变为 α_{t+1}，对应奖励为

$$r_{\text{angle}} = \lambda_4 (|\alpha_t| - |\alpha_{t+1}|) \tag{9}$$

5）特征点数

本文使用的 ORB-SLAM2 算法进行同时定位与建图。对 ORB 特征点的数量与视觉 SLAM 具有很强的关联性，当前帧中的特征点数高于一定数量时，视觉 SLAM 跟踪丢失的风险很低，而低于一定数值时，跟踪丢失的风险将会急剧上升。设执行动作前后，当前帧 ORB 特征点的数量分别为 nmpt_t 和 nmpt_{t+1}，则对应奖励为

$$r_{\text{nmpt}} = \lambda_5 \lg\left(\frac{\text{nmpt}_{t+1}}{\text{nmpt}_t}\right) \tag{10}$$

6）达成目标奖励

为了使局部路径策略更快向局部目标点收敛，在训练过程中，若机器人顺利到达局部目标点，意味着当前策略更加接近目标策略，给予额外的正奖励值，加快算法的训练速度，故设计如式 (11) 所示奖励函数。

$$r_{\text{bonus}} = \begin{cases} \lambda_6, & S_{\text{sts}} = 3 \\ 0, & \text{其他} \end{cases} \tag{11}$$

综上，综合奖励函数为

$$r = r_{\text{obv}} + r_{\text{dis}} + r_{\text{angle}} + r_{\text{nmpt}} + r_{\text{slam}} + r_{\text{bonus}} \tag{12}$$

2.4　PPO 算法

本文采用 PPO 算法来训练局部路径规划算法，这是一种基于策略梯度的算法，采用 Actor-Critic 架构集成了双网络的算法结构，并改进了基于置信域策略优化的强化学习算法[24](trust region policy optimization, TRPO) 的步长选择机制。与 TRPO 算法相比，PPO 算法计算复杂度更低，算法的训练速度更快，可实施性更强。算法的基本框架如图 4 所示。

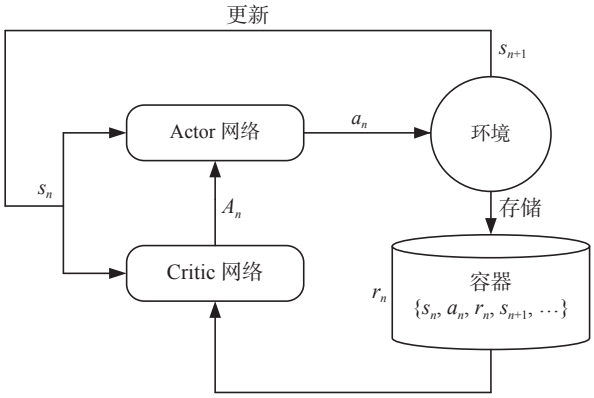


图 4　基于 Actor-Critic 框架的 PPO 算法示意图

Fig. 4　Schematic diagram of PPO algorithm based on Actor-Critic framework

基于 Actor-Critic 框架的 PPO 算法包含 Actor 和 Critic 双网络。Actor 网络负责生成策略，其网络参数为 θ_A。Critic 网络通过计算优势函数 A_n 来评估当前策略，其网络参数为 θ_c。Actor 网络目标函数如式 (13) 所示。

$$L^{\text{clip}}(\theta_A) = E_n\left[r_{\theta_A} A_n, \text{clip}(r_{\theta_A}, 1-\varepsilon, 1+\varepsilon) A_n\right] \tag{13}$$

$$r_{\theta_A} = \frac{\pi_{\theta_A}(a_n \mid s_n)}{\pi_{\theta_{\text{Aold}}}(a_n \mid s_n)} \tag{14}$$

式中：clip 为剪切函数，ε 为剪切参数；ε为 n 次采样的期望函数。$\pi_{\theta_A}(a_n \mid t_n)$是待优化的策略网络，$\pi_{\text{Aold}}(a_n \mid s_n)$为当前用于收集数据的策略网络，通过重要性采样来估计新策略。两者比值越接近 1，说明新旧策略更新偏移越小。更新过程中，PPO 算法利用式 (13) 中的剪切函数来限制策略的更新幅度。当新旧策略更新偏移量过大时，使用剪切项代替，这样确保新旧策略的偏离程度不至于太大，让 Actor 网络以一种相对平稳的方式进行更新，收敛速度更快。

