

3D Gaussian Splatting（3DGS）完整部署与使用流程指南

一、整体目标说明

本流程文档用于完整记录从零开始，在 Linux + NVIDIA GPU 环境下，成功部署并运行 3D Gaussian Splatting（3DGS），
并最终得到高质量新视角渲染结果（图片 / 视频），而非传统三角网格模型。

适用：

- 使用无人机 / 相机拍摄真实场景
- 希望得到“照片级真实”的自由视角效果
- 不以游戏资产 / CAD 建模为主要目标

二、硬件与系统要求

1. 操作系统：Ubuntu 20.04 / 22.04（强烈推荐）
2. GPU：NVIDIA 显卡（建议 $\geq 24\text{GB}$ 显存）
3. 驱动：官方 NVIDIA Driver (≥ 535)
4. CUDA：无需系统级 CUDA，使用 conda 提供的 CUDA 即可
5. 磁盘空间：至少 50GB（含训练缓存与中间文件）

三、环境准备（强烈建议严格按此执行）

1. 安装 Miniconda

官网：<https://docs.conda.io/en/latest/miniconda.html>

2. 创建干净环境

```
conda create -n 3dgs_clean python=3.11 -y  
conda activate 3dgs_clean
```

3. 安装 PyTorch（CUDA 11.8）

```
conda install -y pytorch=2.0.1 torchvision=0.15.2 pytorch-cuda=11.8 -c pytorch -c nvidia
```

4. 验证 PyTorch

```
python - <<'PY'  
import torch  
print(torch.__version__)  
print(torch.version.cuda)  
print(torch.cuda.is_available())
```

PY

四、获取 3D Gaussian Splatting 源码

```
git clone https://github.com/graphdeco-inria/gaussian-splatting.git  
cd gaussian-splatting  
git submodule update --init --recursive
```

五、安装依赖模块（关键步骤）

1. diff-gaussian-rasterization

```
cd submodules/diff-gaussian-rasterization  
python -m pip install -e . --no-build-isolation
```

这里需要单独从 git 上下载 zip 解压该文件，不然会不全

2. simple-knn

```
cd ..//simple-knn  
python -m pip install -e . --no-build-isolation
```

3. Python 依赖

```
conda install -y numpy<2 scipy matplotlib tqdm imageio imageio-ffmpeg scikit-image  
opencv  
python -m pip install plyfile
```

六、数据准备（以无人机影像为例）

目录结构示例：

```
data/  
└── images/  
    ├── 00001.jpg  
    ├── 00002.jpg  
    └── ...
```

1. 使用 COLMAP 或 convert.py 处理：

```
python convert.py -s data
```

生成：

- sparse/
- cameras.txt / images.txt
- points3D.ply

七、模型训练（核心步骤）

基础训练命令：

```
python train.py -s data -m data/output
```

显存不足可用：

```
python train.py -s data -m data/output --resolution 2
```

训练产物：

```
data/output/
├── point_cloud/
├── cfg_args
└── checkpoint_xxxx.pth
    └── train/ours_xxxx/
```

八、渲染新视角（核心成果）

渲染训练与测试相机：

```
python render.py -m data/output
```

输出路径：

```
data/output/train/ours_30000/renders/
```

说明：

- 每一帧是一个新视角渲染
- 这是 3DGS 的“真实结果”，而非网格
- gt 文件夹是原图像，render 文件夹是渲染出来的

九、生成视频（推荐做法）

使用 ffmpeg：

```
ffmpeg -framerate 30 -i %05d.png -c:v h264 -pix_fmt yuv420p output.mp4
```

得到：

- 丝滑环绕视频
- 可用于展示 / 汇报 / 演示

十、关于“3D 模型”的重要认知

1. 3DGS ≠ 三角网格模型

2. Blender / MeshLab 中看到的只是点云
3. 真正的效果来自“渲染新视角”
4. 若强行转 mesh, 会损失真实感

结论：

3DGS 是神经渲染技术，不是建模工具。