

动态环境下基于DRL的无人车自适应路径规划方法



搬砖的旺财 ✅

北京地平线信息技术有限公司 算法工程师

已关注

卓求、做梦的咸鱼、小Y等 112 人赞同了该文章

目录

- 1. 前言
- 2. 主要研究内容
- 3. D3QN PER算法
 - 3.1 DQN算法
 - 3.2 Double DQN算法
 - 3.3 Dueling DQN算法
 - 3.4 优先经验回放
- 4. 环境特征融合方案
 - 4.1 自身状态信息处理方法
 - 4.2 激光雷达点云信息处理方法
 - 4.3 视觉图像信息处理方法
 - 4.4 环境特征融合方案
- 5. 实验环境
 - 5.1 仿真实验环境的搭建
 - 5.2 小车的动作空间划分
 - 5.3 算法参数设置
 - 5.4 奖励函数的设置

▲ 已赞同 112



50 条评论



★ 收藏

...





----6.3 5个静态障碍+5个动态障碍环境
 ----6.4 10个动态障碍环境
7. 完整代码
 ----7.1 Software
 ----7.2 Installation
 -----7.2.1 解决pointgrey_camera_driver编译不过的问题
8. 其他问题

1. 前言

摘要

摘要

在星体表面运动时，行星车的路径规划作为自主决策系统的关键组成部分，是其能够安全高效地开展科学巡视探测的重要保证。特别是，在未来为了大规模探测高价值的科学目标，可能存在宇航员和多个移动机器人联合作业的情景，由此伴随着更为复杂的动态环境。在传统的规划方法中，行星车的所有行为都来自于地面人员预先定义好的规则，但是行星车的作业环境不是完全已知的，这时就需要行星车具备一定的自适应能力，应对环境发生的变化。为了进一步提升行星车的自主决策能力，并解决传统人为规划框架中过于依赖地图信息的问题，基于深度强化学习理论提出了端到端的路径规划方法。直接从传感器信息映射出动作指令再发布给行星车。同时采用不同的神经网络结构分别处理不同的传感器信息，最后将环境特征融合在一起，构成基于D3QN PER的多传感器行星车路径规划方法。

首先，对深度强化学习的基础理论进行了深入的研究，分别采用卷积神经网络处理视觉图像信息，长短周期记忆处理激光雷达点云信息和自身状态信息，推断给出了行星车的环境特征融合方案。同时汲取其他深度强化学习算法的优点，将D3QN PER算法应用于多传感器行星车的路径规划方法中，通过输出速度和角度指令来控制它的运动。

其次，搭建仿真环境，分三步验证算法的有效性。第一步，测试深度强化学习的相关算法是否能够在空旷的环境下引导行星车到达终点，由路径规划的结果可知，D3QN PER比DQN的其他算法更具优势。第二步，将行星车表面的砾砾和岩石转化为静态障碍环境，验证基于D3QN PER的多传感器行星车路径规划方法的避障性能，并将它与传统的RRT*算法进行了对比分析。第三步，将行星车表面的宇航员和多个移动机器人简化为动态障碍环境，直接加载静态障碍环境下训练好的网络模型，由实验结果可知，行星车对环境发生的变化具有较强的自适应能力，同时将该方法与传统的人工势场法进行了对比分析。

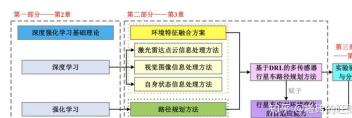
最后，通过加载的仿真环境中训练好的网络模型，使用Jackal移动机器人进行了实物验证。测试了基于D3QN PER的路径规划方法在真实环境中的适应性，从而为实际的巡视探测提供可靠的实验支撑。

关键词：行星车；深度强化学习；深度Q网络；路径规划；端到端

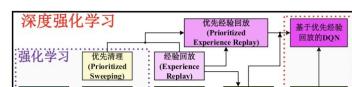
初衷是，在环境不是完全已知的情况下，希望行星车具备一定的自适应能力，应对环境发生的变化。所以，基于深度强化学习（Deep Reinforcement Learning, DRL）理论提出了端到端的路径规划方法，直接从传感器信息映射出动作指令再发布给行星车。同时采用不同的神经网络结构分别处理不同的传感器信息，最后将环境特征融合在一起，构成基于D3QN PER的多传感器行星车路径规划方法。

2. 主要研究内容

主要研究内容如图：



技术途径：



▲ 已赞同 112

▼ 50 条评论

分享

收藏

...



