arXiv:2601.01067v1 [cs.RO] 3 Jan 2026

基于AnyLoc的单目相机拓扑建图与导航

张文正[1], 原义隆[2],和中村宗助[3]

摘要—本文提出了一种使用单目相机进行拓扑建图与导航的方法。该方法基于AnyLoc将关键帧转换为描述子，以构建拓扑关系，实现回环检测和地图构建。与度量地图不同，拓扑地图通过使用关键节点而非精确坐标表示环境，从而简化路径规划与导航。通过比较分段图像与目标节点关联的图像，确定视觉导航的动作。该系统仅依赖单目相机，利用关键节点实现快速地图构建与导航。实验表明，在无需预训练的情况下，该方法能够在真实环境和仿真环境中有效进行回环检测与导航。与基于ResNet的方法相比，该方法的成功率平均提高了60.2%，同时降低了时间和空间成本，为各种场景下的机器人和人类导航提供了一种轻量级解决方案。

I. 引言

在移动机器人学和自动驾驶领域，地图构建对于环境理解与导航至关重要。地图能够帮助机器人进行定位、规划路径并高效执行任务。地图通常分为度量地图和拓扑地图。

度量地图通过使用特定的空间坐标对环境进行建模，提供高精度的空间信息，因此适用于精确定位和路径规划 [1]。PTAM[2],、ORB-SLAM [3], 、LSD-SLAM [4], 和 DSO [5]等方法利用视觉里程计技术进行精确的位姿估计和环境表示，显著推动了同步定位与建图的发展。然而，这些方法需要大量计算资源，并且在动态或未知环境中表现欠佳。此外，度量地图的构建与维护需要进行大量计算，因此在大规模或动态场景中实用性较低。

另一方面，拓扑地图通过描述位置之间的连通性来表示环境，降低了建模复杂度，使其非常适合大规模动态环境。人类通常也会以类似的方式进行导航，关注关键位置和动作，而非精确距离 [6]。尽管具有这些优势，但由于难以在动态或未知环境中有效表示节点、弧和路径，拓扑地图的构建与导航仍然面临挑战。

为应对这些挑战，我们提出了一种基于视觉定位框架 AnyLoc [7] 的拓扑建图与导航方法。AnyLoc 是一种通用视觉地点识别方法，无需重新训练或微调，即可在多种环境中有效运行。在本研究中，我们利用了 AnyLoc 识别图像相似度的能力。此外，借助其强大的泛化能力，我们的方法无需依赖度量信息，即可高效地构建地图并进行导航。该系统包含两个主要部分：（1）建图，利用 AnyLoc 的视觉相似度生成节点和弧；（2）导航，根据观测图像和构建的地图定位目标并规划路径。该方法仅使用单目相机，在仿真环境和真实环境中均展现出稳健的性能。

我们的贡献如下：

- 无需依赖度量信息：我们的方法仅使用单目相机，即可实现纯拓扑建图与导航。
- 增强的泛化能力和成功率：与传统的基于 ResNet 的方法相比，我们的系统展现出显著提升的泛化能力，平均成功率提高约 60.2%。
- 通过一次完整遍历即可立即部署：我们的方法能够在未知环境中快速部署并进行导航。只需一段单目视频记录，即可构建地图并在其中导航，且系统能够快速适应环境变化。

II. 相关工作

Choset 和 Nagatani 开创了拓扑同步定位与建图的概念，展示了无需依赖明确的几何位置测定，也能实现精确定位 [8]。在此基础上，后续研究整合了视觉数据，以增强拓扑

*本研究未获得任何组织的支持[1]张文正就读于日本东京的法政大学科学与工程研究生院。wenzheng.zhang.5u@stu.hosei.ac.jp[2]原义隆任职于千叶工业大学未来机器人技术研究中心（fuRo），所在地为日本千叶。[3]中村宗助任职于日本东京的法政大学理工学部。