Actor 网络根据当前状态生成机器人当前动作，机器人执行当前动作后产生新状态并获得奖励为一次完整交互过程，按训练批次的大小将多次交互数据进行存储，用于更新 Actor 网络和 Critic 网络，获得相对最优的网络参数。

2.5　网络结构

PPO 算法包含了 Actor 和 Critic 两个神经网络。Actor 网络结构如图 5 所示，整个网络包含了一个 Resnet18 网络[25]，4 个全连接层 (fully connected, FC) 和一个 Softmax 层。PPO 算法的观测状态空间为 $S(s_{\text{rgb}}, s_{\text{depth}}, s_{\text{dis}}, s_{\text{angle}}, s_{\text{mpt}})$，RGB 图像 s_{rgb}、深度图像 s_{depth} 及 ORB 特征点图矩阵 s_{mpt} 整合而成的

640×480×5 的张量, 作为 Resnet18 网络的输入, s_{dis} 和 s_{angle} 分别作为全连接层的输入, 三者均输出一维的向量, 将其输出进行拼接 (Concat), 后接两个全连接层和一个 Softmax 层, 输出 PPO 算法动作空间中的动作。Actor 网络的参数设置如表 1 所示, Critic 网络结构及参数与 Actor 网络大致相同, FC4 层作为输出层, 输出一维数据, 用来估计状态价值函数。

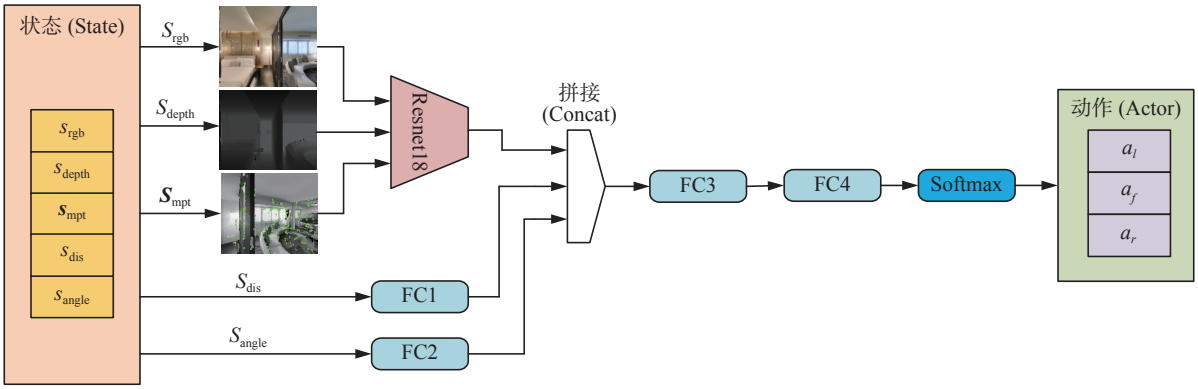


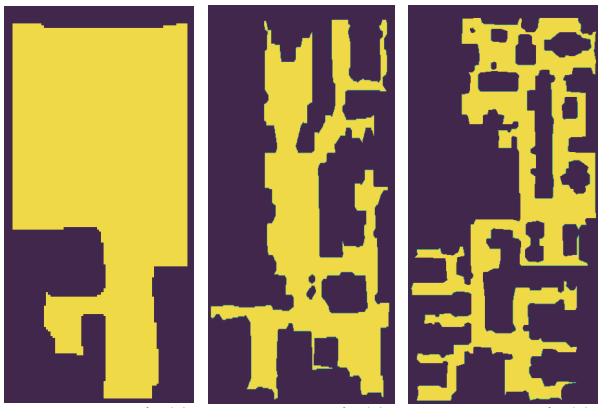
图 5　Actor 网络结构

Fig. 5　Actor network structure

表 1　网络参数设置

Table 1　Network parameters setting

层	输入	Activation	输出
Resnet18	512×512×5	Relu	512
FC1	2	Relu	512
FC2	1	Relu	512
FC3	1 536	Relu	300
FC4	300	Relu	3
Softmax	3	—	3