知乎

首发于
旺财的搬砖历险记

3. D3QN PER算法

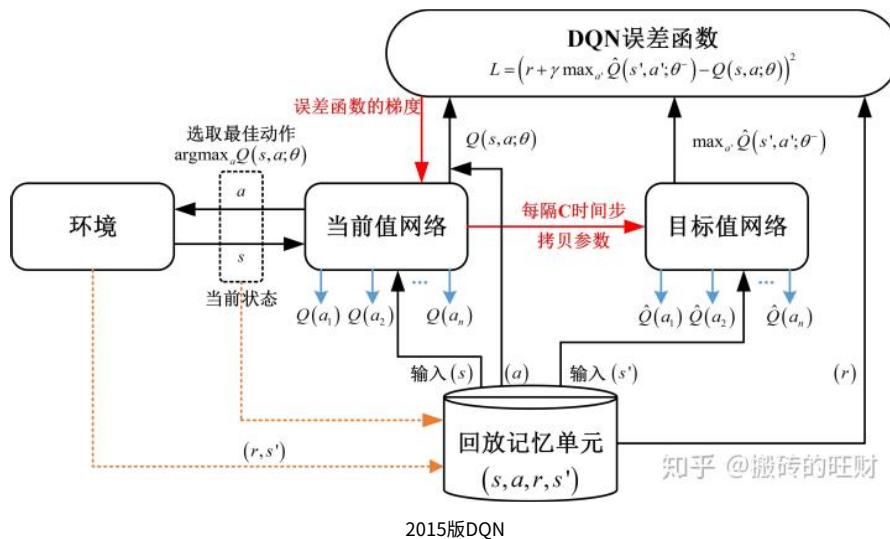
关于强化学习的一些基础知识，推荐大家去看专栏中关于David Silver课程的笔记：

搬砖的旺财：David Silver 增强学习——笔记合集（持续更新）

[zhuanlan.zhihu.com](https://zhuanlan.zhihu.com/collection/111111111111111111)



· 3.1 DQN算法



直接来看DQN的损失函数：

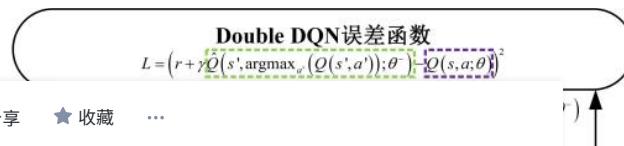
$$\mathcal{L}(\mathbf{w}) = \mathbb{E} \left[\left(r + \gamma \max_{a'} \hat{Q}(s', a', \mathbf{w}^-) - Q(s, a, \mathbf{w}) \right)^2 \right]$$

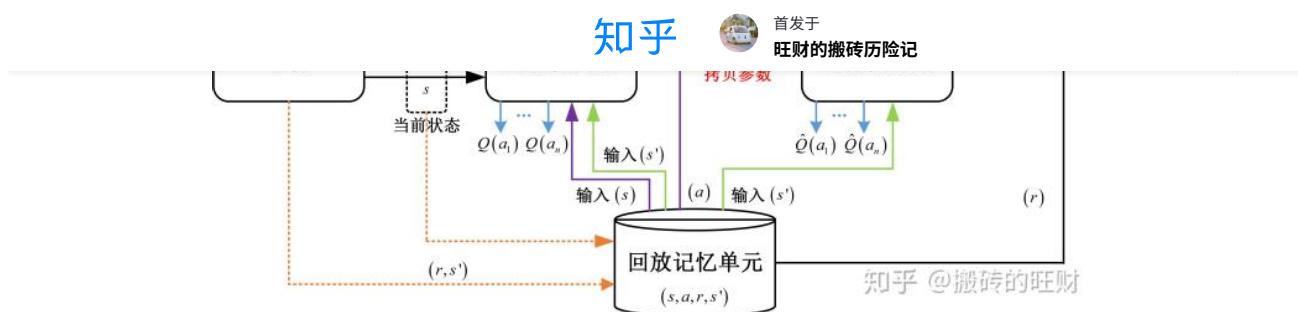
其中， \hat{Q} 是目标Q网络，参数为 \mathbf{w}^- ，负责生成训练过程中的目标，即目标Q值 $r + \gamma \max_{a'} \hat{Q}(s', a', \mathbf{w}^-)$ ； Q 是当前Q网络，参数为 \mathbf{w} ；值得注意的是， \hat{Q} 和 Q 的网络结构完全一致。

