视觉导航的认知绘图和规划

孙广岩 [1]
[1] 中山大学，智能工程学院，学号：20354242

摘要

基于期末课程设计要求，我们需要结合视语言、语音识别、机器人平台，采用 Webots/ROS 等不同开源仿真环境，构建一个单/多智能体的认知导航、认知规划、认知控制仿真算例，也算是期中仿真实验的一个延续。上次的仿真实验是基于人工势场的比较传统的方法，这次我继续延续了认知路径规划这个方向并确定了一篇更加前沿基于深度学习的论文来进行实验仿真。在这次实验中，我更加深刻的理解了认知科学这门学科，并了解了目前这门学科的前沿研究。

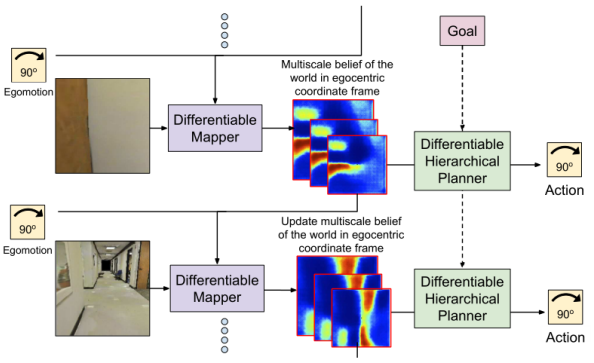

图 1. 总体网络架构：导航网络由绘图和规划模块组成。绘图器将数据写入到一个潜在的空间记忆中，该记忆与自中心的环境绘图相对应，而规划者将此记忆与目标一起使用，以输出导航动作。绘图器不是全监督的，而是在学习过程中自然出现。

1. 引言

作为人类，当我们在新的环境中导航时，我们会借鉴以前在类似情况下的经验。我们对自由空间、障碍物和环境的拓扑结构进行推理，以常识性的规则和启发式的导航方法为指导。例如，要从一个房间到另一个房间，我必须先离开最初的房间；要去大楼另一端的房间，进入走廊比进入会议室更有可能成功；厨房更有可能位于大楼的开放区域而不是隔间的中间。论文 [1]的目标是设计一个学习框架，以获得这种经验，并在机器人在新环境中的导航问题上证明这一点。

作者采用认知绘图和规划（CMP）方法进行视觉导航（图1）。CMP 由以下两部分组成：a) 捕捉全局布局的空间记忆；b) 一个在给定部分信息下可以规划路径的规划器。绘图器和规划器被组合成一个统一的架构，可以通过训练来利用世界的特性。绘图器融合了代理在一段时间内观察到的输入视图的信息，在自上而下的视图中产生一个关于世界的以自我为中心的多尺度信念。规划器使用这个世界的多尺度自我中心信念来规划通往指定目标的路径，并输出要采取的最佳行动。这个过程在每个时间步骤中重复进行，以将代理送到目标。

在每个时间步骤，代理通过以下方式更新前一个时间步骤的世界信念：a）使用自我运动将前一个时间步骤的信念转化为当前的坐标框架；b）结合当前世界观的形成来更新信念。这使代理人在移动过程中逐步改善其世界模型。与以往工作最显著的对比是，我们的方法是经过端到端的训练，以便在这个世界上采取良好的行动。为此，论文没有分析计算信念的更新（通过运动的经典结构），而是将其作为一个学习问题，并训练一个卷积神经网络，根据观察到的第一人称视角来预判更新。这使得提出的方法能够适应真实室内场景中的统计模式，而不需要对绘图阶段进行任何明确的监督。

规划器使用通过上述绘图操作获得的世界的度量信念来规划通向目标的路径。我们使用价值迭代作为我们的规划算法，但关键是使用可训练、可微分和分层版本的价值迭代。这有三个优点，a）

