

Habitat-Sim使用文档(Meta开发的仿真器)

动机

Habitat简介

安装

1. 环境准备:

1.1 系统版本

1.2 Conda环境配置:

官方环境配置:

我的环境配置:

1.3 Habitat安装:

详细介绍

测试使用

没有物理交互的版本:

有物理交互的版本: 能够抓取3D物体, Habitat的物理基于Bullet物理引擎

example.py脚本功能使用

没有显示画面的测试:

运行example.py的一个交互的小例子

复现Habitat ICCV'19的榜单

加载MP3D或者是Gibson house

官方提供的pointnav_mp3d的一个例子

Habitat-sim C++ 查看器

Reference

Github: <https://github.com/facebookresearch/habitat-sim>

官网: <https://aihabitat.org/>

动机

由于配置环境、掌握新软件老是一个麻烦的事情，我通常会先看一下各位前辈的卓越奉献，非常感谢。

现在为了能够帮助大家快速熟悉Facebook（或Meta，总感觉Facebook更有感觉）的Habitat仿真器，同时避免踩坑。

Habitat简介

在安装之前首先介绍一下Habitat是干嘛的吧，Habitat是Facebook做的一个环境仿真器，对于研究机器人等需要交互的技术是一个比较好的选择，并且也兼容比较多数据集和场景。环境相对逼真，并且还能够和环境中的3D物体进行接触。目前应该主要用于研究Embodied AI。不过之前还有看到和ROS一起使用来运行CMU开发TARE等一系列环境探索算法，这个我也作为一个坑，之后填。现在先看一下它的环境效果吧：



支持的一些环境和模型：

Habitat-Sim

A high-performance physics-enabled 3D simulator with support for:

- 3D scans of indoor/outdoor spaces (e.g. [HM3D](#), [MatterPort3D](#), [Gibson](#), [Replica](#))
- CAD models of spaces and piecewise-rigid objects (e.g. [ReplicaCAD](#), [YCB](#), [Google Scanned Objects](#)),
- Configurable sensors (RGB-D cameras, egomotion sensing)
- Robots described via URDF (mobile manipulators like [Fetch](#), fixed-base arms like [Franka](#), quadrupeds like [AlienGo](#)),
- Rigid-body mechanics (via [Bullet](#)).

安装

1. 环境准备：

1.1 系统版本

在 windows 下面是搜索不到库的： [Habitat Sim :: Anaconda.org](#)，这里可以看到

目前Habitat只支持两个版本： MacOS 和 Linux

我的环境： Ubuntu18.04

1.2 Conda环境配置：

python版本可以用 3.7 或者 3.8 ， 3.9 是不支持的

官方环境配置：

```
1 # We require python>=3.7 and cmake>=3.10
2 conda create -n habitat python=3.7 cmake=3.14.0
3 conda activate habitat
```

我的环境配置：

```
1 conda create -n habitat python=3.8 cmake
2 conda activate habitat
```

1.3 Habitat安装:

我的安装命令【最常用】：`conda install habitat-sim withbullet -c conda-forge -c aihabitat`

详细介绍

Habitat安装分别支持以下参数:

- 有显示器版本和无显示器版本: `headless`
- 是否安装 `bullet` 物理引擎: `withbullet`
- 最新版本 `nightly` (仅当您需要最新版本中尚未提供的特定功能时才应使用此功能): 将 `-c aihabitat` 换成 `-c aihabitat-nightly`

以上参数可以自由组合, 通常都是安装最常用版本

- 【最常用版本】有显示器、有 `bullet` 物理引擎:

```
1 conda install habitat-sim withbullet -c conda-forge -c aihabitat
```

- 有显示器版本:

```
1 conda install habitat-sim -c conda-forge -c aihabitat
```

- 无显示器版本(即没有附加显示器, 例如在集群中)和具有多个 GPU 的机器上安装(此参数依赖于 EGL, 因此**不适用于** `MacOS`):

```
1 conda install habitat-sim headless -c conda-forge -c aihabitat
```

- 无显示器、有 `bullet` 物理引擎

```
1 conda install habitat-sim withbullet headless -c conda-forge -c aihabitat
```

对于指定版本只需要用 `=` 指定就行:

如安装0.1.6版本: `conda install habitat-sim=0.1.6 -c conda-forge -c aihabitat`

测试使用

没有物理交互的版本：

1. 下载3D场景：

```
1 python -m habitat_sim.utils.datasets_download --uids habitat_test_scenes --data-path /path/to/data/
```