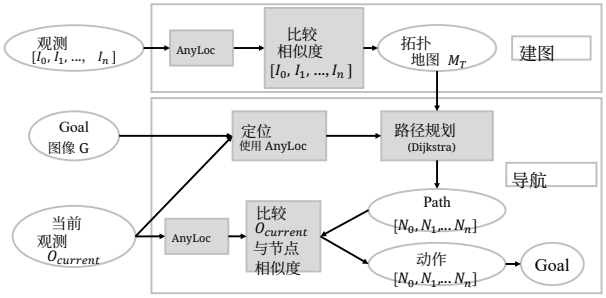

图1. 所提方法的流程。从拓扑地图构建到目标导航。主要包含两个组成部分：(1) 建图和(2) 导航。

建图与导航。例如，Fraundorfer等人提出了基于图像的拓扑建图方法，通过提取视觉特征来建立环境节点，从而在不同环境中构建适应性强的拓扑地图 [9]。

近年来，深度学习的进展显著推动了拓扑SLAM的发展。基于图的学习方法（如图定位网络）提升了拓扑定位的效率 [10]。神经拓扑SLAM进一步将神经网络与视觉感知相结合，以优化地图构建和导航 [11]。Savinov等人提出了一种结合参数化与非参数化表示的记忆模块，使拓扑结构能够高效、动态地更新 [12]。这种方法不仅推动了拓扑导航领域的技术发展，也成为该领域的基准。Meng等人提出了一种分层导航框架，将局部运动控制与全局拓扑结构相结合，实现了高效的大规模导航 [13]。ViNG和PlaceNav等方法探索了动态环境中的自适应导航，将场景识别与路径规划相结合 [14], [15]。Dhruv Shah等人提出了通用导航模型，推动机器人导航朝着泛化和自主探索的方向发展。GNM利用大规模多机器人数据进行训练，实现了跨平台零样本迁移 [16]。ViNT采用基于Transformer的架构，增强基于视觉的导航的泛化能力，并展现出在不同任务和环境中的稳定性能 [17]。NoMaD引入扩散模型，以提高未知环境中的探索效率和目标导向导航能力 [18]。这些方法共同推动机器人导航从专用系统向通用基础模型演进，为未来自主机器人的发展奠定了基础。

这些方法展现了拓扑导航的潜力，但该领域仍面临挑战：(1) 如何快速构建通用地图以应用于未知环境；(2) 对大规模预训练或特定环境数据的依赖。我们提出的方法利用 AnyLoc 的泛化能力，在模拟环境和真实环境中实现高效、稳健的拓扑建图与导航，从而应对这些挑战。

III. 建图与导航

A. 概述

我们研究基于拓扑地图的目标导向视觉导航问题：给定一系列观测图像（I_0，I_1，……，I_n），构建一张拓扑地图M_T。给定目标图像 G 和当前观测 O_{current}，机器人确定自身位置和目标位置 p_0。利用 Dijkstra 算法，在拓扑地图上选取节点序列（N_1，N_2，……，N_n）。导航过程中，通过比较观测图像（a_0，a_1，……，a_n）与关联节点的图像（O_1，O_2，……，O_n），推导出一系列动作（N_1，N_2，……，N_n），使机器人能够到达目标 G，如图 1 所示。

该方法集成了视觉定位技术 AnyLoc。借助 AnyLoc 的图像检索与匹配能力，可以确定关键节点并建立其间的关系，从而构建拓扑地图。AnyLoc 还能辅助机器人在地图中进行定位和动作选择，实现导航。

该方法包含两个主要组成部分：（1）建图；（2）使用构建的拓扑地图进行导航。以下各节将详细介绍每个组成部分。

B. 建图

首先，我们定义拓扑地图的概念。拓扑地图表示为一个由节点 N_i 和弧 L_i组成的抽象有向图。图 2 展示了节点和弧所包含的信息。每个节点 N_i 包含以下属性：数据库中对应图像的$indexvalue$以及可到达节点 N_{i+1}的 $indexvalue$，以及相应的相对距离 $weight$。弧的方向表示相邻节点中从 $indexvalue$到 $indexvalue$的连接。

1）相似度比较：我们使用 AnyLoc 比较不同图像对之间的相似度。AnyLoc 是一种基于全局描述符的视觉特征提取方法，专为视觉定位任务设计。由于能够稳健、高效地处理多样且复杂的视觉数据，它尤其适用于各种环境中的图像相似度任务 [7]。具体而言，AnyLoc 使用 DINOV2 骨干网络提取多尺度视觉特征[19]。这些特征通过 VLAD（局部聚合描述符