(a) Edgemere 场景　(b) Eastville 场景　(c) Mosquito 场景

图 6　导航地图可视化

Fig. 6　Navigation map visualization

3　仿真结果与分析

3.1　实验环境及参数设置

仿真所使用硬件平台为一台 CPU 型号为 Intel Core i9-10900X, 内存为 64 GB, 显卡类型为 NVIDIA RTX 3090 的台式机。软件方面装有: Ubuntu 18.04 系统、python 3.6 版本, Pytorch 1.7.1 版本, ROS Melodic 版本。仿真器使用 Facebook 公开的室内仿真平台 Habitat[18]。训练集采用 Gibson 数据集, 包含 72 个不同场景。算法在与训练集不同的 3 个 Gibson 数据集[26] 场景{Edgemere、Eastville、Mosquito}中测试, 每个场景均包含 71 个导航任务。{Edgemere、Eastville、Mosquito}场景中导航任务的平均最短路径距离[26](geodesic distance along the shortest path, GDSP)分别为 3.24 m、7.51 m、10.84 m, 可分别代表简单、中等和困难 3 种场景, 如图 6 所示。可以看到, Edgemere 场景的布局相对最简单, Mosquito 场景的布局相对最复杂。

仿真过程中, 局部路径规划策略的训练相关参数如表 2 所示。本文选择前进动作时, 距离变化为 0.1 m, 选择转向动作时, 角度变化为 0.3 rad。为了使机器人快速接近局部目标点, 需先快速调整航向角, 再向局部目标点快速移动, 可通过设计奖励函数, 使得角度奖励比距离奖励略大。因此, 距离奖励系数和角度奖励系数可根据经验分别设置为 20 和 40。如 2.3 节所述, 本文在设计特征点奖励函数部分时采用以 10 为底数的对数函数。在实验过程中, 可注意到机器人在纹理较丰富的区域时, 图像帧中的特征点数变化幅度不大, 大概在 850~950。而机器人在由纹理较丰富的区域向低纹理区域运动时, 相邻帧之间的特征点变化幅度较大, 比值 $\mathrm{nmpt}_{t+1}/\mathrm{nmpt}_t$ 大概在 1.3~1.8。因此, 特征点奖励系数可根据经验设置为 80。此外, 本文将碰撞和跟踪失败时的奖励值设为−30, 以此降低导致碰撞和跟踪失败的动作选

择概率。为了提高正常到达局部目标点的动作选择概率，本文将顺利到达局部目标点的额外奖励设为 80。

表 2　算法训练参数
Table 2　Algorithm training parameters

实验参数	值
优化器	Adam
折扣因子γ	0.8
批量大小	64
剪切参数ε	0.2
Actor初始学习率	4×10^{-4}
Critic初始学习率	2×10^{-5}
奖励函数系数[λ_1~λ_6]	[30, 20, 40, 80, 30, 80]

3.2　结果及分析

为了证明所述方法的有效性和优越性，对比了 DWA、TEB、MPC 算法和路径跟随 (path follower, PF) 算法。DWA、TEB、MPC 算法是 ROS 系统中默认的局部路径规划算法，可作为很好的比较基准；PF 算法是文献 [27] 中提到的一种路径跟随算法，算法基于 PID 的离散控制实现机器人沿着规划出的全局路径行走。在行走过程中，若机器人没有面朝局部目标点（当前位置与局部目标的朝向角超过 15 °），则执行左转或右转以缩小自身朝向角，否则执行前进动作。当机器人与全局目标点的距离小于 0.2 m 时，则认为一个导航任务成功。

本文计算了相应的成功率 (success rate, SR)、跟踪丢失率 (tracking failure rate, TFR)、碰撞率 (collision rate, CR)。表 3~5 分别给出了使用不同方法在 Edgemere、Eastville 和 Mosquito 3 种场景任务中的平均结果对比。可以得到以下几点结论：

