

# NeRF&3DGS实践

2024春 计算机图形学  
Project 3



# PJ3 总览

NeRF作为最受关注的3D图形学前沿技术，诞生仅四年，相关领域已经发生了爆发式的增长。本次PJ意在让同学们认识NeRF、学会NeRF的基本使用方式、了解NeRF的前景。**我们这里不区分NeRF和3DGS技术，同学可自由选择使用的技术。**

PJ3总分为20分，包含的内容如下：

- **基础部分**：使用NeRF进行物体的重建或生成，占12分
- **扩展部分**：NeRF探索性研究，占8分

由3~4人组队合作完成，给分全队一致



# 基础部分

**任务：**使用任意一种NeRF算法，重建身边的物体，例如：模型、人或动物、学校里的雕塑和建筑。

**要求：**

- 自己动手，可以参考开源仓库和论文
- 需提交模型链接、代码链接、一份报告和一份PPT
- 汇报和demo演示

**给分标准**

- 按时提交和汇报：8分，迟交酌情扣分，不得晚于给分日期
- 质量得分：4分，按**选题难度（场景，技术等）**和**重建质量（展示效果）**给分，。



# 步骤示例

## Step1: 收集素材

**自己用手机或者相机采集拍摄（推荐）**

**或者选择现有视频网站上的素材，例如：**

飞越复旦大学江湾校区一镜到底：<https://www.bilibili.com/video/BV1HM4y1t76C>

飞越复旦大学邯郸校区一镜到底：<https://www.bilibili.com/video/BV13h411w73U>

飞越复旦大学张江校区一镜到底：<https://www.bilibili.com/video/BV16h4y1Q7iG>

飞越复旦大学邯郸南区文体生活：<https://www.bilibili.com/video/BV1qG411y7Sc>

飞越复旦大学邯郸北苑一镜到底：<https://www.bilibili.com/video/BV1rN4y1C7gC>





# 步骤示例

<https://github.com/nerfstudio-project/nerfstudio>

## Step2: 安装NeRFStudio和COLMAP (推荐Linux环境)

### Create environment

Nerfstudio requires `python >= 3.8`. We recommend using conda to manage dependencies. Make sure to install [Conda](#) before proceeding.

```
conda create --name nerfstudio -y python=3.8
conda activate nerfstudio
pip install --upgrade pip
```

### Dependencies

Install PyTorch with CUDA (this repo has been tested with CUDA 11.7 and CUDA 11.8) and [tiny-cuda-nn](#). `cuda-toolkit` is required for building `tiny-cuda-nn`.

For CUDA 11.8:

```
pip install torch==2.1.2+cu118 torchvision==0.16.2+cu118 --extra-index-url https://download.pytorch.org
conda install -c "nvidia/label/cuda-11.8.0" cuda-toolkit
pip install ninja git+https://github.com/NVlabs/tiny-cuda-nn/#subdirectory=bindings/torch
```

See [Dependencies](#) in the Installation documentation for more.

### Installing nerfstudio

Easy option:

```
pip install nerfstudio
```

OR if you want the latest and greatest:

```
git clone https://github.com/nerfstudio-project/nerfstudio.git
cd nerfstudio
pip install --upgrade pip setuptools
pip install -e .
```

### Installing COLMAP

There are many ways to install COLMAP, unfortunately it can sometimes be a bit finicky. If the following commands do not work, please refer to the [COLMAP installation guide](#) for additional installation methods. COLMAP install issues are common! Feel free to ask for help in on our [Discord](#).

