基于神经表示的三维重建与生成

Fast Inference & Training and 3D-Aware Generative Models

廖依伊

VALSE 2022 Tutorial

Outline

- Real-time rendering of neural radiance fields
- Fast training of neural scene representation
- 3D-aware generative model

rendered in real-time on NVIDIA GTX 1080 Ti*

*affordable consumer GPU

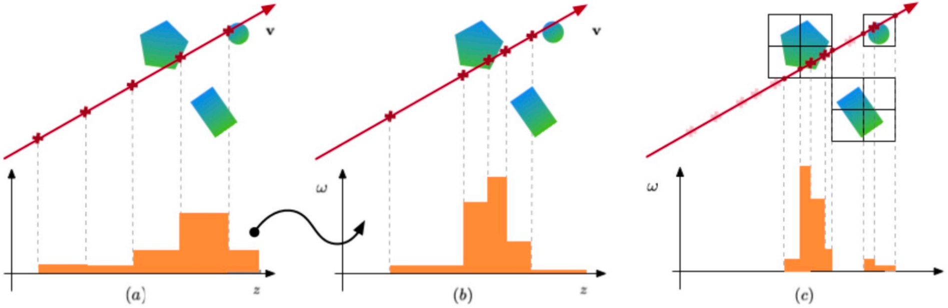
Fast Inference

Key Idea: Reduce FLOPs of volume rendering

FLOPs: $H \times W \times K \times L_r$ K: Samples per ray, L_r: FLOPs of color/density retrieval

- **Reducing K**
  - Classical techniques: Early ray termination, empty space skipping
  - Adaptive sampling
- **Reducing L_r**
  - Tabulation-based methods
  - Smaller networks

Reducing K: Empty space skipping & Early ray termination

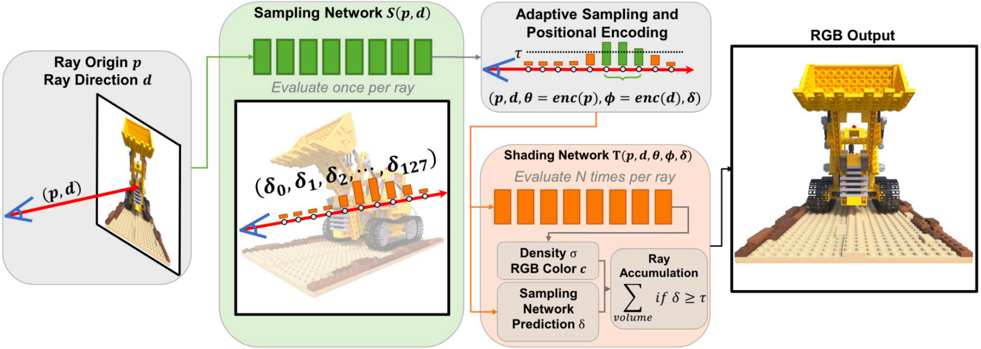


(a)uniform sampling (b) importance sampling (c) ESS sampling

- ESS: Skipping a sample point $\mathbf{x}$ if $\mathbf{x}$ lies in an unoccupied cell
- ERT: Skipping $\mathbf{x}_{i+1}, \mathbf{x}_{i+2}$ if transmittance $T_i < \epsilon$

Liu, Gu, Lin, Chua, Theobalt. Neural Sparse Voxel Fields. NeurIPS 2020.

Reducing K: Adaptive sampling



- Learning where to sample via a **sampling network**
- Fine-tune shading network to desired sample counts (2,4,8,16) for fast rendering

Kurz*, Neff*, Lv, Zollhöfer, Steinberger. AdaNeRF: Adaptive Sampling for Real-time Rendering of Neural Radiance Fields. ECCV 2022.