每训练C步，将当前Q网络的参数完全复制给目标Q网络，那么，接下来C步参数更新的目标将由更新后的目标Q网络负责提供。

具体的实施步骤请参见上图～

· 3.2 Double DQN算法



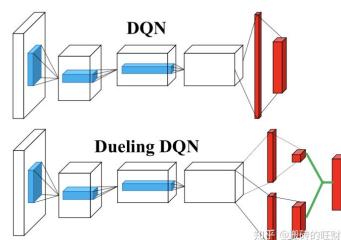


Double DQN算法主要是为了解决DQN算法中严重的过估计问题，它将目标Q值中动作的选择和动作的评估分开，让它们使用不同的Q网络。

先看Double DQN中的目标Q值： $r + \gamma \hat{Q}(s', \text{argmax}_{a'} Q(s', a', w); w^-)$ 。

当前Q网络负责选择动作，而带有“延迟效应”的目标Q网络用来计算目标Q值，具体的实施步骤请参见上图～

3.3 Dueling DQN算法



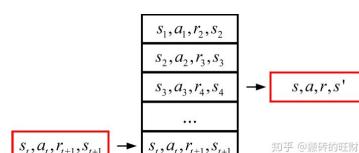
Dueling DQN相比于DQN在网络结构上做出了改进，在得到中间特征后“兵分两路”，一路预测状态值函数，另一路预测相对优势函数，两个相加才是最终的动作值函数。

3.4 优先经验回放

在DQN中使用经验回放的动机是：作为有监督学习模型，深度神经网络要求数据满足独立同分布假设；但样本来源于连续帧，这与简单的RL问题相比样本的关联性增大了不少。假如没有经验回放，算法在连续一段时间内基本朝着同一个方向做梯度下降，那么在同样的步长下这样直接计算梯度就有可能不收敛。所以经验回放的主要作用是：

克服经验数据的相关性，减少参数更新的方差；
克服非平稳分布问题。

经验回放的做法是从以往的状态转移（经验）中随机采样进行训练，如下图所示，这样操作使得数据利用率高，可以理解为同一个样本被多次使用。



优先经验回放（Prioritized Experience Replay, PER）的思路来源于优先清理（Prioritized sweeping），它以更高的频率回放对学习过程更有用的样本，这里使用TD-error来衡量作用的大小。TD-error的含义是某个动作的估计价值与当前值函数输出的价值之差。TD-error越大，则说明当前值函数的输出越不准确，换而言之，能从该样本中“学到更多”。为了保证TD-error暂时未知的新样本至少被回放一次，将它放在首位，此后，每次都回放TD-error最大的样本。

知乎

首发于
旺财的搬砖历险记

的缺失容易导致过拟合。

为了解决上述问题，结合纯粹的贪婪优先和均匀随机采样。确保样本的优先级正比于TD-error，与此同时，确保最低优先级的概率也是非零的。

首先，定义采样经验 i 的概率为：

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha}$$

其中， p_i 是第 i 个经验的优先级；指数 α 决定使用优先级的多少，当 $\alpha = 0$ 时是均匀随机采样的情况。

具体的，采用成比例的优先（Proportional prioritization），即

$$p_i = |\delta_i| + \epsilon$$

其中， δ_i 为 TD-error； ϵ 是为了防止经验的 TD-error 为 0 后不再被回放。在实现时采用二叉树（Sum tree）结构，它的每个叶子节点保存了经验的优先级，父节点存储了叶子节点值的和。这样，头节点的值就是所有叶子结点的总和。采样一个大小为 k 的 minibatch 时，范围 $[0, p_{total}]$ 被均分为 k 个小范围，每个小范围均匀采样，这样，各种经验都有可能被采样到。

为了消除由采样带来的偏差，加入重要性采样（Importance-sampling）。重要性采样权重为

$$\omega_j = \left(\frac{1}{N} \cdot \frac{1}{P(j)} \right)^\beta$$

其中， β 用于调节偏差程度，因为在学习的初始阶段有偏差是无所谓
的，但在后期就要消除偏差。为了归一化重要性采样权重，使 $\omega_j = \left(\frac{(N \cdot P(j))^{-\beta}}{\max_i \omega_i} \right)$ 。