1) PF 算法在所有场景任务中表现都相对较差，这是因为该算法仅仅沿着规划出的全局路径行走，没有考虑如何避障。当全局路径附近存在障碍物时，该算法很难避免发生碰撞。因此，在所有场景的任务中，该算法的平均碰撞率均是最高的。

2) DWA、TEB、MPC 算法运行时需要载入局部地图，根据参数对行走轨迹进行采样，选择运动代价最小的轨迹。因此，DWA、TEB、MPC 算法在进行视觉导航时具有一定的避障能力。但是，DWA、TEB、MPC 算法没有考虑机器人在视觉 SLAM 过程中易在低纹理区域跟踪丢失的问题，无法保证跟踪的稳定性。因此，在 3 个场景的任务中均会发生较高的跟踪丢失率。

3) 所提出的方法在所有的场景任务中都取得了最好的性能，可以在室内杂乱环境中也能较好完成导航任务。具体来说，本文所述方法在所有场景任务中相比于传统的路径规划算法的平均成功率提高了 53.9%，位姿跟踪丢失率减小了 66.5%，碰撞率减小了 30.1%。这说明相比于传统的局部路径规划方法，本文所提出的方法能够实现到达局部目标点的同时，更好地规避障碍物，并保证跟踪稳定性。

4) 在表 3~5 中，本文所提出的方法在 Edgemere 场景任务中表现最好，在 Eastville 场景任务中表现次之，在 Mosquito 场景任务中表现相对较差。这是因为 Mosquito 场景的面积相对更大，包含了多个房间和障碍物，导航任务也相对较难，起点与终点距离较长。而 Edgemere 场景的面积相对较小，仅包含一个卧室和卫生间。

5 类算法导航成功率、碰撞率以及 SLAM 跟踪丢失的概率随导航距离的变化分别如图 7 的 3 种图所示，随着导航距离增加，各类算法碰撞概率和 SLAM 跟踪概率都有所增加，但本文提出算法的导航效果对导航距离更加鲁棒，尤其是 SLAM 跟踪丢失概率受导航距离影响很小。

表 3　不同算法在 Edgemere 场景任务中的平均结果对比
Table 3　Comparison of the average results of different algorithms in the Edgemere scene task

算法	SR	TFR	CR
本文	74.65	1.41	23.94
PF	14.08	9.86	76.06
DWA	55.32	32.39	12.29
TEB	59.15	30.99	9.86
MPC	50.70	36.62	12.68

表 4　不同算法在 Eastville 场景任务中的平均结果对比
Table 4　Comparison of the average results of different algorithms in the Eastville scene task

算法	SR	TFR	CR
本文	56.33	11.27	32.40
PF	5.63	14.08	80.29
DWA	25.35	21.13	53.52
TEB	23.94	26.76	49.30
MPC	21.13	53.52	25.35

表 5　不同算法在 Mosquito 场景任务中的平均结果对比

Table 5　Comparison of the average results of different algorithms in the Mosquito scene task