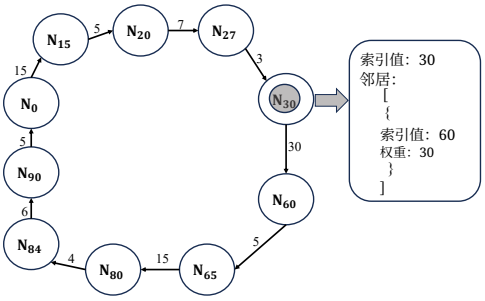

图2. 节点信息。左侧显示简化的拓扑地图，右侧展示每个节点包含的信息。

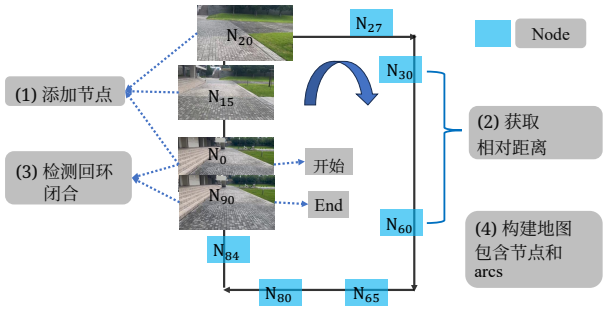

图3. 拓扑建图流程。从 N_0 开始，沿顺时针方向移动至 N_{90} ，图中描述了节点添加、距离计算和闭环检测的过程。

）进行聚合，以生成紧凑的全局描述符。随后，通过计算图像描述符之间的余弦相似度，得出图像间的相似度分数 s 。AnyLoc 的优势在于能够有效应对复杂环境中的光照和视角变化等挑战，为图像相似度比较任务提供稳健的基准。

2) 节点添加：为了构建地图，首先必须添加节点，节点添加策略如图3所示。在添加节点之前，AnyLoc 会处理采集到的图像数据 D_i （其中 i对应于将视频分割为图像后的帧编号），以生成全局描述子。随后，这些描述子会存储在相应的节点 N_i 中。添加节点时有两种情况。

情况1：由于距离增加：机器人从 N_0 出发，朝 N_{15} 前进时，会经过从 N_2 到 N_{14} 的中间节点。对于每个中间节点，都会计算其全局描述符与 N_0 之间的余弦相似度，得到相似度分数 s_i。设置阈值 $T_{\text{add 新节点}}$，用于判断是否应添加新节点。随着距离增加， N_0 与后续节点之间的相似度逐渐降低。当 N_{15} 处的相似度分数低于 $T_{\text{add 新节点}}$时，就会添加新节点 N_{15} 到拓扑地图 M_T中。节点添加算法的正式表达式见公式 1。

$$M_T \leftarrow \begin{cases} M_T \cup \{N_i\}, & \text{if } s_i < T_{\text{add_new_node}}, \\ M_T, & \text{if } s_i \geq T_{\text{add_new_node}}. \end{cases} \tag{1}$$

情况2：由于角度变化显著：从N_{15} 到 N_{20}，相似度分数因方向变化而逐渐降低。因此， N_{20} 会被添加到拓扑地图 M_T中。类似地，在情况2中，节点 N_{27} 到N_{30}、 N_{60} 到 N_{65}以及 N_{80} 到 N_{84} 也会在类似条件下添加。这种方法可在发生显著变化时自动向 M_T 添加节点。

3) 相对距离：在节点添加过程中，会计算相对距离的粗略估计值 M_{distance}。添加新节点 N_i 时，M_{distance}会初始化为0。定义阈值 $T_{\text{add distance}}$，只有当前帧 D_i 与上一帧 D_{i-1} 之间的相似度分数低于该阈值时，距离计数器才会递增。这样，即使帧索引不断增加，也能有效避免机器人静止时距离计数器递增。距离算法的形式化表达见式2。

$$M_{\text{distance}} = \begin{cases} 0, & \text{added newnode}(D_i), \\ M_{\text{distance}} + 1, & \text{similarity}(D_i, D_{\text{previous}}) \\ & \quad < T_{\text{add_distance}}, \\ M_{\text{distance}}, & \text{otherwise}. \end{cases} \tag{2}$$

4) 闭环检测：系统运行期间，会将每个节点 N_i 与拓扑地图 M_T中的所有节点进行相似度比较。定义$T_{\text{loop 闭合}}$以判断是否发生了回环闭合。如果相似度分数 s_i 超过 $T_{\text{loop closure}}$，则检测到回环闭合。

