Instruct-NeRF2NeRF: Editing 3D Scenes with Instructions

- Instruct-NeRF2NeRF: Editing 3D Scenes with Instructions (arxiv.org)
- https://instruct-nerf2nerf.github.io/
- [v1] Wed, 22 Mar 2023 17:57:57 UTC (16,447 KB)

1 Introduction

- Unfortunately, while the pipelines for turning a real scene into a 3D representation are relatively mature and accessible, many of the other necessary tools for the creation of 3D assets (e.g., those needed for editing 3D scenes) remain underdeveloped
- we propose Instruct-NeRF2NeRF, a method for **editing 3D NeRF scenes** that requires as input only text instruction
- Though there exist 3D generative models, the datasources required for training these models at scale are still limited. Therefore, we instead choose to extract shape and appearance priors from a 2D diffusion model. Specifically, we employ a recent image-conditioned diffusion model InstructPix2Pix
- inconsistent edits across viewpoints: simple approach similar to recent 3D generation solutions like DreamFusion. Our underlying method, which we refer to as **Iterative Dataset Update (Iterative DU)**

2 Related Work

Physical Editing of NeRFs

- 相关方法 Most physically-based edits revolve around **changing physical properties** of the reconstructed scene, or **performing physical simulation**

Artistic Stylization of NeRFs

- 相关方法 works have **explored the use of latent representations** from visual language models
- 相关方法 To increase usability (and as explored in other 3D domains such as meshes), ClipNeRF and NeRF-Art extend this line of work **by encouraging similarity between CLIP embeddings of the scene and a short text prompt**.
- 困难 limitation of these CLIP-based approaches is their inability to incorporate localized edits

Generating 3D Content

- 相关方法 Recent progress in pre-trained large-scale models has enabled rapid progress in the domain of generating 3D content from scratch, **either by optimizing radiance fields through vision-language models** like CLIP or **via text-conditioned diffusion models**

- 困难 In all of these approaches, a central challenge is **congealing the inconsistent outputs of a 2D diffusion model into a consistent 3D scene**

  > 保持3D视角下图像的一致性最难解决

- For this paper: one advantage of editing an existing NeRF scene (as opposed to generating 3D content from scratch) is that the captured images are **by definition 3D consistent**, **suggesting that generated imagery should naturally be more consistent.**

Instruction as an Editing Interface

- 意义 We propose the first work that demonstrates instructional editing in the realm of 3D editing. This is particularly significant given the difficulty of the base task, 3D model editing, which typically requires specialized tools and years of experience.

3 Method

- inputs
  - a reconstructed NeRF scene along with its source data（下面几个）
  - a set of captured images
  - corresponding camera poses
  - camera calibration (typically from colmap)
  - a natural-language editing instruction
- outputs
  - an edited version of the NeRF subject to the provided edit instruction
  - edited versions of the input images

background

NeRF

太熟了，略

InstructPix2Pix

- InstructPix2Pix is a diffusion-based method specialized for image editing
- conditions
  - an RGB image c_I
  - a text-based editing instruction c_T
- inputs
  - a noised image or pure noise z_t

- outputs
    - an estimate of the edited image z_0
- method
    - predicts the amount of noise present in the input image z_t using the denoising U-Net ϵ_θ

$$\hat{\epsilon} = \epsilon_\theta(z_t;\ t, c_I, c_T)$$

    - $\hat{\epsilon}$ can be used to derive $\hat{z}_0$, the estimate of the edited image
    - This denoising process can be queried with a noisy image z_t at any timestep t, i.e., containing any amount of noise, up to a pure noise image
    - Larger amounts of noise, i.e., larger values of t, will produce estimates of $\hat{z}_0$ with more variance
    - In practice, the diffusion process operates **entirely on an encoded latent domain**. This means that the abovedefined variables c_I , z_0 are all latent images created by encoding an RGB image $\mathcal{E}(I)$. Similarly, to produce an RGB image from the diffusion model, one must also decode the predicted latents $\hat{z}_0$ via the decoder $\hat{I} = \mathcal{D}(\hat{z}_0)$.

