

# 3D无限场景的生成

## 第一周

图形学最基础知识，包括相机标定，colmap，sfm等方法

[三维重建基础](#)

## LucidDreamer

### 论文解读

In this work, we propose a pipeline called LucidDreamer that utilizes Stable Diffusion and 3D Gaussian splatting to create diverse high-quality 3D scenes from various types of inputs such as text, RGB, and RGBD.

#### 1. 局限：dataset far from real world（特定领域）

- a. 利用stable diffusion生成3D模型，难以保证multi-view consistency
- b. diffusion模型学习：

[diffusion模型基础](#)

#### 2. Step: dream and alignment

- a. 投影点云，利用生成模型。根据深度提升三维空间
- b. Alignment algorithm: integrate
- c. 作为3dgs的initial points

简单来说，初始图像投影之后用diffusion模型生成，重新升维到3维空间后，用alignment算法和原先的点云群融合

#### 3. 输入：初始图像和深度图

#### 4. 输出：点云

#### 5. 优点：多场景，配合文字输入，生成更多视角

#### 6. 具体过程：

- a. 初始化点云，利用stable diffusion得到img和深度图（深度图的生成根据zero-depth the monocular depth estimation model），把二维图像升维到三维
  - i.  $P_0 = \phi_{2 \rightarrow 3}([I_0, D_0], K, P_0)$
- b. 聚合点云：把生成的点云和原来的点云聚合成更大的点云，需要满足一致性

c. dream过程：投影的image会有没有得到的点，因此利用mask来表示填充和没有填充的点，在没有填充的点继续进行stable diffusion和monocular depth的过程

i. 实际过程中，深度图可能会不一致（因为模型缺陷），所以需要optimal depth

$I_i = \mathcal{S}(\hat{I}, M_i)$ ,  $\hat{D}_i = \mathcal{D}(I_i)$ ,  $D_i = d_i \hat{D}_i$  新的图像I，是从旧图像和mask利用S来生成的，depth图也可以用来生成，di是系数来控制

$$d_i = \arg \min_d \left( \sum_{M_i=1} \left\| \phi_{2 \rightarrow 3}([I_i, d\hat{D}_i], K, P_i) - P_{i-1} \right\|_1 \right)$$
 这个是用来逼近di

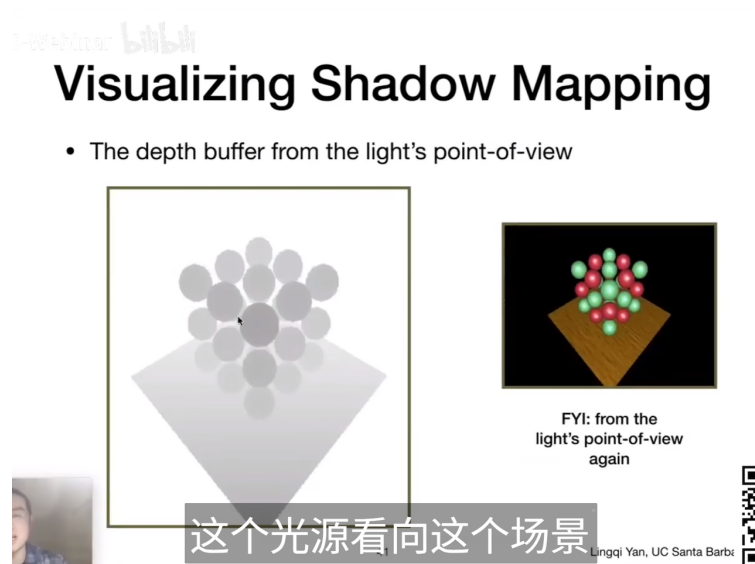
d. alignment：dream的过程是同时生成深度图和新图像，用off the shelf 的深度图生成方法会更好（适配更多的场景，更加的精确）

i. 需要解决consistency的问题，引入alignment算法。因为没有考虑多个depth之间的关系，直接移动点，让之间smooth。（需要计算vector），naive的移动会导致distort，需要利用差值算法解决