如图3所示，当机器人从 N_{84}到达 N_{90}时，通常会因满足情况1中的条件而添加新节点 N_{90} 。然而，由于 s_{90} 在N_{90} 与现有节点 N_0 之间超过了 $T_{\text{loop closure}}$，因此不会将 N_{90}添加到 M_T。相反，N_{84} 会直接连接到 N_0，并相应更新相对距离 M_{distance} 。此过程有效避免向 M_T添加冗余节点，从而确保拓扑地图更加紧凑、高效。

5) 地图优化： T_{interval} 用于确定添加节点与前一节点之间的最短间隔。通过调整 T_{interval}、$T_{\text{add node}}$和$T_{\text{loop closure}}$，可以控制地图的稀疏度，并移除不必要的节点以优化地图。

如图4所示，在情况（a）中，增加 T_{interval}或减少 $T_{\text{add node}}$ 会使拓扑地图 M_T变得更加稀疏。在情况（b）中，节点 N_1 和 N_4 相似，而 N_4 对应于一个已经由

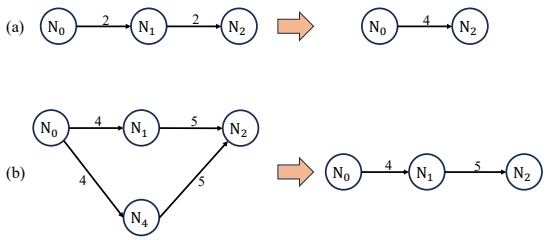

图4. 拓扑地图节点优化。(a)中，N_1 同时靠近 N_0 和 N_2。(b)中，节点 N_1 和 N_4 相似。图像右侧显示优化结果。

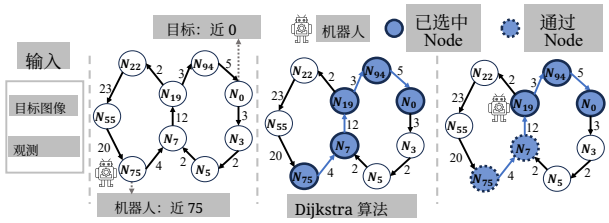

图5. 全局导航系统概览。(a) 在图中定位节点 N_{75}和 N_0，分别对应当前观测和目标图像。(b) 选取图中连接这两个节点的最短路径。(c) 朝目标移动的过程中执行节点选择。

N_1表示的位置，因此不应将 N_4 添加到 M_T中。通过减少$T_{\text{loop closure}}$，可以将 N_4 识别为回环闭合，并与 N_1 合并。随后更新弧，得到图4右侧所示的优化地图。

C. 导航

$Navigation$ 由四个组件构成。图5展示了全局层级中的三个组件，另一个组件则位于局部层级 $Action\ Control$。输入为目标图像 G 和机器人的当前观测 O_{current}。该流程始于 $Localization$，它用于确定目标和机器人的位置。接下来，$Path\ Planning$使用Dijkstra算法生成导航序列 L_0、L_1、……、L_n。$Node\ Selection$ 和 $Action\ Control$ 组件确保局部控制以及沿节点依次导航，并由 $AnyLoc$ 协助完成 $Localization$。此流程使机器人能够到达 G。

1）定位：定位对整个系统至关重要，因为它是导航过程三个关键组成部分的基础。系统设置了阈值 $T_{\text{milestone}}$，以确保定位性能。系统中有两种定位方式。初始定位：使用 AnyLoc 为输入 G 和O_{current}生成全局描述符。利用余弦相似度，将这些描述符与 M_T 中存储的每个节点 N_i 的描述符进行比较，从而生成一系列相似度分数 s_i。如果 s_i 超过$T_{\text{milestone}}$，则选择具有最高 s_i 的节点 N_i 作为定位节点。如果未找到这样的 s_i，机器人会逆时针旋转并再次尝试定位，直到获得超过 $T_{\text{milestone}}$ 的相似度分数。如图 5（a）所示，机器人在 N_{75}附近定位，而目标位于 N_0附近。

常规定位：机器人通过为当前观测 O_{current} 生成全局描述符并重复这一过程，在导航过程中动态跟踪自身位置。

2) 路径规划：以初始化期间获得的两个节点作为输入，使用 Dijkstra 算法，根据 M_T 中节点与弧之间的关系生成导航序列L_0，L_1，……，L_n。该序列对应于选定的节点N_i，如图 5(b) 所示。后续导航通过依次切换导航序列中的节点来进行。