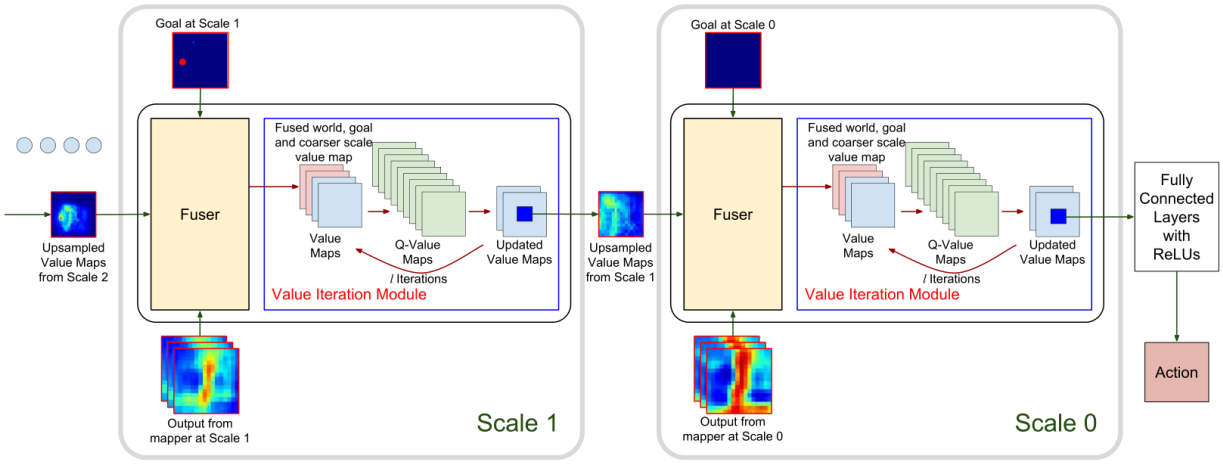

图 2. 分层规划器的体系结构：分层规划器采用绘图器输出的以自我为中心的多尺度世界信念，并使用表示为卷积和通道最大池的值迭代来输出策略。规划器是可训练和可微分的，并将梯度反向传播到绘图器。规划器在问题的多个尺度上进行操作（尺度 0 是最佳尺度），从而提高规划效率。

可训练性，它通过明确学习何时何地探索自然地处理部分观察到的环境，b）可微性，它使我们能够训练绘图器进行导航，以及 c）层次性，它使我们能够以时间复杂度规划到遥远目标位置的路径，该时间复杂度与目标的步数成对数关系。

本次报告安排如下：第 2 节介绍了整个 CMP 模型的框架。在第 3 节中进行模型的仿真实验，第 4 节为整个报告的总结。

2. 算法

作者将机器人建模为一个具有固定半径和高度的圆柱体，装配有视觉传感器（RGB 摄像头或深度摄像头），安装在固定高度，方向为固定方向。机器人配备有低级控制器，提供相对高级的宏观动作 $A_{x,\theta}$。这些宏操作是 a）保持原位，b）按 θ 向左旋转，c）按 θ 向右旋转，以及 d）向前移动 $x$cm，分别由 a_0, a_1, a_2 和 a_3 表示。我们进一步假设环境是一个网格图，机器人使用其宏操作在该图上的节点之间移动。机器人还可以进行精确的自我运动。

我们想了解这个机器人在以前从未遇到过的新环境中导航的策略。我们研究了两个导航任务，一个是几何任务，要求机器人前往机器人坐标系中指定的目标位置（例如，向前 250cm，向左 300cm），另一个是语义任务，要求机器人前往感兴趣的对象（例如椅子）。这些任务将在新环境中执行，机器人无法获得精确的环境地图或其拓扑结构。

2.1. 绘图器

首先来讲解一下绘图器部分，论文描述了学习网络的绘图部分如何将第一人称相机图像集成到自上而下的 2D 环境表示中，同时学习如何利用世界上的统计结构。请注意，与解析绘图系统不同，我们模型中的绘图相当于潜在表示。由于它直接输入到学习的规划模块中，因此它不需要对纯粹的自由空间表示进行编码，而是可以作为一般的空间记忆。该模型学习在地图中存储对生成成功计划最有用的任何信息。然而，为了在本节中进行具体描述，我们假设绘图器预测自由空间。

绘图器架构如图 22所示。在每一个时间步 t，在机器人的坐标系中保持自由空间 f_t 的累积估计。f_t 表示为一个多通道 2D 特征图，该特征图在俯视图中以度量方式表示空间。f_t 根据当前图像 I_t 进行估计，累积估计来自上一个时间步长 f_{t-1} 和使用以下更新规则，在最后一步和此步骤 e_t 之间执行自动作:

$$f_t = U(W(f_{t-1}, e_t), f'_t), where f'_t = \phi(I_t)$$

这里，W 是一个函数，它根据上一步 e_t 中的自运动对上一步 f_{t-1} 的自由空间预测进行转换，ϕ 是一个函数，它将当前图像 I_t 作为输入，并根据当前位置的环境视图（用 f_t 表示）输出一个自由空间的估计。U 是一个函数，它将当前视图中的自由空间预测与先前时间步骤中的累积预测相加。接下来，我们描述每个函数 W、ϕ 和 U 是如