Reducing L_r: Tabulation-based Methods

Naïve Solution:

1. Train a large MLP $f(\mathbf{x}, \mathbf{d})$
2. save network's output for fast inference

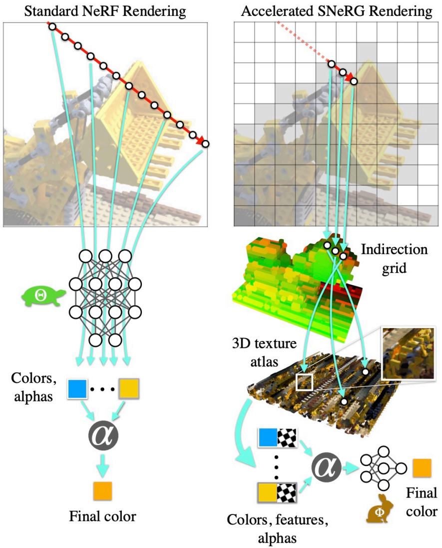
Challenge: Naïve solution requires memory of $O(N^5)$ given 5D input $(\mathbf{x}, \mathbf{d})$

Key Idea: Modeling **view dependency** $\mathbf{d}$ differently to reduce memory to $O(N^3)$

- SNeRG, ICCV 2021
- PlenOctree, ICCV 2021
- FastNeRF, ICCV 2021

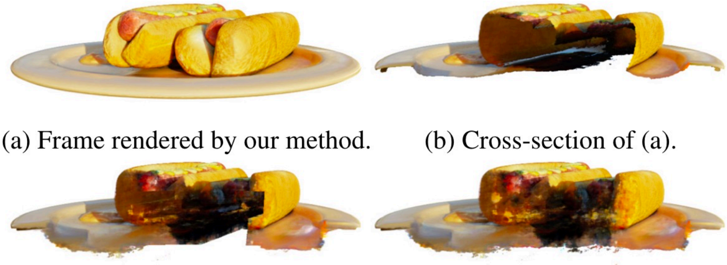
SNeRG

- Replacing slow MLP evaluations with lookups into **cached sparse 3D grid**
- Caching **view-independent** colors, features and alphas
- Model **view dependency** via a small network
- Combine with ESS + ERT



Hedman, Srinivasan, Mildenhall, Barron, Debevec. Baking neural radiance fields for real-time view synthesis. ICCV, 2021.

SNeRG



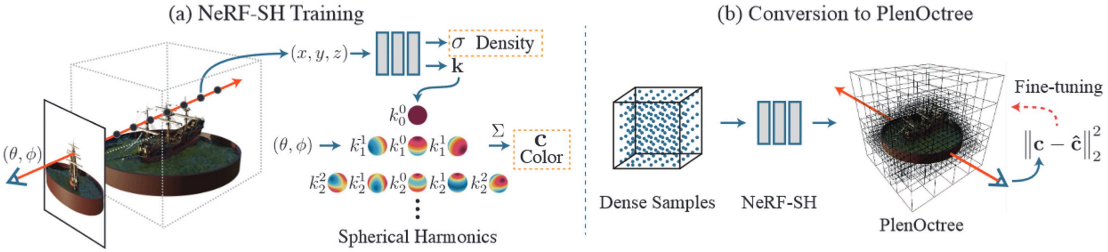
(a) Frame rendered by our method. (b) Cross-section of (a).

(c) Trained without $\mathcal{L}_S$. (d) No $\mathcal{L}_S$, no visibility culling.

- $O(N^3)$ still requires **large memory consumption**
- Culling voxels where the maximum opacity is low, or maximum transmittance $T < \epsilon$ across all views

Hedman, Srinivasan, Mildenhall, Barron, Debevec. Baking neural radiance fields for real-time view synthesis. ICCV, 2021.

PlenOctree



- Modeling view dependency via **spherical harmonics coefficients** $\mathbf{k}$
- Use octree structure to skip large empty space, but still **memory costly**

Yu, Li, Tancik, Li, Ng, Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. ICCV, 2021.

Reducing L_r: Smaller networks

Naïve Solution: Replace the large MLP with a small MLP

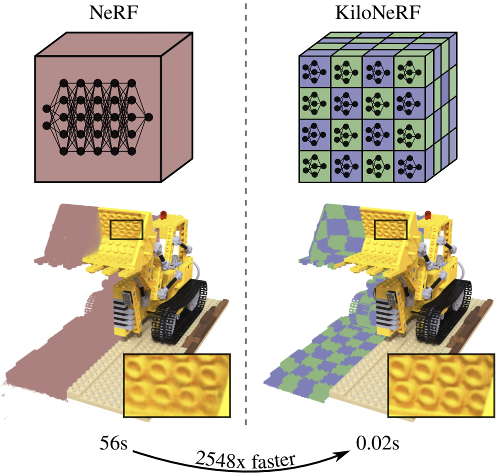
Challenge: Naïve solution leads to degraded image quality

Key Idea: Use a small network to **independently** represent a small region

- KiloNeRF: ICCV 2021
- (BlockNeRF, MegaNeRF)

KiloNeRF

- Replacing **large** MLPs with **small** ones via space partitioning
- No cache required, more **memory friendly**
- Combine with ESS + ERT



Reiser, Peng, Liao and Geiger: KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs. ICCV, 2021.