3) 节点选择：使用前文所述的定位方法，机器人可以识别出最接近当前导航序列的节点。只有当相似度分数 s_i 超过 $T_{\text{milestone}}$时，才能确认机器人位于特定节点附近。此外，还会设置节点切换阈值 $T_{\text{change node}}$。当机器人从当前节点 L_i 向下一个节点L_{i+1}移动时，L_i 与 O_{current}之间的相似度分数 s_i 会逐渐降低。当 s_i 低于 $T_{\text{change node}}$时，机器人会切换到 L_{i+1}，并获得新的相似度分数s_{i+1}。如果 $\bar{s}_{i+1}$ 超过 $T_{\text{milestone}}$，且 i 小于导航序列的长度减一，则认为机器人已成功到达 L_{i+1}。当 i等于导航序列的长度减一时，则认为机器人已到达目标节点。为避免因匹配导航序列中过于靠前或靠后的节点而导致错配，会在 $Localization$ 过程中设置边界，以限制可匹配节点的范围。这些边界确保机器人只能匹配当前节点附近的节点。

4）局部运动控制：为了在节点之间移动，机器人根据当前观测 O_{current}执行运动控制，如图 6 所示。图像被分为三部分——$Left$、$Middle$和 $Right$——并使用 AnyLoc 与 $Node\ Image$ 进行比较，以生成相似度分数。如果最高相似度s_{highest} 超过阈值$T_{\text{limited control}}$，机器人就会向前、向左或向右移动，分别对应 $\bar{M}iddle$、$Left$或 $Right$分段。为确保准确性，该过程分两个周期进行：第一周期存储动作；第二周期中，如果两个周期选择的动作相同，则执行该动作。如果出现冲突，除非某个周期选择了 $Middle$，否则不采取任何动作。在这种情况下，运动会将向前移动与左转或右转结合起来，根据需要向前左转或向前右转。这种方法确保了控制更加平稳可靠。

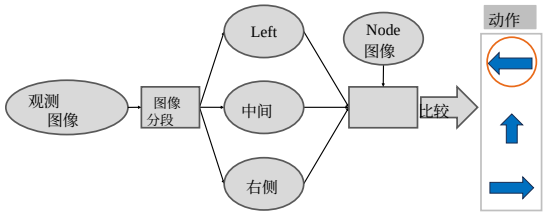

图6. 动作控制。给定一幅观测图像，将其划分为三个部分，以生成三个描述子，并将每个描述子与节点描述符进行比较。随后，根据比较结果选择最合适的动作。

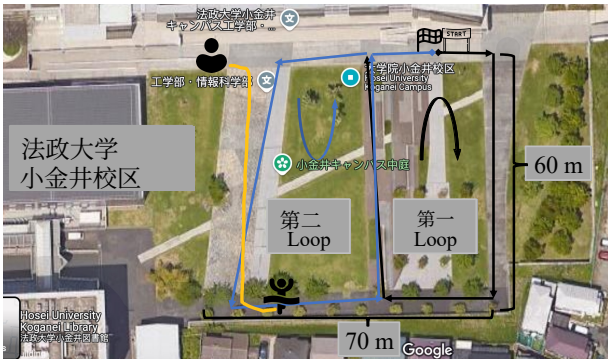

图7. 真实环境。黑色和蓝色线条表示数据采集路线，橙色线条表示室外导航测试路线。

IV. 实验

A. 概述

我们在真实环境和仿真环境中进行了实验，并将我们的方法与以往的工作进行了比较。本研究探讨以下问题：**Q1**：无需预训练，能否在不同环境中有效地实现建图和导航?

Q2：与先前的基线方法相比，所提出的方法表现如何?

Q3：拓扑地图稀疏度对拓扑导航有何影响?

B. 实验环境

1) 真实世界环境：实验在法政大学小金井校区的庭院中进行，场地长约 70 米、宽约 60 米，如图 7 所示。该实验环境包含人行道、绿地和多种植被，行人偶尔会从中经过。

2）仿真环境：本研究使用 NVIDIA Isaac Sim 平台进行仿真实验[20]。Isaac Sim 是一款基于 NVIDIA Omniverse 的高保真机器人仿真工具，支持物理仿真、相机传感器建模以及与 ROS/ROS2 集成。我们使用了两个由 Isaac Sim 提供的仿真环境