[Linux](#) [OSX](#) [Windows](#)

We recommend trying `conda`:

```
conda install -c conda-forge colmap
```

Check that COLMAP 3.8 with CUDA is successfully installed:

```
colmap -h
```



# 步骤示例

## Step3: 使用COLMAP计算相机位姿，进行初步重建

### Images or Video

To assist running on custom data we have a script that will process a video or folder of images into a format that is compatible with nerfstudio. We use **COLMAP** and **FFmpeg** in our data processing script, please have these installed. We have provided a quickstart to installing COLMAP below, FFmpeg can be downloaded from [here](#)

#### Tip

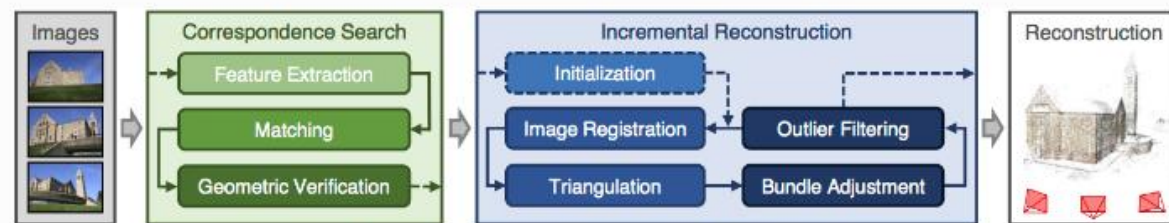
- COLMAP can be finicky. Try your best to capture overlapping, non-blurry images.

### Processing Data

```
ns-process-data {images, video} --data {DATA_PATH} --output-dir {PROCESSED_DATA_DIR}
```

### Training on your data #

```
ns-train nerfacto --data {PROCESSED_DATA_DIR}
```



COLMAP's incremental Structure-from-Motion pipeline.

Develop (G:) > nerfstudio > custom\_data >

<input type="checkbox"/>	Name	Date modified	Type	Size
	colmap	4/27/2024 12:28 PM	File folder	
	images	4/27/2024 11:07 AM	File folder	
	images_2	4/27/2024 11:07 AM	File folder	
	images_4	4/27/2024 11:07 AM	File folder	
	images_8	4/27/2024 11:07 AM	File folder	
	demo.mp4	4/27/2024 10:42 AM	MP4 - MPEG-4 电...	2,623 KB
	sparse_pc.ply	4/27/2024 11:23 AM	PLY File	2,145 KB
	transforms.json	4/27/2024 11:23 AM	JSON File	132 KB

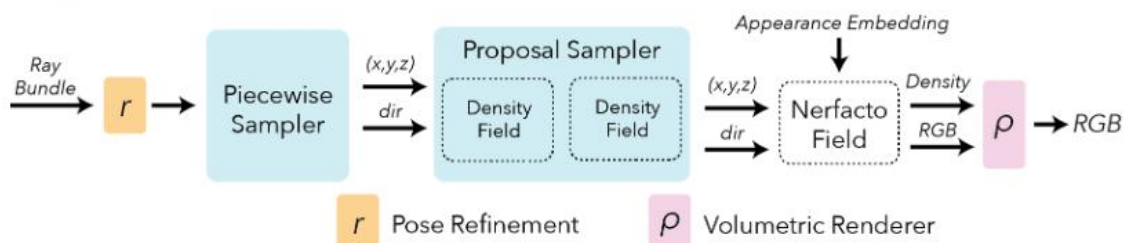




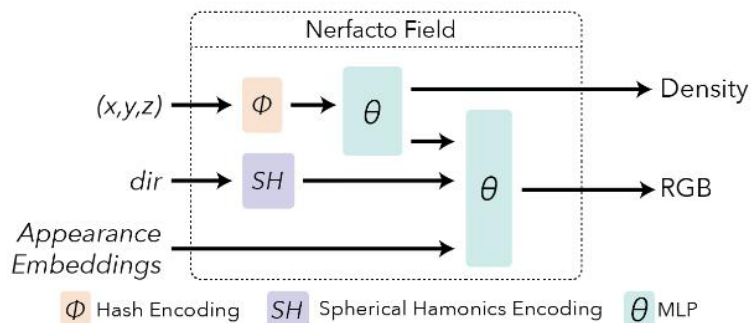
# 步骤示例

## Step4: 训练NeRFacto模型进行细致重建

### Pipeline



### Nerfacto Field



### Training on your data #

```
ns-train nerfacto --data {PROCESSED_DATA_DIR}
```

[12:42:15] Printing max of 10 lines. Set flag --logging.local-writer.max-log-size=0 to disable line wrapping.