何实现的。

函数 W 是用双线性采样实现的。鉴于自运动，我们计算一个后向流场 $\rho(e_t)$。这个后向流将当前自由空间图像 f_t 中的每个像素映射到它应该来自的前一个自由空间图像 f_t_1 中的位置。这个后向流 ρ 可以从自我运动中分析计算出来。函数 W 使用双线性采样，将这个流场应用于前一帧的自由空间估计。双线性抽样允许我们将重力从 f_t 传到 f_{t-1}，这将使我们有可能对这个模型进行端到端的训练。

函数 ϕ 是由卷积神经网络实现的。由于我们选择始终在机器人的坐标框架内表示自由空间，这成为一个相对容易学习的函数，因为网络只需要输出当前坐标的自由空间，而不是由机器人迄今为止的累积自运动决定的任意的世界坐标框架。

直观地说，该网络可以使用语义线索（如地板和墙壁等场景表面的存在，椅子和桌子等常见的家具物体）以及其他关于常见物体的尺寸和形状的先验知识来产生自由空间的估计，甚至对于可能只是部分可见的物体。

如图2所示，方程 ϕ 使用残差连接 [2]的卷积编码器，产生二维图像空间中的场景表示。这个表征通过完全连接层被转化为以自中心的二维俯视图。这个表征通过向上卷积层（也有残差连接）进行向上采样，以获得对当前帧的世界信念的更新。

2.2. 规划器

规划器是基于 Tamar 等人 [56] 提出的价值迭代网络，他们观察到一种被称为价值迭代的特殊类型的规划算法可以通过交替卷积和最大池化操作作为神经网络来实现，从而使规划器在其输入方面有所区别。价值迭代可以被认为是 Dijkstra 算法的一般化，其中每个状态的价值在每次迭代中都是通过对其邻居的价值取最大值，再加上转换到这些邻居状态的奖励来迭代重新计算的。这对二维网格世界的导航问题很有帮助，这些操作可以通过小的 3×3 内核和通道上的最大集合来实现。Tamar 等人还表明，这种价值迭代的重构也可以用来学习规划器（规划器卷积层中的参数），为每个状态的最优操作提供监督。因此，规划可以通过非常深的卷积网工作（具有基于通道最大池化）以可训练和可区分的方式完成。对于这个的问题，绘图器产生世界的二维顶视图，它与上述的二维网格世界结构相同，使用价值迭代网络作为可训练和可区分的规划器。

分层次的规划 我们的分层规划器在多个空间尺度上进行规划。我们从一个 k 次空间降采样的环境开始，在这个降采样的环境中进行 l 次数值迭代。这个值迭代过程的输出被中心裁剪，上采样，并用于在更精细的尺度上进行值迭代。这个过程重复进行，最终达到原始问题的解决。这个过程允许我们可以为远在 $l2^k$ 步之外的目标进行规划，而只进行（和反向传播）lk 次规划迭代。这种效率的提高是以近似规划为代价的。

在部分观测环境规划 价值迭代网络只在环境被完全观察到时才被评估，即在规划时整个地图是已知的。然而，对于我们的导航问题，地图只被部分观察到。因为规划器不是手工指定的，而是从数据中学习的，所以它可以学习自然地将部分观察到的地图考虑在内的策略。请注意，绘图器产生的不仅仅是一个关于世界的信念，还有一个不确定性 c_t，也就是置信度，规划器知道地图的哪些部分已经被观察到，哪些部分还没有被观察到。

2.3. 联合架构

最终架构认知绘图和规划（CMP）将上述的绘图器和规划器结合起来。在每个时间步骤中，绘图器根据当前的观察更新其对世界的多尺度信念。这个更新的信念被输入到规划器，规划器输出要采取的行动。如前所述，网络的所有部分都是可分的，允许进行端到端训练，并且没有额外的直接监督用于训练绘图模块–绘图器不是产生与一些地面真相自由空间相匹配的地图，而是产生允许规划者选择有效行动的地图。

3. 实验复现

实验复现基于 Python 的 Tensorflow 库，实验代码可从链接查看。

3.1. 环境搭建

首先进行 Python 环境的搭建，我的初始环境为 5 台 NVIdIA RTX3080 显卡，Tensorflow 版本为 1.15.5，CUDA 版本为 11.6。