算法	SR	TFR	CR
本文	45.07	14.08	40.85
PF	2.86	15.49	81.65
DWA	18.31	23.94	57.75
TEB	23.94	28.17	47.89
MPC	23.94	26.76	49.30

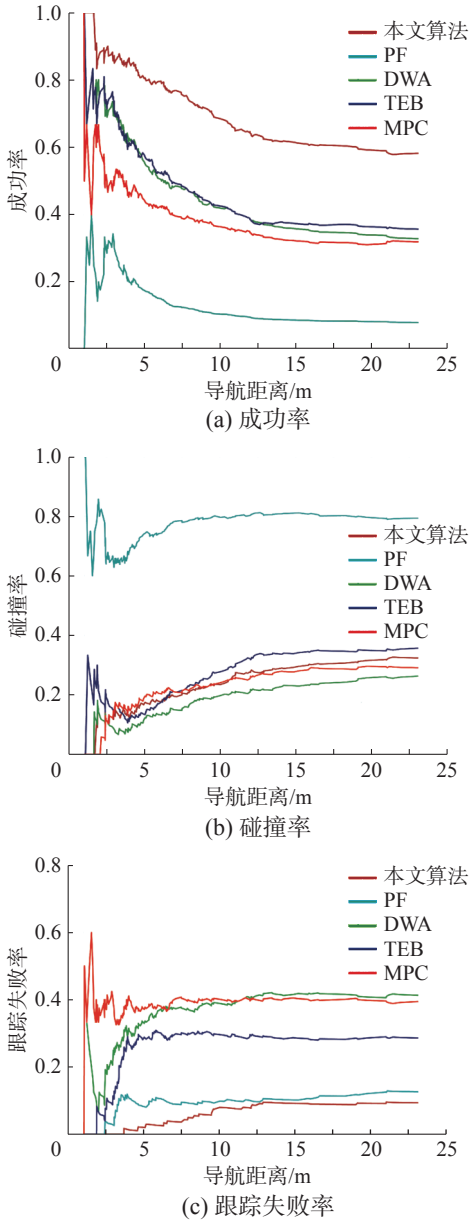


图 7　性能–导航距离变化曲线

Fig. 7　Performance-navigation distance change curves

为了进一步验证所述方法在避免视觉 SLAM 跟踪失败方面的有效性, 以 Edgemere 场景的第 30 个导航任务为例, 将机器人行走过程中的特征点个数进行对比统计。如图 8 所示, PF 方法和 MPC 方法在低纹理区域发生了跟踪丢失现象, 造成视觉 SLAM 失败。而 DWA 方法虽然没有发生跟踪丢失现象, 但在第 5~15 个关键帧之间特征点急剧下降, 有很大的跟踪丢失风险。类似地, TEB 方法在第 30~45 个关键帧之间特征点急剧下降, 也存在较大的跟踪丢失风险。相反, 采用本文所提出的方法后, 机器人在行走过程中所跟踪到的特征始终维持在 580 个以上, 具有较强的稳定性。3 种方法的轨迹对比如图 9 所示(起点在左上角)。绿色轨迹代表本文所提出的方法的行走过程, 红色轨迹代表 DWA 方法的行走过程, 蓝色轨迹代表 PF 方法的行走过程, 青色轨迹代表 TEB 方法的行走过程, 紫色轨迹代表 MPC 方法的行走过程。由于 PF 方法和 MPC 方法在行走的前 15 步时就跟踪失败, 轨迹长度较短。机器人在沿着 DWA 方法规划出的轨迹行走时, 需要近距离, DWA 方法有沿墙行走的习惯, 容易发生碰撞。

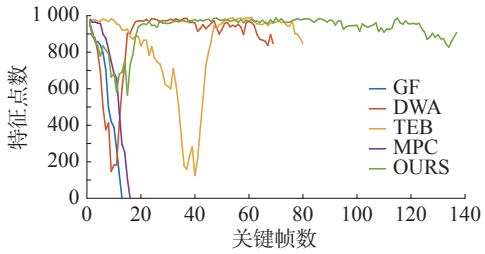


图 8　特征点个数对比（Edgemere 场景的第 30 个导航任务）

Fig. 8　Comparisons of the number of feature points (The 30th navigation mission of the Edgemere scene)

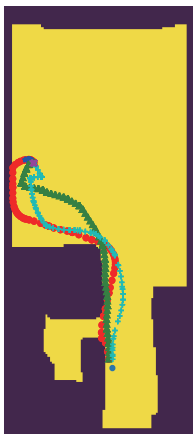


图 9　轨迹对比

Fig. 9　Trajectory comparison

为了更客观地衡量本文奖励函数的设计对于模型性能的影响, 使用消融实验证明其有效性。消融实验是深度强化学习研究中确定某种方法是否有效的最直接的方式。本文在 500 个任务组成的训练集中对所提出的 6 部分奖励函数进行消融实验, 训练过程中每次剔除其中一部分奖励函数以查看对模型指标的影响。评价指标包括成功率、跟踪丢失率和碰撞率。实验结果如图 10 所

示。同时使用 6 部分奖励函数时，模型在点导航问题上具有最好的性能。其他 6 个消融实验在各种指标上都略逊于 6 种奖励函数同时使用的效果，证明了这 6 种奖励函数在点导航任务上具有快速到达局部目标点，并降低碰撞率和跟踪丢失率的能力。另外，在删除角度奖励部分后，模型的性能有较大程度的下降，说明在本文的点导航任务中，角度奖励相比另外 5 种奖励更能提升模型的性能。