KiloNeRF

Method	Render time ↓	Speedup ↑
NeRF	56185 ms	–
NeRF + ESS + ERT	788 ms	71
KiloNeRF	**22** ms	**2548**

Table 2: **Speedup Breakdown.** The original NeRF model combined with KiloNeRF's implementation of ESS and ERT is compared against the full KiloNeRF technique.

- ESS/ERT and small MLPs both contribute to fast rendering

Fast Inference

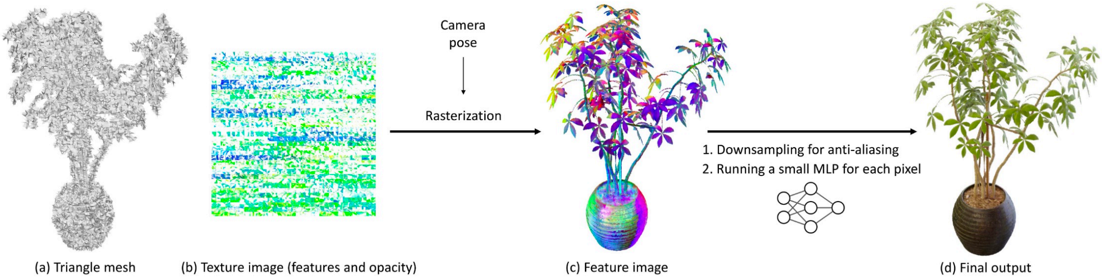
Key Idea: Reduce FLOPs of volume rendering $\rightarrow$ image-order rendering

FLOPs: $H \times W \times K \times L_r$ $\quad K$: Samples per ray, L_r: FLOPs of color/density retrieval

- Reducing K
  - Classical techniques: Early ray termination, empty space skipping
  - Adaptive sampling
- Reducing L_r
  - Tabulation-based methods
  - Smaller networks

How about object-order rendering?

MobileNeRF



- ▶ Exploit fast polygon **rasterization**, using standard GPU rasterization pipeline
- ▶ Avoid semi-transparency, enabling efficient rendering without sorting
- ▶ 10$\times$ faster than SNeRG

Chen et al. MobileNeRF: Exploiting the Polygon Rasterization Pipeline for Efficient Neural Field Rendering on Mobile Architectures. Arxiv 2022.

Discussions

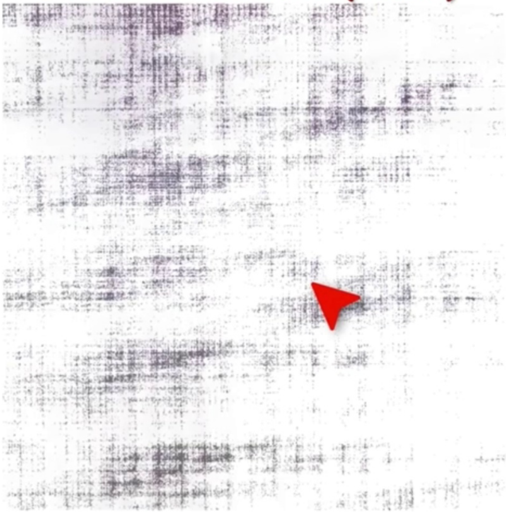
Method	Device	Dataset	FPS	Memory (MB)	PSNR
AdaNeRF*	NVIDIA 3090	DONeRF Dataset (800x800)	20	4	27.5
SNeRG	Laptop with mobile GPU	NeRF Synthetic (800x800)	207.26	4096 (100)	30.38
FastNeRF	NVIDIA 3090		238	16200	29.97
PlenOctree	NVIDIA V100		167.68	1930	31.71
KiloNeRF	NVIDIA 1080TI		50	50	31
MobileNeRF	NVIDIA 2080TI		744.91	125.75	30.9
	iPhone XS		55.89		

- Fast rendering is usually achieved by combining **multiple techniques**
- Image-order: tabulation-based methods are more **efficient** at the cost of **memory**
- Object-order: leverage standard rendering tools