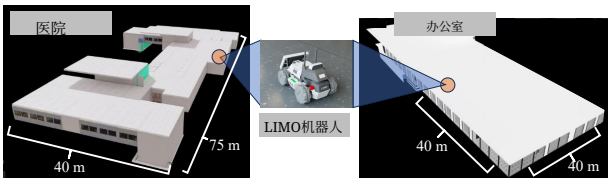

图 8. 仿真环境。(a) 和 (b) 分别展示了医院和办公室环境的外观及布局，并标出了机器人位置的大致位置。

。图 8(a) 中的实验环境代表医院环境。此外，为了评估我们方法的泛化能力，我们还在仿真办公室环境中进行了实验，如图 8(b) 所示。该环境比医院环境明亮许多。在仿真过程中，Isaac Sim 的 Omniverse Kit 界面与 ROS2 集成，实现了实时控制和数据采集。所用机器人模型为 Limo，配备了一台水平视场角（HFOV）为 65° 的单目相机，用于数据采集和视觉导航。

C. 实验方法

所有数据处理和算法执行均在一台配备 NVIDIA GeForce GTX4070 Ti SUPER GPU 和 Intel i7-13700 CPU 的台式电脑上完成。

1) 真实世界环境：数据采集：使用 iPhone 13 Pro Max 在实验区域采集数据，录制分辨率为 1080p、帧率为每秒 30 帧的视频。如图 7 所示，数据采集从 $start$ 位置 开始，沿黑线顺时针前行，再沿蓝线逆时针前行。黑线与蓝线的重叠部分表示发生了回环闭合。路线最终在目标旗帜附近结束。采集的视频在台式电脑上处理，每隔 30 帧保存一帧。

地图构建：使用第 III-B 节所述方法构建拓扑地图。通过调整参数控制地图的稀疏度，生成的地图如图 9 所示。

导航实验：我们在不同时间进行了实验，以评估该方法的泛化能力。从图 7 左上角的小人形图标处出发，一名人员沿黄色轨迹行进，最终停在下方的人形图标处。实验将一张目标图像作为输入，并回放采集的视频，以测试算法能否准确识别当前节点、将目标图像与其对应节点匹配、生成正确的导航序列，并引导机器人朝正确方向移动。实验观察机器人是否能够

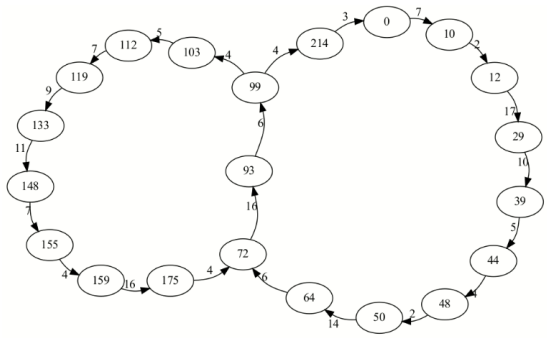

图 9. 真实环境的拓扑地图。

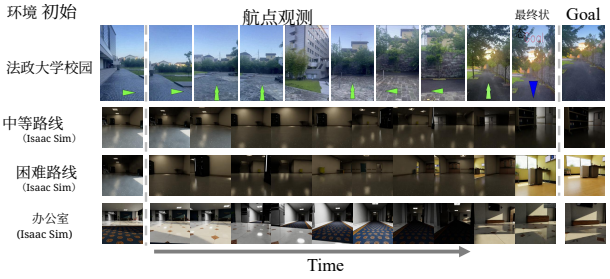

图 10. 泛化实验：我们在不同环境中评估模型。第一列表示实验环境。给定一张目标图像（最后一列），机器人或人员尝试导航至目标。中间列展示来自访问节点的观测图像。倒数第二列展示最终观测，表明机器人成功到达目标；可将其与最后一列中的目标图像进行比较。

准确到达目标并识别目标。

2）仿真环境：数据采集：通过沿预定义路径操控机器人在实验区域内导航来采集数据。在机器人移动过程中，使用 ROS2 捕获图像。

地图构建：地图创建过程与真实世界环境中的过程相同。

导航实验：将目标图像或节点索引作为输入，测试算法能否沿拓扑地图上的规划路线操控机器人导航，并成功实现拓扑导航。

D. 评估方法

我们在仿真中评估了所提方法，随后[21], 将其 $success\ rate$ （SR）与基线 SPTM [12]进行比较。SR 是在一条测试路线上的成功尝试次数与总尝试次数之比，用于表示机器人是否到达目标。该指标对于评估导航性能至关重要。