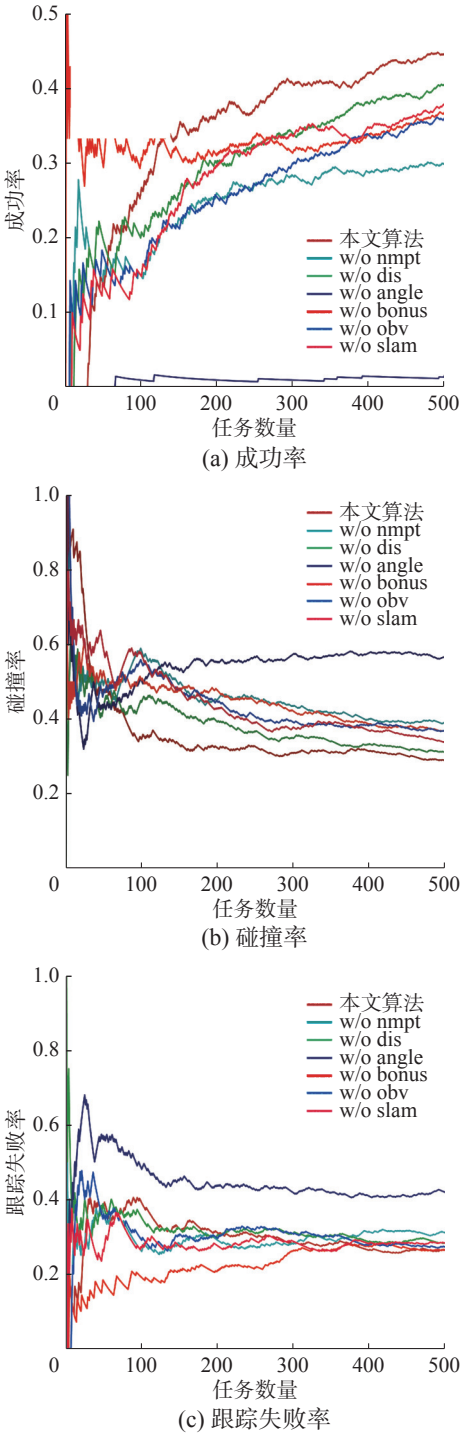


图 10　消融实验结果图

Fig. 10　Results of ablation experiments

4　结论

本文针对传统局部路径规划算法不适用于基于视觉 SLAM 导航的问题，使用深度强化学习算法训练局部路径规划策略，根据输入输出数据特点进行 Actor 网络和 Critic 网络的设计，根据一般导航需求和视觉 SLAM 工作特点设计奖励函数，在 Habitat 仿真平台中使用点导航数据集进行训练，最终得到训练好的策略。在与传统导航策略的对比中，训练好的策略在防止视觉 SLAM 失败避障导航方面都表现出良好的性能，最终在 3 种不同难度场景中的导航成功率，均有巨大提升。下一步计划，考虑将视觉 SLAM 建立的二维地图纳入强化学习的观测空间，使局部导航策略的决策具有记忆性；对特征点分布进行分析，不仅考虑特征点的数量，还要考虑将特征点分布对 SLAM 稳定性的影响。

参考文献：

[1] PANDEY A. Mobile robot navigation and obstacle avoidance techniques: a review[J]. International robotics & automation journal, 2017, 2(3): 96–105.

[2] YASUDA Y D V, MARTINS L E G, CAPPABIANCO F A M. Autonomous visual navigation for mobile robots: a systematic literature review[J]. ACM computing surveys, 2021, 53(1): 13.

[3] FANG Baofu, MEI Gaofei, YUAN Xiaohui, et al. Visual SLAM for robot navigation in healthcare facility[J]. Pattern recognition, 2021, 113: 107822.

[4] YANG Shaowu, SCHERER S A, YI Xiaodong, et al. Multi-camera visual SLAM for autonomous navigation of micro aerial vehicles[J]. Robotics and autonomous systems, 2017, 93: 116–134.