官方示例推荐使用虚拟环境，但由于我是租用服务器使用，所以就跳过搭建虚拟环境，直接安装需要的库。

```
pip install --upgrade pip
# Install simple dependencies.
pip install -r requirements.txt

# Patch bugs in dependencies.
sh patches/apply_patches.sh
```

接下来安装Swiftshader，一个基于 CPU 的渲染器来渲染网格。也可以使用其他渲染器，用自己的渲染器的绑定来替换 render/swiftshader_renderer.py 中的 SwiftshaderRenderer。

```
mkdir -p deps
git clone --recursive
    https://github.com/google/swiftshader.git
    deps/swiftshader-src
cd deps/swiftshader-src && git checkout
    91da6b00584afd7dcaed66da88e2b617429b3950
git submodule update
mkdir build && cd build && cmake .. && make -j 16
    libEGL libGLESv2
cd ../../../
cp deps/swiftshader-src/build/libEGL* libEGL.so.1
cp deps/swiftshader-src/build/libGLESv2*
    libGLESv2.so.2
```

接下来安装PyAssimp来加载网格。可以使用其他库来加载网格，替换 Shape render/swiftshader_renderer.py 为自己的库来加载网格。

```
mkdir -p deps
git clone https://github.com/assimp/assimp.git
    deps/assimp-src
cd deps/assimp-src
git checkout
    2afeddd5cb63d14bc77b53740b38a54a97d94ee8
cmake CMakeLists.txt -G 'Unix Makefiles' && make
    -j 16
cd port/PyAssimp && python setup.py install
cd ../../../..
cp deps/assimp-src/lib/libassimp* .
```

接下来安装graph-tool来进行图像处理。

```
mkdir -p deps
git clone https://git.skewed.de/count0/graph-tool
    deps/graph-tool-src
cd deps/graph-tool-src && git checkout
    178add3a571feb6666f4f119027705d95d2951ab
bash autogen.sh
./configure --disable-cairo --disable-sparsehash
    --prefix=$HOME/.local
make -j 16
make install
cd ../../
```

实验环境搭建过程中也出现了一些问题，比如他的一些脚本代码是基于虚拟环境来写的，所以为了让整个环境顺利搭起来，我改了一些代码，还有因为可能是比较久没有维护过代码了，一些库版本的不匹配的问题我尝试了几次才把环境成功搭建起来。事实上，在整个实验复现中，搭建环境算是最耗时的环节了（如果不算数据集的下载）。

3.2. 数据集与模型准备

下载数据集，这其中包括两个部分：

1. 原始网格。我们需要这些在 noXYZ 文件夹中的网格。下载 tar 文件并把它们放到 stanford_building_parser_dataset_raw 文件夹。你需要下载 area_1_noXYZ.tar, area_3_noXYZ.tar, area_5a_noXYZ.tar, area_5b_noXYZ.tar, area_6_noXYZ.tar 用于训练和 area_4_noXYZ.tar 用于评估。
2. 用于设置任务的注释。我们将需要名为 Stanford3dDataset_v1.2.zip 的文件。将该文件放在 stanford_building_parser_dataset_raw 目录下。

预处理数据。

1. 使用 script_preprocess_meshes_S3DIS.sh 提取网格。之后之后，ls data/stanford_building_parser_dataset/mesh 应该有 6 个文件夹 area1, area3, area4, area5a, area5b, area6, 每个文件夹里有纹理和 obj 文件在每个目录中。
2. 用 script_preprocess_annoations_S3DIS.sh 从压缩文件中提取房间信息和语义。在这之后，应该有 data/stanford 中的 room-dimension 和 class-maps 文件夹。

