环境配置

1. conda环境

从命令行窗口进入工程目录，执行以下命令（需要先安装Anaconda）

|  |
| --- |
| conda env create -f environment.yaml |
| conda activate neuconw |

1. pip安装python包

|  |
| --- |
| pip install -r requirements.txt |

1. 安装kaolin

|  |
| --- |
| git clone –recursive https://github.com/NVIDIAGameWorks/kaolin |
| cd kaolin |
| git checkout v0.12.0 |
| export IGNORE\_TORCH\_VER=1 |
| export KAOLIN\_INSTALL\_EXPERIMENTAL=1 |
| export TORCH\_CUDA\_ARCH\_LIST="8.6 8.6"  # If using heterogeneous GPU setup, set the architectures for which to compile the CUDA code. |
| export CUB\_HOME=/usr/local/cuda-\*/include/ |
| python setup.py develop |

1. 安装nvidiffrast

|  |
| --- |
| git clone https://github.com/NVlabs/nvdiffrast |
| cd nvdiffrast |
| python setup.py install |

1. 下载数据集

Hritage-Rcon Dataset:

|  |
| --- |
| mkdir data && cd data |
| gdown --id 1q0IMij036L\_8FRpFeJf-A0ZPul3EySiV |
| gdown --id 1yR0rnuySmTUq6VU8\_hwrWXyT9FCqkfXt |
| gdown --id 1L7hNv1n5g19txwENWqlFGFCpP4Ys1aDt |
| gdown --id 1moVu5jtczxA279qoFMPrJfErQL4QmI4c |

Data目录如下：

|  |
| --- |
| └── NeuralRecon-W  └── data  └── hritage-rcon  ├──brandenburg\_gate  ├──lincoln\_memorial  ├──palacio\_de\_bellas\_artes  ├──pantheon\_exterior |

UrbanScene3D Dataset：

|  |
| --- |
| 下载链接：https://drive.google.com/drive/folders/1e91lEw56DUBbQgRTo48T3lVjo53SzEOd?usp=sharing |

1. 数据准备

对于Hritage-Rcon Dataset场景，进入NeuralRecon-W目录命令行执行:

|  |
| --- |
| for SCENE\_NAME in brandenburg\_gate lincoln\_memorial palacio\_de\_bellas\_artes pantheon\_exterior; do  scripts/data\_generation.sh data/heritage-recon/${SCENE\_NAME}  done |

对于UrbanScene3D Dataset场景，首先通过colmap构建工作空间:

|  |
| --- |
| colmap feature\_extractor --database\_path ./database.db --image\_path ./images |
| colmap exhaustive\_matcher --database\_path ./database.db |
| mkdir sparse && colmap mapper --database\_path ./database.db --image\_path ./images --output\_path ./sparse |
| mkdir dense && colmap image\_undistorter --image\_path ./images --input\_path ./sparse/0 --output\_path ./dense --output\_type COLMAP |

输出：dense文件夹，目录结构如下：

|  |
| --- |
| └── dense    ├── images    │   ├── 0.JPG    │   ├── ...    │   └── 48.JPG    ├── run-colmap-geometric.sh    ├── run-colmap-photometric.sh    ├── sparse    │   ├── cameras.bin    │   ├── images.bin    │   └── points3D.bin    └── stereo        ├── consistency\_graphs        ├── depth\_maps        ├── fusion.cfg        ├── normal\_maps        └── patch-match.cfg |

将dense文件夹，转移到data目录下，重名为场景名（以PolyTech为例），进入NeuralRecon-W目录命令行执行：

|  |
| --- |
| python tools/prepare\_data/prepare\_semantic\_maps.py --root\_dir $WORKSPACE\_PATH --gpu 0 |

将UrbanScene3D\_config\PolyTech中的config文件复制到data/PolyTech目录下（其他场景的需要确定重建区域，并通过ReconArea/main.py计算并修正config文件中的参数），进入NeuralRecon-W目录命令行执行：

|  |
| --- |
| scripts/data\_generation.sh ./data/PolyTech |

注：使用banScene3D Dataset构建工作空间时，如果内存空间不足，可以通过PictureSmpling.py脚本进行压缩（要修改脚本中的图片路径）。

1. 训练

对于Hritage-Rcon Dataset场景，进入NeuralRecon-W目录命令行执行:

|  |
| --- |
| # `SCENE\_NAME` 改为要重建的场景名  scripts/train.sh $EXP\_NAME config/train\_${SCENE\_NAME}.yaml $NUM\_GPU $NUM\_NODE |

对于UrbanScene3D Dataset场景（以PolyTech为例），进入NeuralRecon-W目录命令行执行:

|  |
| --- |
| # `SCENE\_NAME` 改为要重建的场景名  scripts/train.sh PolyTech config/train\_PolyTech.yaml 4 1 |

1. 可视化

提交代码中包含四个重建场景的模型，可通过以下命令进行可视化:

|  |  |
| --- | --- |
| 模型展示GUI | python main.py |
| Brandenburg 场景 | python view/brandenburg\_view.py |
| lincoln\_memorial场景 | python view/lincoln\_memorial\_view.py |
| Castle场景 | python view/castle\_view.py |
| L7场景 | python view/L7\_view.py |