### 图形学基础（有差值算法）

i. 具体的过程：将图片沿着射线移动（因为深度是这个方向），找到最接近Pi-1位置的点，并计算深度改变的幅度（因为移动）

ii. 对于没有真实图像的地方，用线性差值的方法解决（深度吐的图片）



iii. 重复整个过程，就可以完成整个深度图的构造（其实整个alignment，就是在找更精确的di）

---

**Algorithm 1:** Constructing point cloud

---

**Input:** A single RGBD image  $[\mathbf{I}_0, \mathbf{D}_0]$ **Input:** Camera intrinsic  $\mathbf{K}$ , extrinsics  $\{\mathbf{P}_i\}_{i=0}^N$ **Output:** Complete point cloud  $\mathcal{P}_N$ 

```
1  $\mathcal{P}_0 \leftarrow \phi_{2 \rightarrow 3}([\mathbf{I}_0, \mathbf{D}_0], \mathbf{K}, \mathbf{P}_0)$ 
2 for  $i \leftarrow 1$  to  $N$  do
3    $\hat{\mathbf{I}}_i, \mathbf{M}_i \leftarrow \phi_{3 \rightarrow 2}(\mathcal{P}_{i-1}, \mathbf{K}, \mathbf{P}_i)$ 
4    $\mathbf{I}_i \leftarrow \mathcal{S}(\hat{\mathbf{I}}_i, \mathbf{M}_i)$ ,  $\hat{\mathbf{D}}_i \leftarrow \mathcal{D}(\mathbf{I}_i)$ 
5    $d_i \leftarrow 1$ 
6   while not converged do
7      $\tilde{\mathcal{P}}_i \leftarrow \phi_{2 \rightarrow 3}([\mathbf{I}_i, d_i \hat{\mathbf{D}}_i], \mathbf{K}, \mathbf{P}_i)$ 
8      $\mathcal{L}_d \leftarrow \frac{1}{\|\mathbf{M}_i\|} \sum_{\mathbf{M}_i=1} \|\tilde{\mathcal{P}}_i - \mathcal{P}_{i-1}\|_1$ 
9     Calculate  $\nabla_d \mathcal{L}_d$ 
10     $d_i \leftarrow d_i - \alpha \nabla_d \mathcal{L}_d$ 
11  end
12   $\mathbf{D}_i \leftarrow d_i \hat{\mathbf{D}}_i$ 
13   $\hat{\mathcal{P}}_i \leftarrow \phi_{2 \rightarrow 3}([\mathbf{I}_i, \mathbf{D}_i | \mathbf{M}_i = 0], \mathbf{K}, \mathbf{P}_i)$ 
14   $\mathcal{P}_i \leftarrow \mathcal{P}_{i-1} \cup \mathcal{W}(\hat{\mathcal{P}}_i)$ 
15 end
```

---

e. 最后就是利用3dgs高斯来完成训练和渲染（整个luciddreamer其实关注的就是初始化点云）

i. 值得注意的是：高斯损失函数只关注有真实结果的，mask=0的地方不会计入损失函数

### 高斯学习相关知识

#### 7. 疑问：

- a. 相机的内参和外参如何得到：The camera intrinsic matrix and the extrinsic matrix of  $\mathbf{I}_0$  are denoted as  $\mathbf{K}$  and  $\mathbf{P}_0$ , respectively. For the case where  $\mathbf{I}_0$  and  $\mathbf{D}_0$  are generated from the diffusion model, we set the values of  $\mathbf{K}$  and  $\mathbf{P}_0$  by convention regarding the size of the image.
- b. 高感知质量是什么：high perceptual quality
- c. 训练的时候添加M照片是为了什么（实验经验吗）：For the images to train the model, we use additional  $M$  images as well as  $(N + 1)$  images for generating the point cloud, since the initial  $(N + 1)$  images are not sufficient to train the network for generating the plausible output. The  $M$  new images and the masks are generated by reprojecting from the point cloud  $\mathcal{P}_N$  by a new camera sequence of length  $M$ , denoted as  $\mathcal{P}_{N+1}, \dots, \mathcal{P}_{N+M}$ .
- d. alignment移动点，为什么就可以解决一致性的问题，直接使用差值不可以么