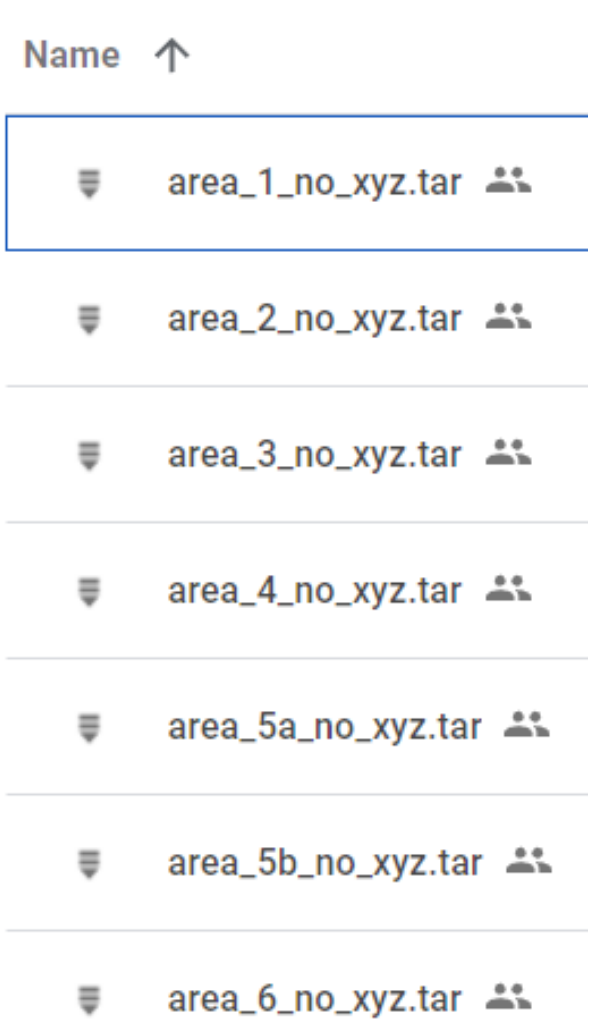

图 3. 数据集

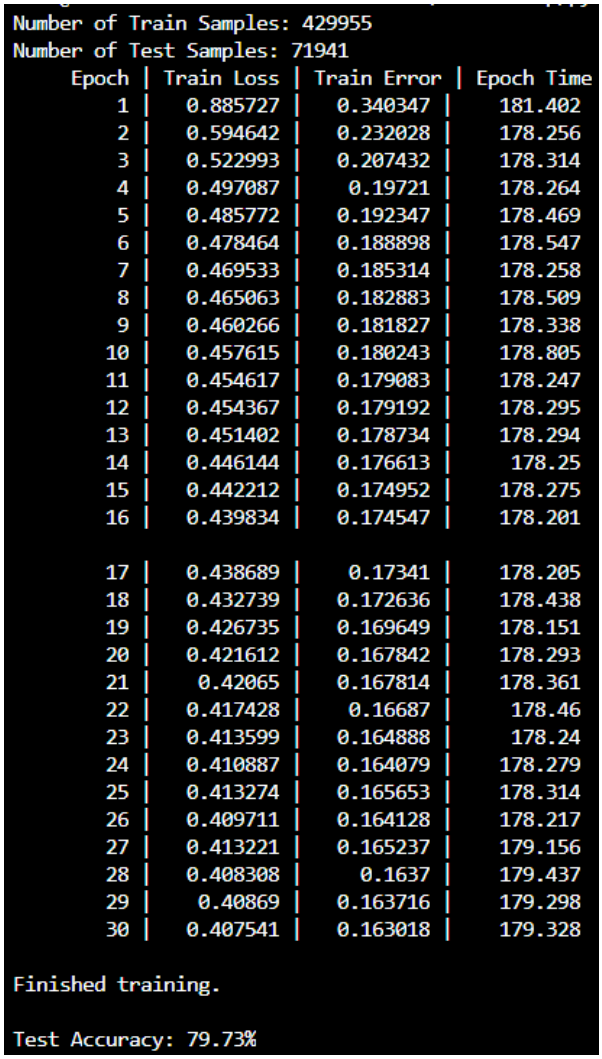

图 5. 运行截图

作者也提供了预训练模型供下载：

Config Name	Checkpoint	Mean Dist.	50%ile Dist.	75%ile Dist.	Success %age
cmp.lmap_Msc.clip5.sbpd_d_r2r	ckpt	4.79	0	1	78.9
cmp.lmap_Msc.clip5.sbpd_rgb_r2r	ckpt	7.74	0	14	62.4
cmp.lmap_Msc.clip5.sbpd_d_ST	ckpt	10.67	9	19	39.7
cmp.lmap_Msc.clip5.sbpd_rgb_ST	ckpt	11.27	10	19	35.6
cmp.lmap_Msc.clip5.sbpd_d_r2r_h0_64_80	ckpt	11.6	0	19	66.9
bl.v2.noclip.sbpd_d_r2r	ckpt	5.90	0	6	71.2
bl.v2.noclip.sbpd_rgb_r2r	ckpt	10.21	1	21	53.4
bl.v2.noclip.sbpd_d_ST	ckpt	13.29	14	23	28.0
bl.v2.noclip.sbpd_rgb_ST	ckpt	13.37	13	20	24.2
bl.v2.noclip.sbpd_d_r2r_h0_64_80	ckpt	15.30	0	29	57.9