来看一下 PER 和 Double DQN 结合后的伪代码：

输入：步长 η ，衰减因子 γ ，探索率 ϵ ，当前 Q 网络 Q ，目标 Q 网络 Q_{target} ，批量梯度下降的样本数 k ，目标 Q 网络参数更新频率 C ，指数 α 和 β ，迭代轮数 T ，经验池大小 N
输出：Q 网络参数

初始化经验池 $\mathcal{H} = \emptyset$ ，累计权重变化 $\Delta = 0$ ，优先级 $p_i = 1$

repeat ($t=1$ to T)

- 初始化状态 s ，得到特征向量 $\phi(s)$
- repeat** (对于 Episode 中的每一步)
 - 在 Q 中输入 $\phi(s)$ ，使用 ϵ -贪婪法在当前输出 Q 值中选择动作 a
 - 执行动作 a ，得到 s' 对应的特征向量 $\phi(s')$ 、奖励 r 以及是否终止 is_end
 - 将经验 $(\phi(s), a, r, \phi(s'), is_end)$ 以最大优先级 $p_i = \max_{i < t} p_i$ 存储于经验池 \mathcal{H}
 - $s \leftarrow s'$
 - repeat** ($j=1$ to k)
 - 从 SumTree 中基于概率 $P(j) = p_j^\alpha / \sum_i p_i^\alpha$ 采样样本
 - 计算重要性采样权重 $\omega_j = (N \cdot P(j))^{-\beta} / \max_i \omega_i$
 - 计算 TD-error $\delta_j = R_j + \gamma Q_{target}(S_j, \arg \max_a Q(S_j, a)) - Q(S_{j-1}, A_{j-1})$
 - 更新经验的优先级 $p_j \leftarrow |\delta_j|$
 - 累积权重变化 $\Delta \leftarrow \Delta + \omega_j \cdot \delta_j \cdot \nabla_\theta Q(S_{j-1}, A_{j-1})$
- until**
- 更新权重 $\theta \leftarrow \theta + \eta \cdot \Delta$ ，重置 $\Delta = 0$
- if** $T \% C = 1$ **then**
 - 更新目标 Q 网络参数 $\theta_{target} \leftarrow \theta$
- until** s 是终止状态

知乎 @搬砖的旺财

▲ 已赞同 112

50 条评论

分享

收藏

...

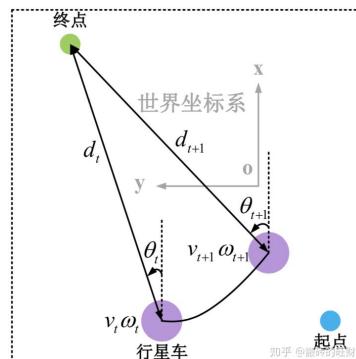


知乎

首发于
旺财的搬砖历险记

4. 环境特征融合方案

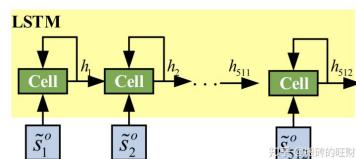
· 4.1 自身状态信息处理方法



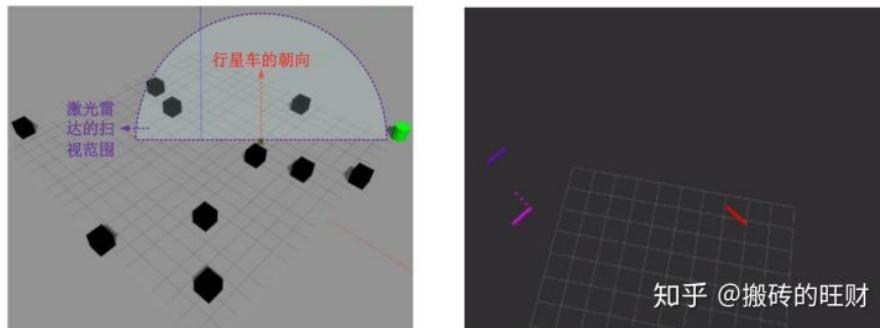
自身状态信息的表示方法请参见上图，它可以表示为一个数组 $[v_t, \omega_t, d_t, \theta_t]$ ，其中， v_t 和 ω_t 为 t 时刻小车的速度和角速度信息， d_t 和 θ_t 为 t 时刻小车相对终点的距离和角度信息。自身状态信息时刻指引着小车向终点移动。

