

main

1 Branch








0 Tags

Go to file

Go to file







<> Code

...

 Lyz	release pre-trained weights and inference code	c53ad42 · 8 hours ago	🕒 2 Commits
 scripts	release pre-trained weights and inference code	8 hours ago	
 slam3r	release pre-trained weights and inference code	8 hours ago	
 .gitignore	release pre-trained weights and inference code	8 hours ago	
 README.md	release pre-trained weights and inference code	8 hours ago	
 recon.py	release pre-trained weights and inference code	8 hours ago	
 requirements.txt	release pre-trained weights and inference code	8 hours ago	

About

Real-time dense scene reconstruction with SLAM3R

-  Readme
-  Activity
-  Custom properties
-  86 stars
-  18 watching
-  2 forks

Report repository

Releases

README

SLAM3R

Paper: [arXiv](#)

TL;DR: A real-time RGB SLAM system that performs dense 3D reconstruction via points regression with feed-forward neural networks.

TODO List

- ☒ Release pre-trained weights and inference code.
- ☐ Release Gradio Demo.
- ☐ Release evaluation code.
- ☐ Release training code and data.

Installation

1. Clone SLAM3R

```
git clone https://github.com/PKU-VCL-3DV/SLAM3R.git
cd SLAM3R
```

2. Prepare environment

```
conda create -n slam3r python=3.11 cmake=3.14.0
conda activate slam3r
# install torch according to your cuda version
pip install torch==2.5.0 torchvision==0.20.0 torchaudio==2.5.0 --index-url https://download.pytorch.org/whl/cu118
pip install -r requirements.txt
# optional: install XFormers according to your pytorch version, see https://github.com/facebookresearch/xformers
pip install xformers==0.0.28.post2
```

3. Optional: Compile cuda kernels for RoPE

```
cd slam3r/pos_embed/curope/
python setup.py build_ext --inplace
cd ../../../../
```

4. Download the SLAM3R checkpoints for the [Image-to-Points model](#) and the [Local-to-World model](#), and place them under

```
./checkpoints/
```

Demo

Replica dataset

To run our demo on Replica dataset, download the sample scene [here](#) and unzip it to `./data/Replica/`. Then run the following command to reconstruct the scene from the video images

```
bash scripts/demo_replica.sh
```

The results will be stored at `./visualization/` by default.

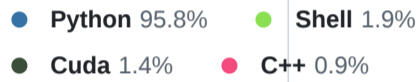
Self-captured outdoor data

We also provide a set of images extracted from an in-the-wild captured video. Download it [here](#) and unzip it to `./data/wild/`.

Packages

No packages published

Languages



Set the required parameter in this [script](#), and then run SLAM3R by using the following command

```
bash scripts/demo_wild.sh
```

You can run SLAM3R on your self-captured video with the steps above.

Citation

If you find our work helpful in your research, please consider citing:

```
@article{slam3r,
  title={SLAM3R: Real-Time Dense Scene Reconstruction from Monocular RGB Videos},
  author={Liu, Yuzheng and Dong, Siyan and Wang, Shuzhe and Yin, Yingda and Yang, Yanchao and Fan, Qingnan and Chen, Baoquan},
  journal={arXiv preprint arXiv:2412.09401},
  year={2024}
}
```

Acknowledgments

Our implementation is based on several awesome repositories:

- [Croco](#)
- [DUST3R](#)
- [NICER-SLAM](#)
- [Spanner](#)

We thank the respective authors for open-sourcing their code.