图 4. 预训练模型汇总

3.3. 结果分析

我使用了 5 块 3080，接下来对结果进行简单的分析。我训练了三种环境，分别是 8x8,16x16 和 28x28 的网格环境，下面是 28x28 环境的运行结果。

最终将结果可视化分别从三种环境中挑了一张结果，可以看到如上图。训练的准确度评分如下表：

GRID	Train Acc(%)	Test Acc(%)
8x8	99.34	96.00
16x16	96.53	92.73
28x28	79.73	38.16

总的来说，我在查看结果时发现了模型会出现一些问题，比如在狭小空间中导航有时出现问题（通过部分打开的门进入，并卡在角落中，可能是由于间隙不够大，无法通过，缺少开口会导致路径缩短，在空间中就无法推理出路径而没有进展。这是由于整个环境的网格环境比较简单，只有 28x28，而在更加复杂的数据集中会有更精细的地图，这时模型应该会表现得更好。

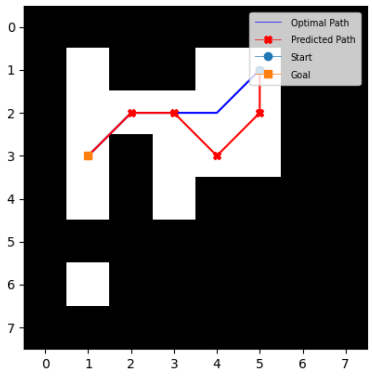

图 6. 8x8 环境结果

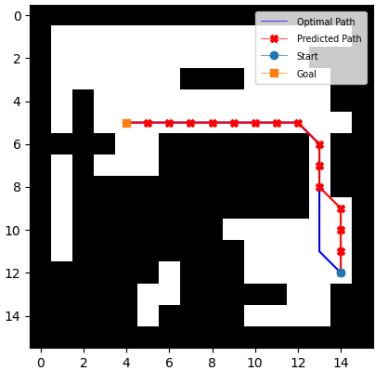

图 7. 16x16 环境结果

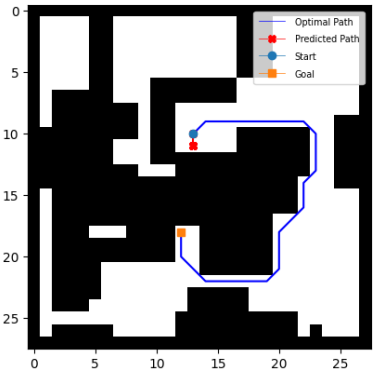

图 8. 28x28 环境结果

4. 总结

论文的方法让人想起了导航领域的经典工作，也涉及到构建地图，然后在这些地图中规划路径以到达所需的目标位置。然而，论文的方法在以下重要方面不同于经典工作：除了维护度量信念的架构选择之外，其他一切都是从数据中学习的。这导致了一些非常理想的特性：a）模型可以以任务驱动的方式学习室内环境的统计规律；b）联合训练绘图器和规划器，使规划器对绘图器的错误更加鲁棒；c）模型可以在新环境中在线使用，而无需预先构建地图。

在本文中，作者介绍了一种新的端到端神经网络结构，用于在新环境中导航。论文的体系结构学习从第一人称的角度绘制地图，并使用具有所学地图的规划器来规划行动，以便导航到环境中的不同目标。实验表明，这种方法优于其他不使用显式绘图和规划模块的直接方法。当然虽然工作在解决尚未从学习角度看待的问题方面取得了令人兴奋的进展，但要解决新环境中面向目标的视觉导航问题，还需要做更多的工作。

整体工作中的一个主要限制是对完美测距的假设。在现实世界中运行的机器人没有完美的测距，同时模型在不确定的运动中成功运行是必要的。

另外一个限制是构建和维护空间的度量表示。对于大型环境，这无法很好地扩展。我们通过对空间使用多尺度表示来克服这一问题。虽然这使我们能够研究更大的环境，但一般来说，如果在较粗的尺度下分辨率较低，则会使规划更接近实际情况，这可能会导致连接信息丢失。研究不受此类限制的空间的表示是未来重要的工作。

这次的实验整体来讲还是非常有趣的，我更加了解了前沿的利用视觉信息来进行路径规划的工作，也加深了在课上对于认知科学的理解。同时复现实验在科研中也是一个很重要的环节，这次的复现整体来讲确实还是比较有挑战的。当然这仅仅是个开始，之后我也会更加努力去在科研上去创造成就。

参考文献

[1] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik. Cognitive mapping and planning for visual navigation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017. 1

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016. 3