· 4.2 激光雷达点云信息处理方法

LIDAR产生的点云属于长序列信息，比较难直接拆分成一个个独立的样本来通过CNN进行训练。所以采用LSTM网络来处理LIDAR点云信息，其中cell单元为512个，具体的网络结构如下图所示：



为了方便，控制LIDAR输出的点云信息为360维，更新频率是50赫兹，探测范围为 $[-90^\circ, 90^\circ]$ ，探测距离为 $[0.1, 10.0]$ ，单位是米，如下图所示：



· 4.3 视觉图像信息处理方法

尽管LIDAR擅长测量障碍物的距离和形状，但它实际上并不能用于确定障碍物的类型。计算机视觉可通过分类来完成这项任务，即给定相机的图像，可以标记图像中的对象。

本文采用CNN处视觉图像信息。输入图像 1024×728 经过预处理和叠帧（4帧）后变为 $80 \times 80 \times 4$ 的单通道灰色图。



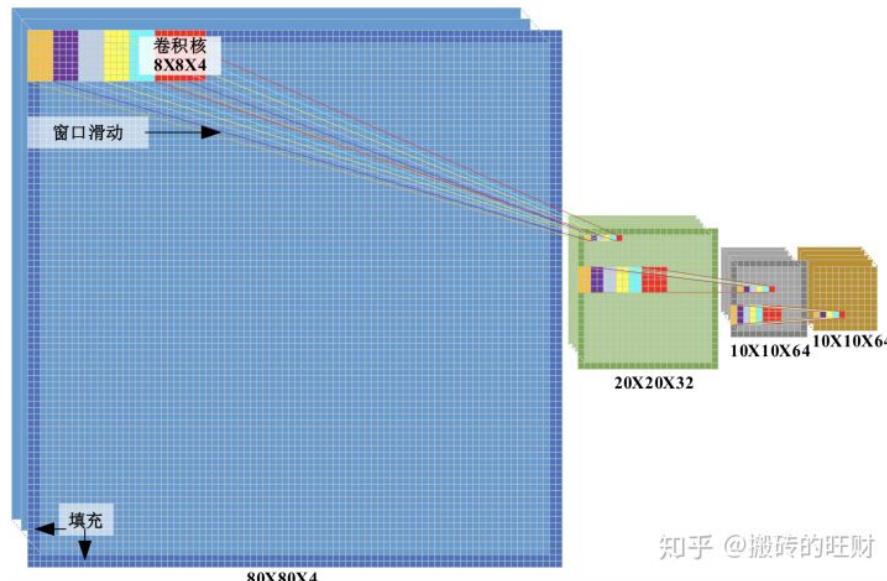
知乎

首发于
旺财的搬砖历险记

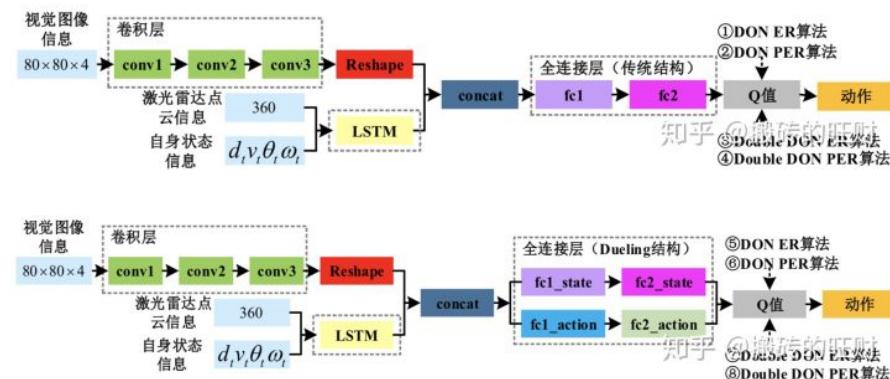
这里共采用了三个卷积层，它们分别是 conv1、conv2 和 conv3，具体参数如下：

卷积核名称	高度	宽度	输入通道数	输出通道数	步长	是否填充
conv1	$h_1 = 8$	$w_1 = 8$	4	32	$s_1 = 4$	'SAME'
conv2	$h_2 = 4$	$w_2 = 4$	32	64	$s_2 = 2$	'SAME'
conv3	$h_3 = 3$	$w_3 = 3$	64	64	$s_3 = 1$	'SAME'