/path/to/data/ 是你想将数据放在的文件夹

这个下载的场景并不提供予以标签，如果想要测试 `example.py` 程序的语义功能，需要下载 [Matterport3D](#)

2. 下载3D物体：

```
1 python -m habitat_sim.utils.datasets_download --uids habitat_example_objects --data-path /path/to/data/
```

3. 测试运行：

如果是直接conda安装的使用C++版本的命令就好

注意：源码安装应该要调整路径，小白慎用

C++：

```
1 #C++
2 # ./build/viewer if compiling locally
3 habitat-viewer /path/to/data/scene_datasets/habitat-test-scenes/skokloster-castle.glb
```

Python：用conda安装好像是没法用的

```
1 #Python
2 #NOTE: depending on your choice of installation, you may need to add '/path/to/habitat-sim' to your PYTHONPATH.
3 #e.g. from 'habitat-sim/' directory run 'export PYTHONPATH=$(pwd) '
4 python examples/viewer.py --scene /path/to/data/scene_datasets/habitat-test-scenes/skokloster-castle.glb
```

4. 按键使用（详见[Habitat C++查看器](#)）

W、S、A、D就是前后左右，Z是飞升，X是下地，鼠标左键是调整视角。

后面对于有接触部分详细介绍各种按键操作

有物理交互的版本：能够抓取3D物体，Habitat的物理基于 **Bullet** 物理引擎

1. 首先需要下载ReplicaCAD apartment 数据集（140MB）：下不下来找我

默认数据会下载到 `habitat-sim/data/` 路径下，可以通过添加 `--data-path /path/to/data/` 来修改

```
1 python -m habitat_sim.utils.datasets_download --uids replica_cad_dataset
```

源码安装的可以运行以下命令：

```
1 # with source (from inside habitat_sim/)
2 python src_python/habitat_sim/utils/datasets_download.py --uids replica_cad_dataset
```

比较完整有光照的数据，将 `replica_cad_dataset` 换成 `replica_cad_baked_lighting`（480MB）

2. 测试使用

C++:

```
1 habitat-viewer --enable-physics --dataset data/replica_cad/replicaCAD.scene_dataset_config.json -- apt_1
```

Python:

```
1 python examples/viewer.py --dataset data/replica_cad/replicaCAD.scene_dataset_config.json --scene apt_1
```

运行有光照的数据：

```
--dataset data/replica_cad_baked_lighting/replicaCAD_baked.scene_dataset_config.json --scene Baked_sc1_staging_00
```

3. 场景功能指令（详见[Habitat C++查看器](#)）：

对着场景窗口点击H能够在命令行看到控制指令的详情，这里提供中文翻译版本

example.py脚本功能使用

example.py

- change the sensor configuration: 改变传感器配置
- print statistics of the semantic annotations in a scene: 打印场景的标签数据
- compute action-space shortest path trajectories: 计算最短路径（以动作空间的组合形式表达）
- set other useful functionality: 设置其他有用的功能

更多功能请查看源码: `example.py` 和 `demo_runner.py`

conda安装的没有example.py，但是可以直接下载github的examples文件夹就行，在chrome浏览器有个插件 `GitZip for github`，能够只下载单个文件，不需要进行git clone了，方便conda安装的伙伴使用。



没有显示画面的测试：

这个例子是agent（机器人）按照规定路径走动，最终能够看到统计信息，类似：

```
640 x 480, total time: 3.208 sec. FPS: 311.7
```

```
1 python /path/to/habitat-sim/examples/example.py --scene /path/to/data/scene_datasets/habitat-test-scenes/skokloster-castle.glb
```

运行example.py的一个交互的小例子

```
1 python examples/example.py --scene /path/to/data/scene_datasets/habitat-test-scenes/skokloster-castle.glb --enable_physics
```

其实没什么卵用，就是一个固定视角下，看到几个物体能够掉落在桌子上，以体现物理交互特性。

加上 `--save_png` 能够生成每一个时刻的第一视角图像，注意会在当前的文件夹底下生成。

我为此写了一个脚本，能够将生成的图像放到一个 `demo` 文件夹，并且输出视频 `demo.mp4`，运行如下：

Github: <https://github.com/GuoPingPan/Habitat-Sim-Usage-Chinese>

```
1 #其他参数见脚本
2 python img_2_video.py ./
```

复现Habitat ICCV'19的榜单

注意：这里应该需要先下载MP3D数据集

```
1 examples/benchmark.py --scene /path/to/mp3d_example/17DRP5sb8fy/17DRP5sb8fy.glb
```

加载MP3D或者是Gibson house

```
1 examples/example.py --scene path/to/mp3d/house_id.glb
```


官方提供的 `pointnav_mp3d` 的一个例子

<https://aihabitat.org/docs/habitat-lab/habitat-lab-demo.html>

Habitat-sim C++ 查看器

- ▼ 欢迎使用 Habitat-sim C++ 查看器应用程序!