我们在医院环境中评估了不同长度的路线——10米、20米和35米——并根据转弯次数和障碍物数量，将路线分为三个难度等级： $easy$、 $moderate$和$hard$。此外，还在办公环境中设置了一条长度约为40米的路线

表 1 不同方法和难度等级下的成功率比较

方法	Easy 10 m	中等 20 m	Hard 35 m	办公室 40 m	平均值
ResNet+AnyLoc	1.00	0.64	0.47	0.25	0.59
Ours	1.00	0.92	0.86	1.00	0.95

，并将其难度归类为中等。该评估框架使我们能够与基线方法进行全面比较，突出所提方法的优势与局限性。遵循 SPTM 方法，通过在环境中进行探索来收集数据，并利用迁移学习使用 ResNet18 模型训练定制的动作预测模型，从而增强其竞争力。此外，在办公环境中，基于 ResNet 的方法采用在医院环境中训练的模型来评估其泛化能力。为确保比较结果的准确性，两种方法均使用 AnyLoc 进行定位。为实现公平且一致的评估，每个模型在每条路线上均测试了14次。

E. 结果

如图 10 所示，我们的方法在真实和虚拟环境中，无论室内还是室外，都能以最少的节点成功实现导航，且无需预训练。这解决了 $Q1$。

此外，如表 1 所示，我们的方法在所有测试路线中均持续优于基线方法，而且随着路线难度和长度的增加，这一优势愈发明显。同时，在办公环境中开展的评估进一步验证了模型的泛化能力：性能提升了 75%，总体平均成功率提高了 60.2%，从而明确回应了 $Q2$。

为解决 $Q3$，我们使用稀疏度各异的地图开展了导航实验，如图11所示。当地图中的节点彼此过于接近时，导航轨迹会更贴近原始路径，但也更容易发生重新定位。相比之下，较稀疏的地图能显著减少重新定位的发生。这是因为过于密集的地图会增加机器人在向前运动时错过导航节点的可能性，导致频繁尝试重新定位。因此，拓扑地图中的节点越密集，结果不一定越好。适当的稀疏度对于有效的拓扑导航至关重要。

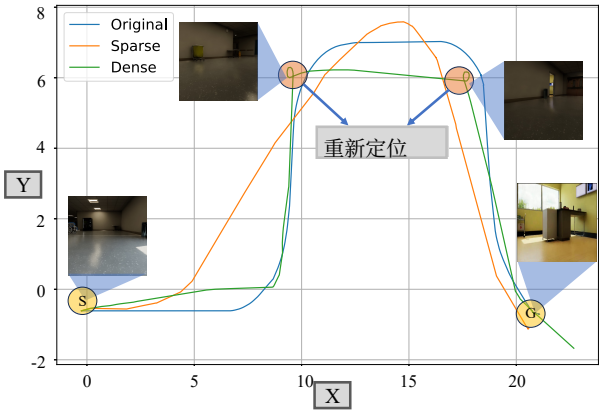

图11. 拓扑地图稀疏度对导航的影响。S 为起点，G 为目标点。蓝线表示机器人在数据采集期间的原始轨迹，橙线和绿线分别表示拓扑地图稀疏和稠密时的轨迹。

V. 结论

本文介绍了一种系统，可在真实环境和仿真环境中使用单目相机进行拓扑建图和目标导向导航。通过集成 AnyLoc，该系统根据图像相似度的变化高效地选择关键节点，从而快速构建拓扑地图。在视觉导航过程中，AnyLoc 被用于节点定位和动作选择。通过大量实验，我们回答了三个关键研究问题，并证明我们的方法在多种路线上的表现优于基线方法。值得注意的是，我们的系统无需预训练，即可在新环境中快速完成地图构建和视觉导航。

当前的局限性包括地图构建和导航过程中需要手动调整参数。目前，我们已在解决参数调整问题方面取得进展，计划未来完成集成工作。此外，我们希望利用观测图像中的更多信息，进一步提升导航过程的鲁棒性和可靠性。