Instruct-NeRF2NeRF

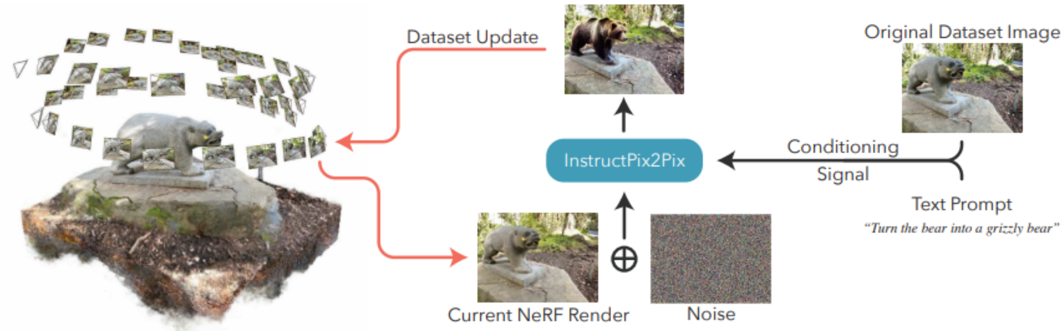


Figure 2: **Overview**: Our method gradually updates a reconstructed NeRF scene by iteratively updating the dataset images while training the NeRF: (1) an image is rendered from the scene at a training viewpoint, (2) it is edited by InstructPix2Pix given a global text instruction, (3) the training dataset image is replaced with the edited image, and (4) the NeRF continues training as usual.

> Our method works through an alternating update scheme, in which the training dataset images are iteratively updated using a diffusion model and are subsequently consolidated into the globally consistent 3D representation by training the NeRF on these updated images.

edit one single image

- using InstructPix2Pix
- input（InstructPix2Pix的conditions+input，一样的）
    - an conditioning image c_I （图中conditioning signal的图）
        - instantiated as the unedited image I_0^v at viewpoint v or a render from the NeRF

- a text-based editing instruction c_T （图中conditioning signal的指令）
- a noisy input z_t （图中异或后的结果）
    - instantiated as a noised version of the current render at optimization step i, that is, a linear combination of $\mathcal{N}(0,1)$ and $z_0 = \mathcal{E}(I_i^v)$
    - denoted as $I_{i+1}^v \leftarrow U_\theta(I_{i+1}^v, t;\ I_0^v, c_T)$ where a noise level t is randomly chosen from a constant range $[t_{\min}, t_{\max}]$ and U_θ is a DDIM sampling process, with a fixed number of intermediate steps s taken between initial timestep t and 0

> A key thing to note is that while our method continually repeats the process of rendering from the NeRF, editing the image, and updating the NeRF, the diffusion model is conditioned on the *un-edited* images, and thus remains grounded, **preventing the characteristic drift commonly associated with recurrent synthesis**

iterative dataset update

- process
    - At the beginning, our image dataset consists of the originally captured images from a range of viewpoints denoted as v, which we represent as I_0^v
        - original images are cached separately and **used as conditioning for the diffusion model at all stages**
    - At each iteration, we perform a number of image updates d, followed by a number of NeRF updates n
        - Image updates are performed **sequentially in a random ordering** of v determined at the start of training
        - NeRF updates always **sample a set of random rays from the entire training dataset**, such that the supervision signal contains a mixture of *old* information and *new* pixels from recently updated dataset images
- At early iterations, these images may perform inconsistent edits (as InstructPix2Pix does not typically perform consistent edits across different viewpoints). Over time, as images are used to update the NeRF and progressively re-rendered and updated, they begin to converge on a globally consistent depiction of the edited scene