卷积层的输入输出示意图如下：



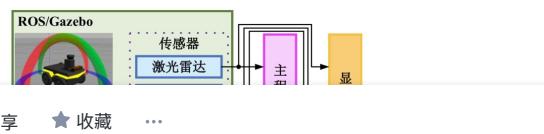
4.4 环境特征融合方案



5. 实验环境

5.1 仿真实验环境的搭建

基于ROS及Gazebo搭建的仿真实验环境的框架如图所示：





关于如何使Gazebo的仿真速度加快 10 倍, 请参考:

搬砖的旺财: 将Gazebo的仿真速度加快
10倍!!!
zhanlan.zhihu.com

知

· 5.2 小车的动作空间划分

动作编号	速度	角速度
0	1m/s	-1rad/s
1	1m/s	-0.5rad/s
2	1m/s	0rad/s
3	1m/s	0.5rad/s
4	1m/s	1rad/s
5	0.5m/s	-1rad/s
6	0.5m/s	0rad/s
7	0.5m/s	1rad/s
8	0m/s	-1rad/s
9	0m/s	知乎 @搬砖的旺财 +1rad/s
10	0m/s	

· 5.3 算法参数设置

参数	含义	大小
γ	衰减系数	0.99
Learning_rate	学习率	0.001
ϵ_{final}	贪婪策略终止参数	0.1
Num_action	动作数量	11
Num_max_episodes	最大训练幕数	500
Num_replay_memory	记忆池大小	5000
Num_training	训练步数	50000
Num_update	目标网络更新步数	500
Num_batch	minibatch 大小	知乎 @搬砖的旺财 32
Num_stackFrame	叠帧数量	4

ϵ - 贪婪法的 ϵ 参数从初始值1.0开始按照训练步数线性下降 $\epsilon = \epsilon - \frac{1.0}{\text{Num training}}$ 。

很有意思的一件事情, 经过测试发现, 记忆池的大小和训练步数保持10倍的关系是最好的~

参数	含义	大小
ε	防止经验的 TD-error 为 0 后不再被回放	0.00001
α	使用多少优先级	0.6
β_{init}	调节偏差程度	知乎 @搬砖的旺财 0.4

β 参数按照训练步数线性下降: $\beta = \beta + \frac{1.0 - \beta_{init}}{\text{Num training}}$ 。

· 5.4 奖励函数的设置

情景	数值
进入障碍物周围 1.5m 范围内	-1
远离目标	$-0.1 \times (d_t - d_{t-1})$
靠近目标	$0.1 \times (d_{t-1} - d_t)$
速度或角速度发生变化	$-0.01 \times (\omega_t - \omega_{t-1} + v_t - v_{t-1})$

▲ 已赞同 112

50 条评论

分享

收藏

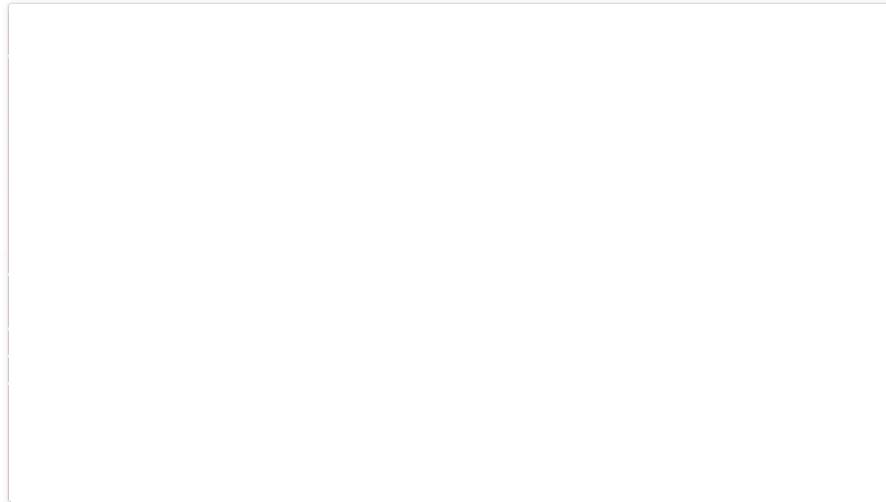
...