8. **和项目的关系：**luciddreamer是利用stable diffusion和3dgs的整体框架，可以利用stable diffusion和深度图来提供多视角，新的三维场景。但是我们的项目是想要做超分，相当于并不是实现横向的新场景添加，是要实现纵向的超分效果。提供了一种思路，就是利用stable生成点云和alignment进行结合。因为luciddreamer本身有提供框架和代码，所以在luciddreamer上结合power of ten的思路，可以尝试实现。

## 代码复现

- 前几天主要是在进行luciddreamer的环境配置，周二开始进行代码调试，修改相应的参数来观察对生成速度和质量的影响
  - 我尝试自定义输入图像，发现就是复杂纹理区域生成的效果非常的差，可能是因为遮挡关系处理的不够准确
  - 文件结构：
    - model.py是3dgs模型定义
    - dreamer.py是文本到图像生成模块
    - alignment.py:是多视角对齐优化
    - renderer.py是 3d渲染
  - 参数调整：gaussian\_args是调整高斯球数量，密度。ptimization 是 修改学习率和迭代次数
- 之后尝试对luciddreamer来进行一些代码的修改
  - 在dreamer.py中添加了预处理函数，这样可以增强输入图像的细节
  - 修改renderer.py理的渲染参数，从64变成了128
    - self.num\_samples\_per\_ray=128

## Power of ten

### 论文解读

We achieve this through a joint multi-scale diffusion sampling approach that encourages consistency across different scales while preserving the integrity of each individual sampling process.

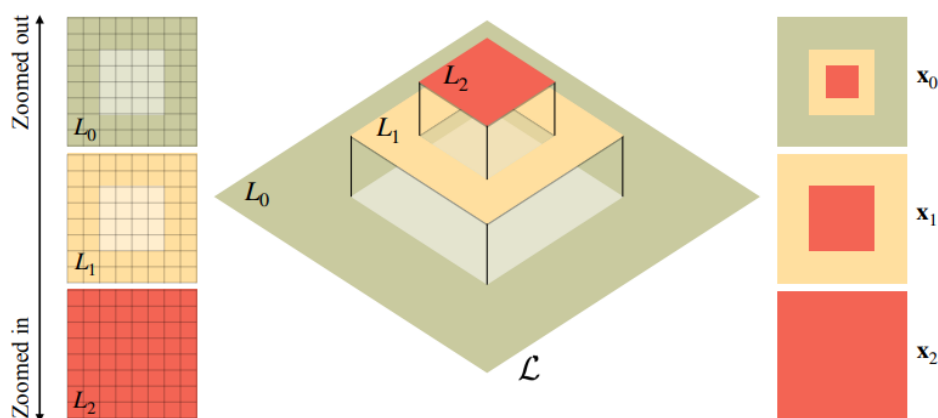
our method enables deeper levels of zoom than traditional super-resolution methods that may struggle to create new contextual structure at vastly different scales

- 优点：**相比传统的超分结构，在contextual上，上下文结构上具有更好的效果。现有的文生图模型，无法实现在zoom level上的consistent

- a. 传统的超分仅仅依靠图像信息，因此无法在deeper上实现细节，但是我们有text prompt可以解决这个问题
  - b. 每个scale上的plausible image，同时每个scale之间又是连续
  - c. 传统的超分也可以实现zoom in和zoom out的效果，但是上下文关系薄弱，递归的时候会产生问题
  - d. 同时生成整个consistent sequence
  - e. 总的来说，我希望在zoom in和zoom out过程中，整个语音可以被保持
2. 难点：zoom in的问题是语义semantic：因为我们生成的图片之间应该要有很多的过度关系（比如一个人的手掌放大会会有皱纹）
  3. 输入：text提示词（会包括不同的scale的提示词）
  4. 过程：
    - a. 先介绍了diffusion模型生成图像的过程，逐步添加高斯噪声来生成图片（train），如果train完了，就需要利用denoise的过程，来得到干净的图像
      - i. 我对stable diffusion模型的学习 diffusion模型的核心原理就是预测噪声，利用噪声消除来得到
    - b. 目标：利用生成模型生成一组图片，图片之间的在不同的 zoom level上（大图片经过裁剪），能够保持consistent

consistent way. This means that the image  $x_i$  at any specific zoom level  $p_i$ , should be consistent with the center  $H/p \times W/p$  crop of the zoomed-out image  $x_{i-1}$ .