[5] 张瑜, 宋荆洲, 张琪祁. 基于改进动态窗口法的户外清扫机器人局部路径规划 [J]. 机器人, 2020, 42(5): 617–625.
ZHANG Yu, SONG Jingzhou, ZHANG Qiqi. Local path planning of outdoor cleaning robot based on an improved DWA[J]. Robot, 2020, 42(5): 617–625.

[6] 王殿君. 基于改进 A*算法的室内移动机器人路径规划 [J]. 清华大学学报(自然科学版), 2012, 52(8): 1085–1089.
WANG Dianjun. Indoor mobile-robot path planning based on an improved A* algorithm[J]. Journal of tsinghua university (science and technology edition), 2012, 52(8): 1085–1089.

[7] 张飞, 白伟, 乔耀华, 等. 基于改进 D*算法的无人机室内路径规划 [J]. 智能系统学报, 2019, 14(4): 662–669.
ZHANG Fei, BAI Wei, QIAO Yaohua, et al. UAV indoor path planning based on improved D* algorithm[J]. CAAI transactions on intelligent systems, 2019, 14(4): 662–669.

[8] FOX D, BURGARD W, THRUN S. The dynamic window approach to collision avoidance[J]. IEEE robotics & automation magazine, 1997, 4(1): 23–33.

[9] ROESMANN C, FEITEN W, WOESCH T, et al. Trajectory modification considering dynamic constraints of autonomous robots[C]//ROBOTIK 2012; 7th German Conference on Robotics. Munich, VDE, 2012: 1−6.

[10] Rösmann C. Time-optimal nonlinear model predictive control[D]. Dissertation, Technische Universität Dortmund, 2019.

[11] CHEN Chunlin, LI Hanxiong, DONG Daoyi. Hybrid control for robot navigation-A hierarchical Q-learning algorithm[J]. IEEE robotics & automation magazine, 2008, 15(2): 37–47.

[12] VAN HASSELT H, GUEZ A, SILVER D. Deep reinforcement learning with double q-learning[C] // Proceedings of the AAAI Conference on Artificial Intelligence, Arizona, USA 2016: 2094−2100.

[13] 张福海, 李宁, 袁儒鹏, 等. 基于强化学习的机器人路径规划算法 [J]. 华中科技大学学报(自然科学版), 2018, 46(12): 65–70.
ZHANG Fuhai, LI Ning, YUAN Rupeng, et al. Robot path planning algorithm based on reinforcement learning[J]. Journal of Huazhong university of science and technology (natural science edition), 2018, 46(12): 65–70.

[14] GULDENRING R, GÖRNER M, HENDRICH N, et al. Learning local planners for human-aware navigation in indoor environments[C]//2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Las Vegas, IEEE, 2021: 6053−6060.

[15] BALAKRISHNAN K, CHAKRAVARTY P, SHRIVASTAVA S. An A* curriculum approach to reinforcement learning for RGBD indoor robot navigation[EB/OL]. (2021−01−01)[2021−12−12]. https://arxiv. org/abs/2101.01774.

[16] CHAPLOT D S, GANDHI D, GUPTA S, et al. Learning to explore using active neural slam[C]// 2020 International Conference on Learning Representations (ICLR), Addis Ababa, 2020.

[17] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal policy optimization algorithms[EB/OL]. (2017−08−28)[2020−12−12]. https://arxiv.org/abs/1707.06347.

[18] SAVVA M, KADIAN A, MAKSYMETS O, et al. Habitat: a platform for embodied AI research[C]//2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul, IEEE, 2019: 9338−9346.

[19] 林志林, 张国良, 王蜂, 等. 一种基于 VSLAM 的室内导航地图制备方法 [J]. 电光与控制, 2018, 25(1): 98–103.
LIN Zhilin, ZHANG Guoliang, WANG Feng, et al. A method for indoor navigation mapping based on VSLAM[J]. Electronics optics & control, 2018, 25(1): 98–103.

[20] 马跃龙, 曹雪峰, 万刚, 等. 一种基于深度相机的机器人室内导航点云地图生成方法 [J]. 测绘工程, 2018, 27(3): 6–10,15.
MA Yuelong, CAO Xuefeng, WAN Gang, et al. A method of generating point cloud maps for indoor auto-navigation of robots based on depth camera[J]. Engineering of surveying and mapping, 2018, 27(3): 6–10,15.