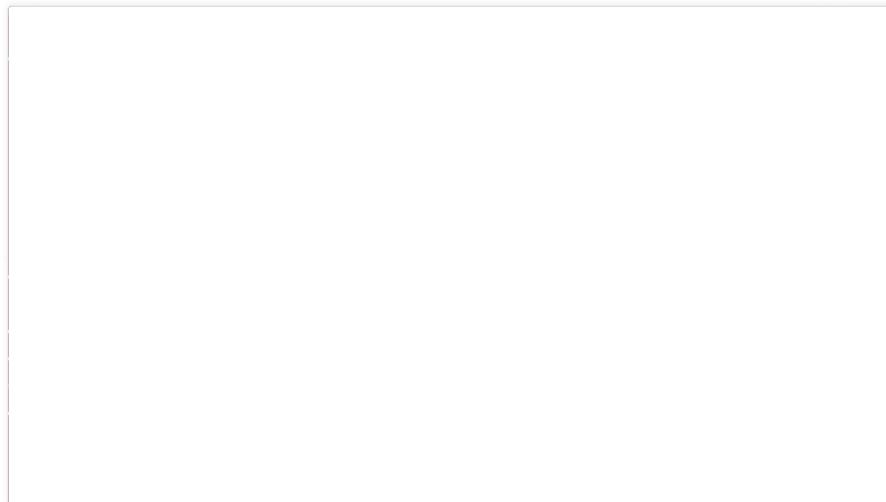


6. 效果展示

· 6.1 无障碍环境



· 6.2 静态障碍环境



· 6.3 5个静态障碍+5个动态障碍环境



▲ 已赞同 112



50 条评论



分享



收藏



...





首发于
旺财的搬砖历险记

· 6.4 10个动态障碍环境

7. 完整代码

▲ 已赞同 112



50 条评论



分享 收藏

...



知乎

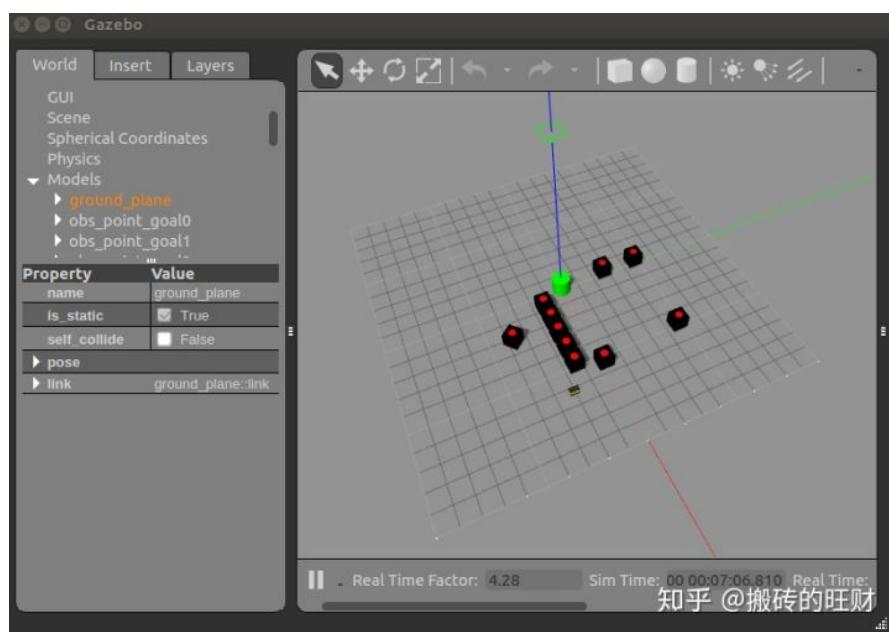
首发于
旺财的搬砖历险记

Ubuntu 16.04
ROS Kinect
Python 2.7.12
Tensorflow 1.12.0

· 7.2 Installation

```
git clone https://github.com/CoderWangcai...
(PS: 这一步请耐心等待, 训练好的model比较大~)
cd DRL_Path_Planning
catkin_make
source devel/setup.bash
roslaunch multi_jackal_tutorials ten_jackal_laser_add_apriltag.launch
```

出现的便是下面的场景:



然后, 把DRL_Path_Planning/src/tf_pkg/scripts/D3QN_PER_image_add_sensor_dynamic_10obstacle_world_30m_test.py中的路径补充完整:

```
# 静态障碍下训练好的模型
self.load_path = '../jackal/src/tf_pkg/scripts/saved_networks
/10_D3QN_PER_image_add_sensor_obstacle_world_30m_2_2019_06_02'
```

(PS: 如果需要运行其他python文件, 也请将其中的路径补充完整~)

另起一个终端:

```
python D3QN_PER_image_add_sensor_dynamic_10obstacle_world_30m_test.py
```