    > 在训练的初期，出现3D不一致和漂流是正常的，这种现象在InstructPix2Pix中也会出现
    >
    > 但是随着图像更新和3D一致性的作用，会收敛

details

- use Nerfacto
- use L1 and LPIPS losses

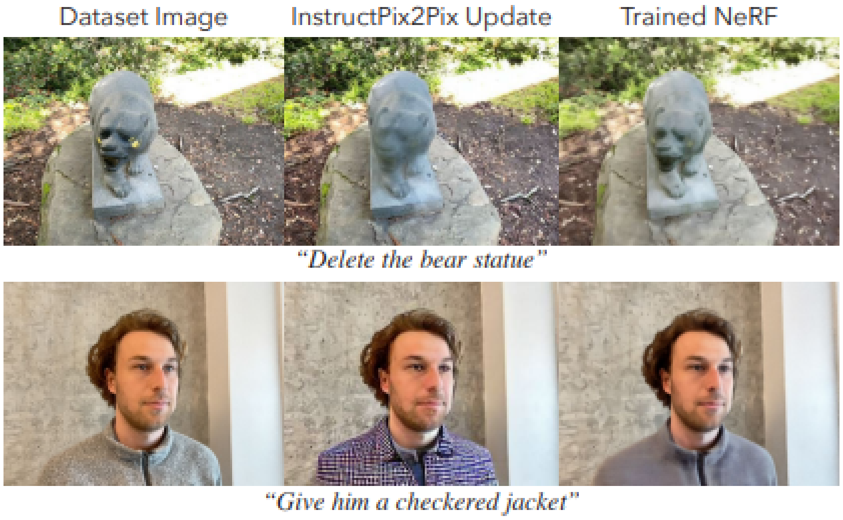
- Our process of optimizing for an edited NeRF uses a diffusion model as guidance, which can produce a collection of temporally varying images (i.e., varying over the course of optimization). As a result, the optimization process does not have a single convergence point, as standard NeRF optimization does, where all images are sufficiently well explained by the reconstructed model. Therefore, the edited NeRF also varies in type and strength of edit over the course of optimization, and one must select an iteration at which to terminate optimization and visualize the edited scene.

> 也就是说，由于随机性、多个收敛点和漂流的存在，需要使用者随时观察渲染结果，当渲染结果理想时，就要手动结束训练了

4 Evaluation

limitations

- inherits many of the limitations of InstructPix2Pix, such as the inability to perform large spatial manipulations
    - the inability to perform large spatial manipulations
    - the occasional inability to perform binding between objects referred to in the instruction and the corresponding scene objects
    - adding or removing large objects.
- as in DreamFusion, our method uses a diffusion model on a single view at a time, and thus our method may suffer from similar artifacts
- it’s worth noting that the edit instructions provided to InstructPix2Pix are sometimes more relevant to certain views than others



“Delete the bear statue”

“Give him a checkered jacket”

Figure 9: **Limitations:** InstructPix2Pix cannot always perform the desired edit (top), and thus our method does not perform an edit. Sometimes InstructPix2Pix produces correct, but inconsistent edits in 2D that our method fails to consolidate in 3D (bottom).

> 1. 3D一致性
> 2. 编辑能力基本完全受限于InstructPix2Pix
> 3. 虚影，这在demo中的蝙蝠侠部分也可以看到

CLIP: Learning Transferable Visual Models From Natural Language Supervision

- [2103.00020] Learning Transferable Visual Models From Natural Language Supervision (arxiv.org)
- 学习自然语言而不是学习预定义的标签，从自然语言中学习有两点优势：
  - 使用自然语言学习的方法可以从互联网上大量的文本数据中学习；
  - 与大多数无监督或自监督的学习方法相比，从自然语言中学习不只是学习一个表征，而且还将该表征与语言联系起来，从而实现灵活的zero-shot learning（自然语言的信息、泛化要比图像的好）
- 一个图像可以对应很多个caption文本，为了提升效率，不再去预测文本，而是去预测文本和图像是否属于一个pair
  - 是否匹配的方法很简单，通过点乘计算匹配程度即可，最后希望接近一个单位矩阵（对角线上的image-caption是匹配的）
- 与NeRF结合：基本上是利用CLIP检测NeRF渲染出的图像与caption是否匹配，并由此指导优化、编辑radiance fields
- **核心：检查image和caption的匹配**