[21] 张毅, 陈起, 罗元. 室内环境下移动机器人三维视觉 SLAM[J]. 智能系统学报, 2015, 10(4): 615–619.
ZHANG Yi, CHEN Qi, LUO Yuan. Three dimensional visual SLAM for mobile robots in indoor environments[J]. CAAI transactions on intelligent systems, 2015, 10(4): 615–619.

[22] RUBLEE E, RABAUD V, KONOLIGE K, et al. ORB: an efficient alternative to SIFT or SURF[C]//2011 International Conference on Computer Vision. Barcelona, IEEE, 2011: 2564−2571.

[23] MUR-ARTAL R, TARDÓS J D. ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras[J]. IEEE transactions on robotics, 2017, 33(5): 1255–1262.

[24] SCHULMAN J, LEVINE S, ABBEEL P, et al. Trust region policy optimization[C]//International Conference on Machine Learning(PMLR), Lille, 2015: 1889−1897.

[25] HE Kaiming, ZHANG Xiangyu, REN Shaoqing, et al. Deep residual learning for image recognition[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, IEEE, 2016: 770−778.

[26] XIA Fei, ZAMIR A R, HE Zhiyang, et al. Gibson env: real-world perception for embodied agents[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern

Recognition. Salt Lake City, IEEE, 2018: 9068−9079.

[27] MISHKIN D, DOSOVITSKIY A, KOLTUN V. Benchmarking classic and learned navigation in complex 3d environments[EB/OL]. (2019−03−28)[2021−04−05]. https://arxiv.org/abs/1901.10915.

作者简介：

朱少凯，硕士研究生，主要研究方向为基于视觉的同时定位与建图、机器人视觉导航。

孟庆浩，教授，博士生导师，主要研究方向为机器人感知、导航与控制。完成科研项目 10 余项。发表学术论文百余篇。

金晟，博士研究生，主要研究方向为机器人视觉导航、深度强化学习。

2022 年第八届 IEEE 云计算与智能系统国际会议
The 8th IEEE International Conference on Cloud Computing and Intelligence Systems 2022

2022 年第八届 IEEE 云计算与智能系统国际会议（以下简称“CCIS 2022”）由中国人工智能学会和 IEEE 北京分会联合主办，西南交通大学、四川大学、CAAI 智能服务专委会、CAAI 会员服务工委会联合承办，成都市科协协办，四川省人工智能学会支持，将于 11 月 26–28 日在成都举办。

IEEE 云计算与智能系统国际会议由中国人工智能学会联合国际电气与电子工程师协会（IEEE）北京分会发起，已陆续在北京、杭州、深圳、香港、南京、新加坡和西安成功举办七届，以高规格、高水平的特色深受国际同行关注，形成了“小而美”的会议风格。该会议旨在对云计算、人工智能的前沿技术和热点问题进行深入研究和探讨，以促进相关技术和产业的发展。

CCIS 2022 现正面向全球征集稿件，经会议审稿后录用的稿件将由 IEEE 出版，符合 IEEE 标准的会议论文可纳入 IEEE Xplore 数字图书馆。优秀论文将推荐至大会合作的 CAAI Transactions on Intelligence Technology,World Wide Web Journal 等 SCI 期刊。

投稿要求：

（一）论文未曾在国内外杂志或会议上发表；

（二）稿件写作必须使用英文，并严格按照模板要求进行排版（模板点击“阅读全文”下载）；

（三）所有论文采用网上投稿，投稿系统网址为：https://ccis2022.casconf.cn/。

出版检索：

本次会议征集的所有文章将由程序委员会严格审核，所有录用论文将收录至会议论文集，并提交 IEEE Xplore 在线数据库检索，挑选出的优秀论文扩充内容后可推荐至检索期刊。

时间节点：

论文投稿截止日期：2022 年 9 月 10 日

论文录用通知日期：2022 年 10 月 10 日

会议注册/终稿提交截止日期：2022 年 10 月 20 日

会议召开日期：2022 年 11 月 26 日–28 日