表1.1 三维重建方法对比

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 方法 | 类型 | 数据输入 | 重建精度 | 重建速度 | 方法假设和限制 |
| SfM | 被动型 | 一组图像序列 | 只能稀疏重建 | 慢 | 待重建物体需为刚性且纹理丰富的物体 |
| 结构光法 | 主动型 | 主动投射结构光 | 较高  （毫米级） | 快 | 场景环境亮度低  待重建物体距离近  成本高 |
| 三角测距法 | 主动型 | 主动发射激光 | 高  （微米级） | 快 | 待重建物体距离近  成本高 |
| 阴影恢复法 | 被动型  （单目） | 一张图像 | 一般 | 一般 | 光源参数准确  成本高 |
| 纹理恢复法 | 被动型  （单目） | 一张图像 | 较高 | 快 | 待重建物体纹理规则  成本高  应用范围小 |
| 对极几何法 | 被动型  （双目） | 双目相机采集的图像或两台相机同时采集的图像 | 较高  （毫米级） | 较慢 | 两视点差距较小  相机已标定  待重建物体纹理丰富 |
| 深度学习法 | 被动型 | 双目相机采集的图像或两台相机同时采集的图像 | 视具体情况 | 快 | 模型效果依赖数据集 |

表5.1 开发环境

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 名称 | 版本/型号 | 说明 |
| 服务器 | Ubuntu | 16.04.6 LTS | amd64体系架构 |
| 显卡 | NVIDIA | Geforce RTX 2080 Ti | cuda V10.0，cudnn V7.0 |
| 开发语言 | Python | V2.7.12 | 充分适配Tensorflow-GPU的版本 |
| 深度学习框架 | Tensorflow | V1.13.1 | 版本由显卡内核决定 |
| 三维可视化平台 | Meshlab[33] | V2020.12 | 开源的点云模型可视化软件 |
| IDE | VSCode | V1.56.1 | Python集成开发环境 |
| 容器技术 | Docker | V20.10.5 | 将不同方法环境打包成独立的镜像 |
| SFTP | Cyberduck | V7.9.0 | SFTP远程连接服务器软件 |
| 版本控制 | Git | V2.24.3 | 开源的分布式版本控制工具 |

表5.2 主要依赖包参数

|  |  |  |
| --- | --- | --- |
| 名称 | 版本/型号 | 说明 |
| tensorflow-gpu | V1.13.1 | Tensorflow在GPU下的依赖环境 |
| numpy | V1.13 | Python科学计算依赖环境 |
| opencv-python | V3.2.0.8 | Python使用计算机视觉算法的依赖环境 |
| scikit-learn | V0.18 | Python机器学习库 |
| scipy | V0.18 | Python机器学习库 |
| matplotlib | V1.5 | Python绘图及数据可视化依赖环境 |
| Pillow | V3.1.2 | Python图像处理库 |
| nvidia-ml-py | V375.53 | NVIDIA GPU管理依赖环境 |

表5.3 不同三维重建方法在DTU测试集上的距离度量（*mm*）定量评估结果

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名称 | Acc. | Comp. | overall |
| SurfaceNet | 0.450 | 1.043 | 0.746 |
| MVSNet | 0.444 | 0.741 | 0.592 |
| 经典光度一致性方法 | 1.565 | 1.378 | 1.472 |
| 无监督学习方法 | 1.255 | 1.491 | 1.373 |

表5.4 不同三维重建方法在DTU测试集上的百分比度量（<1*mm*）定量评估结果

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名称 | Acc. | Comp. | f-score |
| SurfaceNet | 75.73 | 59.09 | 66.38 |
| MVSNet | 82.93 | 62.71 | 71.42 |
| 经典光度一致性方法 | 46.90 | 42.16 | 44.40 |
| 无监督学习方法 | 61.54 | 44.98 | 51.98 |