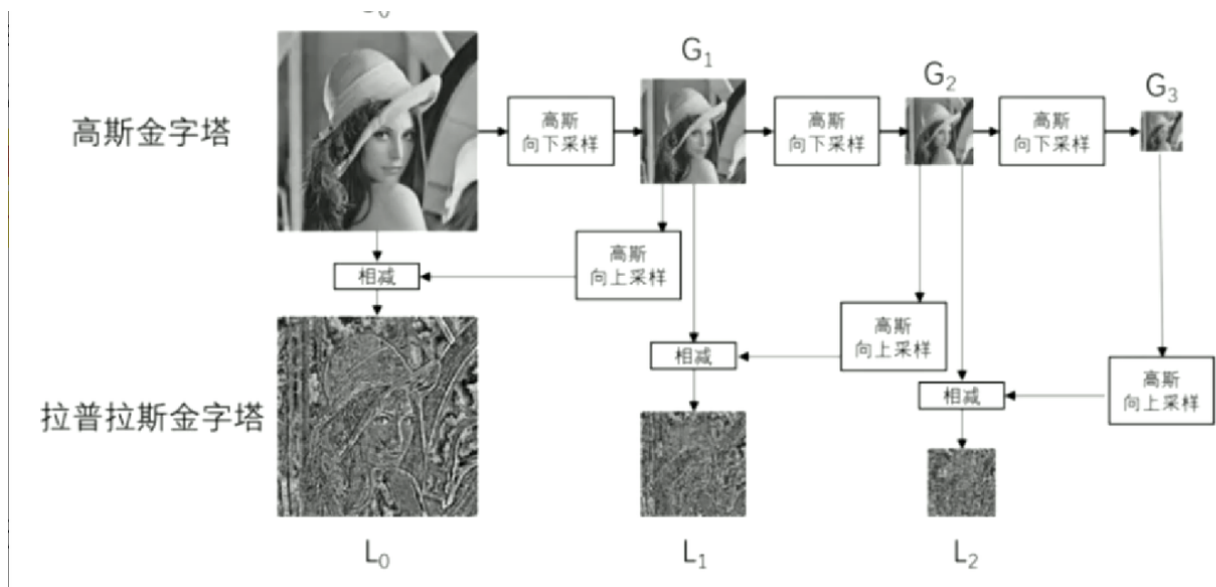
- c. 因此提出了两个方法：multi-scale joint sampling和 zoom stack representation
  - i. Zoom stack帮助我们在不同的level来渲染图片。image rendering保证了在不同的zoom level，都可以在overlap的地方实现一致性
    1. Image rendering：将图像进行下采样（降低分辨率），同时用zero-pad进行填充，然后利用高斯核进行迭代替换，这样可以保证图像的连续性



2. Noise image: 这里保证了语义的连续性, 因为本质上diffusion是VAE和DDPM的集合, 是DDPM在语义层面的扩散, VAE确保了语义过程的实现。这里就实现了和刚刚image rendering类似的操作, 将独立的噪声打包成zoom-consistant, 这个过程和image rendering中, 对zoom stack的打包类似, 同样有高斯核的过滤。