Outline

- Real-time rendering of neural radiance fields
- Fast training of neural scene representation
- 3D-aware generative model

Previous method (NeRF)



00:04
minutes : seconds

Source: [Müller et al. 2022][Keil et al. 2021]

Fast Training

Key Idea: Reduce FLOPs of volume rendering

FLOPs: $H \times W \times K \times L_r$ K: Samples per ray, L_r: FLOPs of color/density retrival

- **Reducing K**
  - Empty space skipping
- **Reducing L_r**
  - Directly optimize $O(N^3)$ voxel grid or optimize $O(N^3)$ local feature + small MLP

Fast Training

Key Idea: Reduce FLOPs of volume rendering

FLOPs: $H \times W \times K \times L_r$ K: Samples per ray, L_r: FLOPs of color/density retrival

- **Reducing K**
  - Empty space skipping
- **Reducing L_r**
  - Directly optimize $O(N^3)$ voxel grid or optimize $O(N^3)$ local feature + small MLP

Additional benefits of optimizing local tensors?

- Each voxel or feature tensor is only responsible for a small region
- Shallow computational graph, allowing for use large learning rate

Fast Training

Challenge: Memory complexity of $O(N^3)$ still **expensive**

Idea: Reduce required resolution or use compact representation

- Reduce required resolution via activation design
  - DVGO
- Explore sparsity
  - Plenoxel: Sparse data structure
- Parameter sharing
  - Instant NGP: hard parameter-sharing
  - TensoRF: tensor factorization

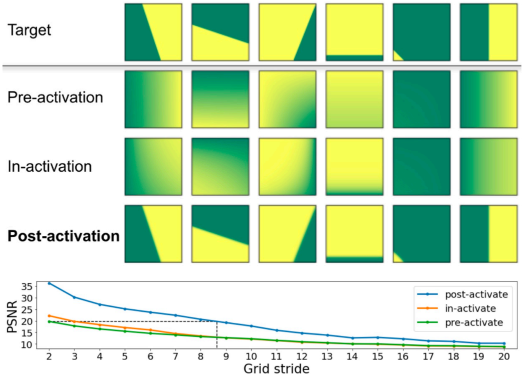
DVGO

- **Post-activation**: modeling sharp surface within a cell
- Resolution $160 \times 160 \times 160$ for all NeRF-Synthetic objects

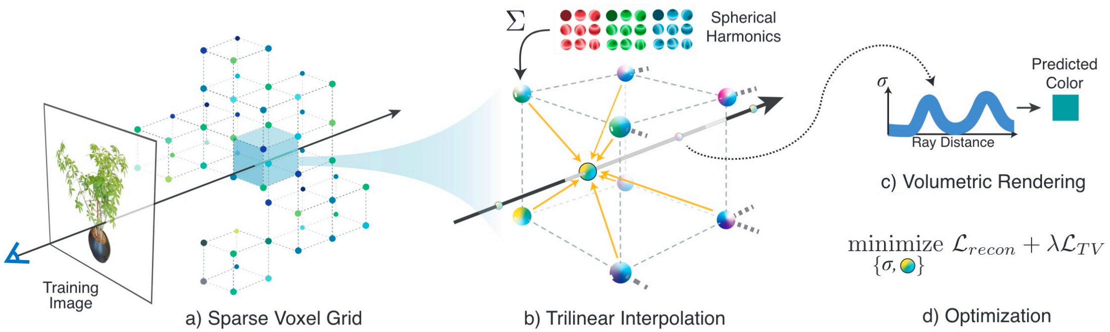
$$\alpha^{(\text{pre})} = \text{interp}\left(\mathbf{x}, \text{alpha}\left(\text{softplus}\left(\mathbf{V}^{(\text{density})}\right)\right)\right),$$

$$\alpha^{(\text{in})} = \text{alpha}\left(\text{interp}\left(\mathbf{x}, \text{softplus}\left(\mathbf{V}^{(\text{density})}\right)\right)\right),$$

$$\alpha^{(\text{post})} = \text{alpha}\left(\text{softplus}\left(\text{interp}\left(\mathbf{x}, \mathbf{V}^{(\text{density})}\right)\right)\right).$$