参考文献

[1] S. Thrun、W. Burgard 和 D. Fox，《概率机器人学》。麻省理工学院出版社，2005 年。[2] G. Klein 和 D. Murray，“适用于小型增强现实工作空间的并行跟踪与建图”，载于《IEEE与ACM混合现实和增强现实国际研讨会（ISMAR）》，2007 年。[3] R. Mur-Artal、J. M. M. Montiel 和 J. D. Tard´os，“ORB-SLAM：一种灵活且准确的单目SLAM系统”，《IEEE机器人学汇刊》，第 31 卷，第 5 期，第 1147–1163 页，2015 年。[4] J. Engel、T. Sch¨ops 和 D. Cremers，“LSD-SLAM：大规模直接单目SLAM”，载于《欧洲计算机视觉会议（ECCV）》，2014 年。[5] J. Engel、V. Koltun 和 D. Cremers，“直接稀疏里程计”，《IEEE模式分析与机器智能汇刊》，第 40 卷，第 3 期，第 611–625 页，2018 年。

[6] S. Gillner 和 H. A. Mallot，“虚拟迷宫中的导航与空间知识获取”，《认知神经科学杂志》，第 10 卷，第 4 期，第 445–463 页，1998 年。[7] N. Keetha、A. Mishra、J. Karhade、K. M. Jatavallabhula、S. Scherer、M. Krishna 和 S. Garg，“AnyLoc：迈向通用视觉地点识别”，《IEEE机器人与自动化快报》，第 9 卷，第 2 期，第 1286–1293 页，2024 年。[8] H. Choset 和 K. Nagatani，“拓扑同步定位与建图（SLAM）：在无需显式定位的情况下实现精确定位”，《IEEE机器人学与自动化汇刊》，第 17 卷，第 2 期，第 125–137 页，2001 年。[9] F. Fraundorfer、C. Engels 和 D. Nister，“使用图像集合进行拓扑建图、定位与导航”，载于《IEEE/RSJ智能机器人与系统国际会议（IROS）》，2007 年。[10] K. Chen、J. P. de Vicente、G. Sepulveda、F. Xia、A. Soto、M. Vazquez 和 S. Savarese，“利用图定位网络实现视觉导航的行为方法”，载于《机器人学：科学与系统（RSS）》，2019 年。[11] D. Singh Chaplot、R. Salakhutdinov、A. Gupta 和 S. Gupta，“用于视觉导航的神经拓扑SLAM”，载于《IEEE/CVF计算机视觉与模式识别会议（CVPR）》，2020 年。[12] N. Savinov、A. Dosovitskiy 和 V. Koltun，“用于导航的半参数拓扑记忆”，载于《国际学习表征会议（ICLR）》，2018 年。[13] M. Xiangyun、R. Nathan、X. Yu 和 F. Dieter，“将局部控制扩展至大规模拓扑导航”，载于《IEEE机器人与自动化国际会议（ICRA）》，2020 年。[14] D. Shah、B. Eysenbach、G. Kahn、N. Rhinehart 和 S. Levine，“ViNG：利用视觉目标学习开放世界导航”，载于《IEEE机器人与自动化国际会议（ICRA）》，2021 年。[15] L. Suomela、J. Kalliola、H. Edelman 和 J.-K. K¨am¨ar¨ainen，“PlaceNav：通过地点识别实现拓扑导航”，载于《IEEE机器人与自动化国际会议（ICRA）》，2024 年。[16] D. Shah、A. Sridhar、A. Bhorkar、N. Hirose 和 S. Levine，“GNM：驱动任意机器人的通用导航模型”，载于《IEEE机器人与自动化国际会议（ICRA）》，2023 年。[17] D. Shah、A. Sridhar、N. Dashora、K. Stachowicz、K. Black、N. Hirose 和 S. Levine，“ViNT：用于视觉导航的基础模型”，载于《机器人学习会议（CoRL）》，2023 年。[18] A. Sridhar、D. Shah、C. Glossop 和 S. Levine，“NoMaD：用于导航与探索的目标掩码扩散策略”，载于《2024 IEEE机器人与自动化国际会议（ICRA）》，2024 年。[19] M. Caron、H. Touvron、I. Misra、H. Jegou、J. Mairal、P. Bo[20] janowski 和 A. Joulin，“自监督视觉变换器中的新兴特性”，载于《IEEE/CVF计算机视觉国际会议（ICCV）》，2021 年。[20] NVIDIA，“Isaac Sim：高保真机器人仿真平台”，https://developer.nvidia.com/isaac-sim，未注明日期，访问日期：2024-12-25。[21] P. Anderson、A. Chang、D. S. Chaplot、A. Dosovitskiy、S. Gupta、V. Koltun、J. Kosecka、J. Malik、R. Mottaghi、M. Savva 和 A. R. Zamir，“关于具身导航智能体的评估”，arXiv:1807.06757，2018 年。