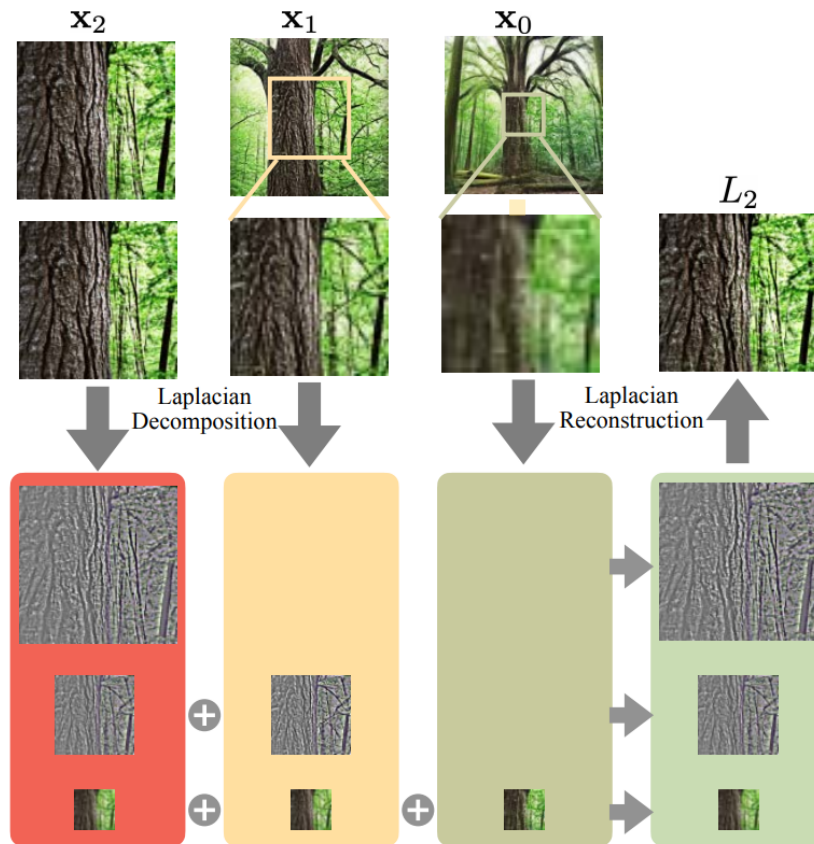
d. Multi-resolution blending:

- i. 把多个zoom level整合成一个stack来渲染
  - ii. 因为在不同的zoom level形成的图像不具有连续性, 一种方法就是直接朴素平均, 但是朴素平均会导致信息的丢失。
  - iii. 解决这个问题的方法是 multi resolution blending, 用拉普拉斯金字塔来融合不同的 frequency band (这里设计到一个知识, 图像金字塔)
1. 简单的下采样和上采样会导致信息的丢失, 图像金字塔有高斯金字塔和拉普拉斯金字塔。高斯金字塔就是简单的模糊并且下采样
  2. 拉普拉斯金字塔在高斯金字塔的基础上, 进行上采样, 同时用高斯金字塔的结果减去同一层, 就可以得到拉普拉斯金字塔 (关注的事差异)

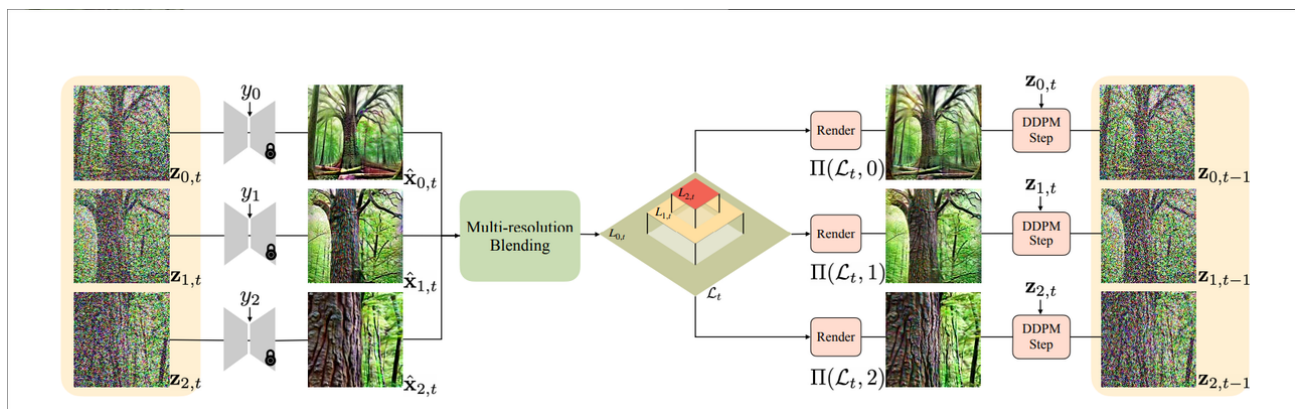


3. 拉普拉斯金字塔就可以保存每一层的信息, 在这里, 生成的每个照片已经有连续性, 而且存在分辨率大小的问题, 因此把每个图片都看做拉普拉斯金字塔的一层, 做模糊处理同时得到拉普拉斯金字塔, 就可以得到保留的信息, 这样不同层的信息就可以保留