Step (% Done)	Train Iter (time)	ETA (time)	Train Rays / Sec	
Step (% Done)	Vis Rays / Sec	Train Iter (time)	ETA (time)	Train Rays / Sec
19190 (63.97%)	38.080 ms	6 m, 51 s	110.05 K	
19200 (64.00%)	37.825 ms	6 m, 48 s	113.05 K	
19210 (64.03%)	31.311 ms	5 m, 37 s	138.25 K	
19220 (64.07%)	26.921 ms	4 m, 50 s	154.19 K	
19230 (64.10%)	26.484 ms	4 m, 45 s	156.39 K	
19240 (64.13%)	26.495 ms	4 m, 45 s	156.36 K	
19250 (64.17%)	27.067 ms	4 m, 50 s	153.32 K	
19260 (64.20%)	26.480 ms	4 m, 44 s	156.54 K	
19270 (64.23%)	26.359 ms	4 m, 42 s	157.29 K	
19280 (64.27%)	27.094 ms	4 m, 50 s	153.39 K	

viewer running locally at: <http://localhost:7007> (listening on 0.0.0.0)

<https://docs.nerf.studio/nerfology/methods/nerfacto.html>



# 步骤示例

## Step5: 可视化训练效果

<http://localhost:7007/>

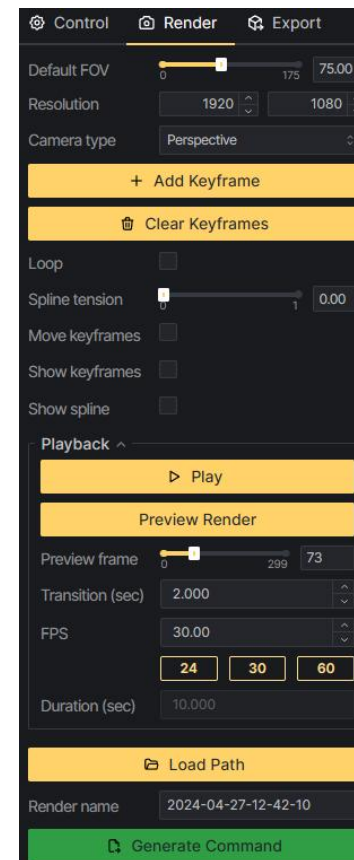






# 步骤示例

## Step6: 渲染路径





# 步骤示例

## Extra: 使用Splatfacto进行重建



<https://docs.nerf.studio/nerfology/methods/splat.html>

## Running the Method

To run splatfacto, run `ns-train splatfacto --data <data>`. Just like NeRF methods, the splat can be interactively viewed in the web-viewer, loaded from a checkpoint, rendered, and exported.

We provide a few additional variants:

Method	Description	Memory	Speed
splatfacto	Default Model	~6GB	Fast
splatfacto-big	More Gaussians, Higher Quality	~12GB	Slower



# 扩展部分

扩展部分为开放式，由同学自选NeRF相关课题完成，强调创新

要求：

- BE COOL, 发挥想象力, 展示创意。
- 不得完全照搬，要体现自己的工作量。
- 可以参考论文、博客与开源代码等资源，但需要给出明确的引用。
- 如果是纯报告类的课题，对报告质量的要求会更高。
- 提交代码链接、数据链接、模型链接和报告（与基础部分一起）
- 汇报和demo演示（与基础部分一起）

给分标准：

- 质量给分：8分，按选题难度和创新性给分
- 如果包含原创性的、有效的创新，可获得整个PJ3满分（20分）



# 扩展部分-推荐课题

## 一、静态场景 (NeRF/3DGS) :

- 提升合成质量: 去除NeRF和3DGS合成中的噪声。
- 低质图像输入: 噪声、模糊、低分辨率、反光等。
- 深度监督: 深度图/深度摄像头在NeRF中的应用。
- 其他改进: 合成质量、优化相机位姿、不同场景表示混合、大规模、多模态融合。