搞定啦, 出现的便是训练好的10个动态障碍环境~

· 7.2.1 解决pointgrey_camera_driver编译不过的问题

catkin_make可能会遇到关于pointgrey_camera_driver的问题:

▲ 已赞同 112

▼ 50 条评论

分享

收藏

...

ckage):

首发于
旺财的搬砖历险记

```
Add the installation prefix of "pointgrey_camera_driver" to
CMAKE_PREFIX_PATH or set "pointgrey_camera_driver_DIR" to a directory
containing one of the above files. If "pointgrey_camera_driver" provides a
separate development package or SDK, be sure it has been installed.
Call Stack (most recent call first):
multi_jackal/multi_jackal_description/CMakeLists.txt:7 (find_package)
```

解决方法如下：

```
cd src/
git clone github.com/ros-drivers/...
cd ..
catkin_make
```

```
-- => add_subdirectory(multi_jackal/multi_jackal_nav)
-- Using these message generators: gencpp;geneus;genlisp;gennodejs;genpy
-- +++ processing catkin package: 'wfov_camera_msgs'
-- => add_subdirectory(pointgrey_camera_driver/wfov_camera_msgs)
-- Using these message generators: gencpp;geneus;genlisp;gennodejs;genpy
-- wfov_camera_msgs: 3 messages, 0 services
-- +++ processing catkin package: 'pointgrey_camera_driver'
-- => add_subdirectory(pointgrey_camera_driver/pointgrey_camera_driver)
-- libflycapture not found in system library path
INFO:root:CPU architecture is x86_64
INFO:root:OS code name is xenial
INFO:root:Logging into ptgrey.com.
```

知乎 @搬砖的旺财

编译到上面这里的时候卡住了，没关系，ctrl+c，然后把刚刚Download的 pointgrey_camera_driver文件夹删除，因为现在已经生成相关的消息了，它不起作用了。

```
catkin_make
```

没问题啦，整个文件夹都编译过了～

8. 其他问题

如有问题请提issue～

特别鸣谢YP师兄，师兄在整个毕设的过程中，给了我很多技术上的支持，最后也帮我反复修改论文的结构～

请大家批评指正，谢谢～

编辑于 2019-09-01

「真诚赞赏，手留余香」

赞赏

还没有人赞赏，快来当第一个赞赏的人吧！

▲ 已赞同 112

▼ 50 条评论

分享

收藏

...



知乎

首发于
旺财的搬砖历险记

文章被以下专栏收录

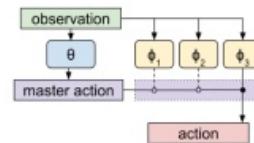


旺财的搬砖历险记

路漫漫其修远兮，吾将上下而求索。

关注专栏

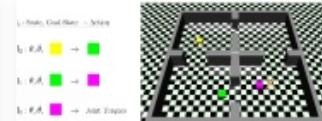
推荐阅读



分层强化学习 (MLSH)

赵英男

发表于强化学习-...



分层强化学习 (Learning Multi-Level Hierarchies)

赵英男

发表于强化学习-...

Rand
基于支这篇文
上的论
Distill
Expert
论文使
来做模
Beyon

▲ 已赞同 112



50 条评论



★ 收藏

...



知乎

首发于
旺财的搬砖历险记

50 条评论

切换为时间排序

写下你的评论...



云淡风轻

1 个月前

大佬 能不能膜拜论文

赞

1469

1 个月前

在哪能看到代码呀，本人在学习强化学习路径规划方面知识，想学习一下



赞

打铁工程师

29 天前

感谢分享

赞

Captain Hector

18 天前

大佬，我在尝试运行代码的时候出现了一个问题，就是无法导入cv2。因为PC上还要用另外的opencv做别的项目，这个问题困扰了我好久，有咩有办法解决

赞

ernest

17 天前

pointgrey_camera_driver/include/pointgrey_camera_driver/PointGreyCamera.h:46:36:
fatal error: flycapture/FlyCapture2.h: 没有那个文件或目录，下载了之后还有这个问题，大佬呀，这个该怎么解决？

赞

奔跑的小蜗

3 天前

你好，请问你这篇工作有没有发表英文论文呢？

赞

搬砖的旺财 (作者) 回复 奔跑的小蜗

3 天前

暂时没有

赞

上一页 1 2

已赞同 112

50 条评论

分享

收藏

...