鼠标功能

在 LOOK 模式下（默认）：

左键：单击并拖动以旋转代理并向上/向下查看。

右键：（物理）随机产生一个物体。

shift + left：取被点击物体的语义ID和标签（目前只支持HM3D）；

shift + right：显示物体的表明网格。

ctrl + right：（物理）单击一个对象以对其进行体素化并显示体素化。

滚轮：视野放大和缩小（+shift 用于细粒度控制）

在 GRAB 模式下（需要在命令行加上使用“--enable-physics”）：

左键：并拖动以拾取和移动具有点对点约束的对象（例如球形接头）。

右键：拖动物体不受重力影响。

右键拖动物体不放开 + 滚轮：

+ alt：拖住的物体进行yaw轴旋转

+ ctrl：拖住的物体进行pitch轴旋转

+ alt+ctrl：拖住的物体进行roll轴旋转

关键命令：

esc：退出应用程序。

'H'：显示此帮助信息。

'm'：切换鼠标模式 (LOOK | GRAB)。

TAB/Shift-TAB：循环到场景数据集中的下一个/上一个场景。

ALT+TAB：重新加载当前场景。

代理控制：

'wasd'：向前/向后、向左/向右移动代理的身体。

'zx'：向上/向下移动代理的身体。

箭头键：向左/向右转动特工的身体，让相机向上/向下看。

'g': 将代理随机放置在 NavMesh 上（如果已加载）。

'q': 查询代理的状态并打印到终端。

'l': 将代理位置/方向保存

到“./saved_transformations/camera.year_month_day_hour-minute-second.txt”。

'j': 从文件系统加载代理位置/方向，或者从当前实例中的上次保存加载代理位置/方向。

相机设置

'4': 循环切换相机模式（相机、鱼眼、等距柱状）

'5': 切换正射/透视相机。

'6': 重置正射相机缩放/透视相机 FOV。

'7': 循环渲染模式（RGB、深度、语义）

可视化实用程序：

'l': 使用“default_light_override.lighting_config.json”中配置的设置覆盖默认照明设置。

'e': 启用/禁用视锥体剔除。

'c': 显示/隐藏 UI 覆盖。

'n': 显示/隐藏 NavMesh 线框。

'i': 将屏幕截图保存到“./screenshots/year_month_day_hour-minute-second/#.png”。

',': 渲染子弹碰撞形状调试线框覆盖（白色=活动，绿色=睡觉，蓝色=想睡觉，红色=不能睡觉）

对象交互：

SPACE: 打开/关闭物理模拟

',': 如果不是连续模拟，则进行单个模拟步骤。

'8': 在代理前面实例化一个随机原始对象。

'o': 在代理前面实例化一个随机的基于文件的对象。

'u': 删除最近实例化的刚体对象。

't': 通过在出现提示时输入文件路径，从 URDF 文件实例化相机前面的铰接对象。

+ALT: 导入带有固定基础的对象。

+SHIFT 快速重新加载先前指定的 URDF。

'b': 切换对象边界框的显示。

'p': 将当前模拟状态保存到 SceneInstanceAttributes JSON 文件（具有非冲突文件名）。

'v': （物理学）反转重力。

'g': （物理）显示阶段的有符号距离梯度矢量场。

'k': （物理）迭代舞台体素化符号距离场的不同范围。

附加实用功能：

'r': 将最近模拟帧的重播写入 --gfx-replay-record-filepath 指定的文件。

'/' : 将当前场景的元数据信息写入控制台。

导航轨迹可视化：

'1': 切换轨迹可视化的记录位置。

'2': 构建并显示轨迹可视化。

'3': 切换单色/多色轨迹。

'+' : 增加弹道直径。

'-' : 减小轨迹直径。

'F': （音频）在代理前面添加音频源

'O': （音频）运行音频模拟

Reference

其他问题还是看官网叭：<https://github.com/facebookresearch/habitat-sim>

还可以在这留言，我尽力解答！