

参考文献

AA-RMVSNet [\[arXiv\]](#) [\[CVF\]](#) [\[PDF\]](#)

Wei Z, Zhu Q, Min C, et al. Aa-rmvsnet: Adaptive aggregation recurrent multi-view stereo network[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 6187-6196.

Cascade-MVSNet [\[arXiv\]](#) [\[CVF\]](#) [\[PDF\]](#)

Gu X, Fan Z, Zhu S, et al. Cascade cost volume for high-resolution multi-view stereo and stereo matching[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 2495-2504.

TransMVSNet [\[arXiv\]](#) [\[PDF\]](#)

Ding Y, Yuan W, Zhu Q, et al. TransMVSNet: Global Context-aware Multi-view Stereo Network with Transformers[J]. arXiv preprint arXiv:2111.14600, 2021.

LoFTR [\[arXiv\]](#) [\[CVF\]](#) [\[PDF\]](#)

Sun J, Shen Z, Wang Y, et al. LoFTR: Detector-free local feature matching with transformers[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 8922-8931.

EPP-MVSNet [\[CVF\]](#) [\[PDF\]](#)

Ma X, Gong Y, Wang Q, et al. EPP-MVSNet: Epipolar-Assembling Based Depth Prediction for Multi-View Stereo[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 5732-5740.

PatchmatchNet [\[arXiv\]](#) [\[CVF\]](#) [\[PDF\]](#)

Wang F, Galliani S, Vogel C, et al. PatchmatchNet: Learned Multi-View Patchmatch Stereo[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 14194-14203.

ResNeSt [\[arXiv\]](#) [\[PDF\]](#)

Zhang H, Wu C, Zhang Z, et al. Resnest: Split-attention networks[J]. arXiv preprint arXiv:2004.08955, 2020.

NeuS [\[arXiv\]](#) [\[PDF\]](#)

Wang P, Liu L, Liu Y, et al. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction[J]. arXiv preprint arXiv:2106.10689, 2021.

致谢

稀疏重建部分使用[Colmap](#)完成相机参数的获取。

稠密重建部分的代码主要来源于[AA-RMVSNet](#)。

点云切割与可视化使用[CloudCompare](#)及[Meshlab](#)完成。

调用[Open3D](#)进行表面重建。

Cascade+Transformer的代码主要基于[kwea123](#)实现的[pytorch-lightning](#)版本的[Cascade-MVSNet](#)以及[LoFTR](#)进行实现。

窗户识别算法中部分思路参考了Color Space的[矩形识别算法](#)，图像处理技术主要基于冈萨雷斯的[数字图像处理（第三版）](#)。

语义分割部分调用了[PyTorch-Encoding](#)。

网站开发使用了[Element-Plus](#)的组件，使用[Potree](#)和[Three.js](#)进行渲染。

白模生成使用了[NeuS](#)的代码进行训练并测试。