## 二、动态场景 (DynamicNeRF/4DGS) :

- 隐式引入变量 $T$ : 建立辐射场和流场, 比如Dynamic NeRF, 4DGS。
- 显式定义一个变形场, 将观测空间映射到规范化空间: 人脸 (Talking Head Synthesis)、人体 (Human Avatar) 等。

## 三、三维生成 (DreamFusion/DreamGaussian) :

- 使用DreamGaussian/LGM等工具生成3D物体和人物。
- 使用mixamo等软件为生成的物体或人物制作动画。

## 四、与其他学科进行结合: 比如医疗, 生物等等





# 扩展部分-质量评估

如果是**静态场景**的话，需要采用视角合成**NVS**作为代理任务的方式，评价重建质量。

**NVS**指Novel View Synthesis，目标是从已知视角合成新视角图片，建议使用**标准的NeRF测试集**，比如Mip-NeRF360, Tanks&Temples, Deep Blending等。

**具体的操作方式为：**

- 对同一个物体拍摄N张不同视角图片，随机采样其中的**20%**作为测试集。
- 用剩下的**80%**图片训练NeRF模型。
- 使用得到的NeRF模型渲染得到与测试集相同视角的图片。
- 采用**PSNR/SSIM/LPIPS**等指标，评估渲染结果和原测试集图片的相似度。
- 可以基于NeRFStudio进行修改，它有完整的训练和测试框架。

如果是**其他课题**的话，需要介绍参考的原论文并与之对比，评价重建或生成质量。





# 报告要求明细

报告标题：**2024图形学Project3**

作者：**姓名1 姓名2 姓名3 姓名4**

要求格式规范(推荐使用latex)，使用中文，描述清晰，可以参考NIPS会议模板  
(<https://media.neurips.cc/Conferences/NeurIPS2024/Styles.zip>)

**基础部分包含以下部分：**

- 结果展示与评估
  - 主观结果
  - 数值评估（可选）
- 所选方法介绍、数据介绍
- 训练/渲染设置和时间消耗（说明使用的设备）
- 给出组员的**姓名、学号与分工情况**

**扩展部分可根据选题自行组织。基础和扩展两部分整合为一份报告。**



# 演示要求明细

制作ppt，讲解如下内容：

- 选题的内容与意义
- 结果
- 所用方法
- 成员与分工

**6月13日 (16周) 进行demo展示，每组时间控制在5分钟左右：**

- 展示内容可以是图片、GIF、视频等
- 推荐使用实时渲染、交互式的方式展示
- BE COOL，发挥想象力，震撼的展示效果和创新的應用能够加分

**提交到elearning：报告+PPT打包成zip，DDL：6月13日23:59**



# 深度学习帮助文档

建议使用Pytorch框架

设备需求

- 本地平台：带有Nvidia游戏显卡的电脑，要求支持CUDA
- 远程平台：Google Colab/Kaggle/PaddlePaddle等

Colab使用教程：[Google Colab 快速上手](#)

本地环境搭建：

- 推荐使用Linux，按照要使用的开源NeRF仓库给出的要求搭建环境
- 注意根据自己的显卡型号选择CUDA版本和pytorch版本



# NeRF帮助文档

## NeRF方法梳理:

- <https://github.com/yenchenlin/awesome-NeRF>
- <https://github.com/MatrixBrain/awesome-NeRF>
- <https://github.com/weihaox/awesome-neural-rendering>

## 推荐代码仓库:

- 3D重建技术聚合 nerfstudio: <https://github.com/nerfstudio-project/nerfstudio>
- 3D生成技术聚合 threestudio: <https://github.com/threestudio-project/threestudio>
- 3DGS: <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/>
- 4DGS: <https://github.com/hustvl/4DGaussians>
- DreamGaussian: <https://github.com/dreamgaussian/dreamgaussian>
- LGM: <https://github.com/3DTopia/LGM>