4. 每个zoom level的image都形成自己的拉普拉斯金字塔，把细节进行相加，利用相加的细节在进行重建，来保持图像的一致性



- e. 整个过程的简要概述，噪声图片先用diffusion model来生成一系列的图片，然后对这些图片采用multi-resolution，也就是融合，融合得到的 zoom stack再次重新渲染，保证更好的一致性之后再重新用DDPM进行训练，这样DDPM的结果就可以保持一致性
- i. 这样是一个DDPM的一个步骤，相当于我在去噪声的时候，对每个噪声进行这样一步的处理
- ii. 等全部的噪声处理完，我就可以得到干净的图片

- 从当前zoom stack  $\mathcal{L}_t$  渲染出一致的噪声图像  $\mathbf{z}_{i,t}$  和噪声  $\epsilon_i$  (使用渲染函数  $\Pi_{\text{image}}$  和  $\Pi_{\text{noise}}$ , Algorithm 1: 这涉及下采样更高层并掩码融合，确保低频一致)。
- 将  $\mathbf{z}_{i,t}$  和对应提示  $y_i$  输入预训练扩散模型，预测噪声  $\hat{\epsilon}_{i,t-1}$ ，从而计算估计的干净图像  $\hat{\mathbf{x}}_{i,t-1}$ 。(此时，每个尺度的  $\hat{\mathbf{x}}_{i,t-1}$  可能在重叠区域不一致。)

f. 最终就可以完成这样的构造（不仅可以用文字生成，也可以直接用图片生成）

## 5. 疑问：

- a. 在noise rendering过程中，因为对zoom level存在下采样，DDPM又要求是高斯分布，是如何保证每次噪声都是高斯分布的
- b. 同样是在noise rendering过程中，noise image之间相互独立，是如何把他们打包成一个zoom-consistent noise。（文章中提到是和image rendering类似的操作，但是噪音是如何和图像做到一样的操作的）
- c. 我在image rendering 的时候已经做到了一致性，为什么后面还需要multi-resolution利用拉普拉斯金字塔来保持一致性

6. **和项目的关系：**将这个文章的想法和luciddreamer结合在一起，基本上可以很好的完成无限放大3D的效果。因为lucid利用text prompt可以预测出新图在三维空间的相机位置，而power of ten的思想，多重融合的角度可以让三维中的某一个视角完成超分。

- a. 问题的难点在于三维的一致性如何保持，也就是在zoom的过程中，luciddreamer生成的周边图像也依然需要保持一致性。（多视角的一致性）
- b. 但是已经可以解决固定视角的交互性问题，对于单视角来说，利用multi-resolution bending的思想，在生成新的超分图像的同时，利用luciddreamer确定新的点云的位置（有了新图像，可以确定新的深度图），那么这个点云就可以用作高斯渲染，完成单视角的固定
- c. 所以现在的难点就是在三维多视角下，保持一致性。

## LocalNeRF: Efficient Neural Radiance Fields with Local Sampling (NeurIPS 2023)

### 论文解读

主要提到了nerf的局部细节方法，用来增强3dgs的细节，因为目前阶段主要是想要实现3D的缩放效果，还没有考虑具体的细节优化，所以并没有想过多参考

## ZoeDepth: Zero-Shot Transfer by Combining Relative and Metric Depth (CVPR 2023)

### 论文解读

这个主要是luciddreamer中提到的，用来解决depth的论文，可以很好的处理遮挡关系，如果有细节可以在考虑根据这个来进行优化