SDS: Score Distillation Sampling

- [2209.14988] DreamFusion: Text-to-3D using 2D Diffusion (arxiv.org)
- > SDS把扩散模型的*预测噪声的残差与梯度*联系起来，使得在parameter space上进行扩散，而不是image pixel space上扩散（），每一步预测的噪声与图像上的梯度有一定联系。
- > 蒸馏，就是知识蒸馏，将教师网络(teacher network)的知识迁移到学生网络(student network)上，使得学生网络的性能表现如教师网络一般。

```
z_t = random.normal(img_shape)
for t in linspace(tmax, tmin, nstep):
    epshat_t = diffusion_model.epshat(z_t, y, t) # Score function evaluation.
    if t > tmin:
        eps = random.normal(img_shape)
        z_t = ddpm_update(z_t, epshat_t, eps) # 1 iteration, decreases noise level.
x = diffusion_model.xhat(z_t, epshat_t, t_min) # Tweedie’s formula: denoise the
laststep.
return x
```

DDPM使用的反向过程（原论文把反向过程称为采样sampling）

1. 获取一个随机噪声（反向过程的开始）
2. 获取时间序列
3. 让diffusion估计噪声（epsilon hat: $\hat{\epsilon}$）
4.
5. 获取一个随机噪声
6. 反向过程中的降低噪声，这里其实就看得出来是在image pixel space上更新了，z_t就是一个图像的大小（反向过程的每一次降低也要添加噪声）
7. 反向过程的最后一步

```
params = generator.init()
opt_state = optimizer.init(params)
diffusion_model = diffusion.load_model()
for nstep in iterations:
    t = random.uniform(0., 1.)
    alpha_t, sigma_t = diffusion_model.get_coeffs(t)
    eps = random.normal(img_shape)
    x = generator(params, <other arguments>...) # Get an image observation.
    z_t = alpha_t * x + sigma_t * eps # Diffuse observation.
    epshat_t = diffusion_model.epshat(z_t, y, t) # Score function evaluation.
    g = grad(weight(t) * dot(stopgradient[epshat_t - eps], x), params)
    params, opt_state = optimizer.update(g, opt_state) # Update params with optimizer.
return params
```

SDS使用的反向过程：

5. 随机获取一个时间
6. 根据时间，获取这个时间对应的系数
7. 获取一个随机噪声
8. 让生成器生成一张图片
9. 获取隐变量，相当于一步模拟了DDPM的前向过程
10. 让diffusion估计噪声（epsilon hat: $\hat{\epsilon}$）
11. 计算梯度，代码中的实现是$g = matmul(weight(t) * (-epshatt - eps), grad(x, params))$
12. 更新，梯度下降

reference

- InstructPix2Pix: 一种无需微调新的快速图像编辑方法 - 知乎 (zhihu.com)
- AIGC的可控图像编辑 - 知乎 (zhihu.com)
- Nerfacto - nerfstudio

- CLIP Learning Transferable Visual Models From Natural Language Supervision (arxiv.org)
    - 为什么Clip可以用于zero shot分类？ - 知乎 (zhihu.com)
    - 【多模态】CLIP模型 - 知乎 (zhihu.com)
    - [论文精读] ICML 2021 | Learning Transferable Visual Models From Natural Language Supervision - 知乎 (zhihu.com)
    - 【论文笔记】CLIP：Learning Transferable Visual Models From Natural Language Supervision - 知乎 (zhihu.com)
- NeRF与CLIP的默契交锋——CLIP-NeRF, DreamFields论文分享 - 知乎 (zhihu.com)
- Stable Diffusion-Unet浅谈 - 知乎 (zhihu.com)
- SDS - Score Distillation Sampling
- DDIM
    - 扩散模型之DDPM - 知乎 (zhihu.com)
    - 扩散模型之DDIM - 知乎 (zhihu.com)
    - 一文读懂DDIM凭什么可以加速DDPM的采样效率 - 知乎 (zhihu.com)
    - 一文解决你关于扩散模型ddpm的所有疑惑 - 知乎 (zhihu.com)
    - 十分钟读懂Diffusion：图解Diffusion扩散模型 - 知乎 (zhihu.com)
    - DDPM原理与代码剖析_Andy Dennis的博客-CSDN博客
    - DDIM是一个确定的隐变量模型，也是DDIM名字的由来。所以DDIM是包含了DDPM的结果，将Diffusion变得更加一般化。
    - 模型的输入是x_t, t，输出是对应的高斯噪声ϵ_θ