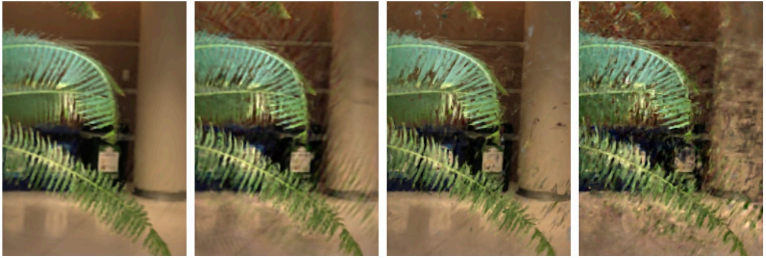
Plenoxel



- **Sparse data structure** using progressive training
- Higest resolution $512 \times 512 \times 512$ for NeRF-Synthetic objects

Yu*, Fridovich-Keil*, Tancik, Chen, Recht, Kanazawa. Plenoxels: Radiance Fields without Neural Networks. CVPR 2022.

DVGO & Plenoxel

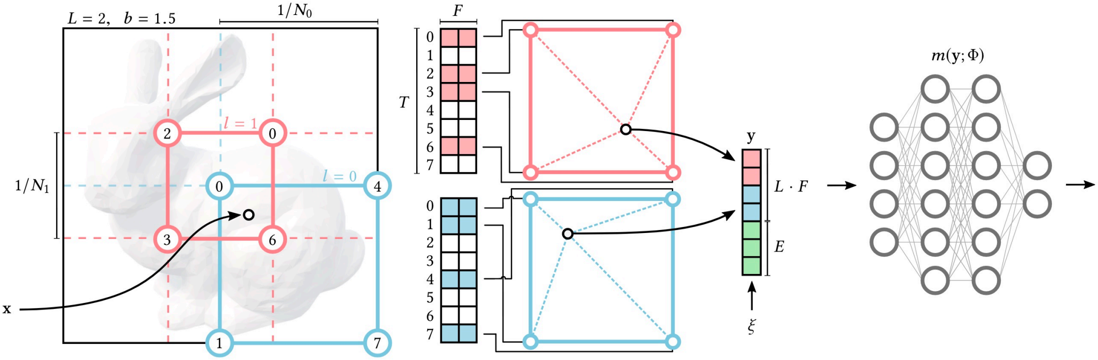


Full | No SH TV | No σ TV | No TV

Regularization: both optimize density voxel grid directly, additional ”tricks” required

- DVGO: low density initialization, lower learning rate to voxels visible to fewer views
- Plenoxel: total variation loss, (sparsity loss for forward-facing scenes)

Instant NGP

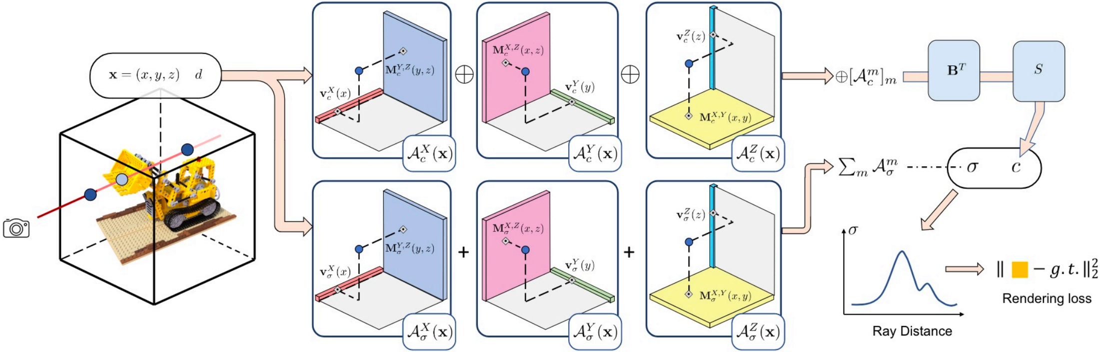


(1) Hashing of voxel vertices **(2)** Lookup **(3)** Linear interpolation **(4)** Concatenation **(5)** Neural network

- Multi-resolution feature grid saved in hash table + small MLP
- Random **Parameter sharing** by not resolving hash conflition

Müller, Evans, Schied, Keller. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. SIGGRAPH 2022.

TensoRF



- Feature grid represented by factorized tensor + small MLP
- **Parameter sharing** along a certian axis / plane

Chen*, Xu*, Geiger, Yu, Su. TensoRF: Tensorial Radiance Fields. ECCV 2022.

Discussions

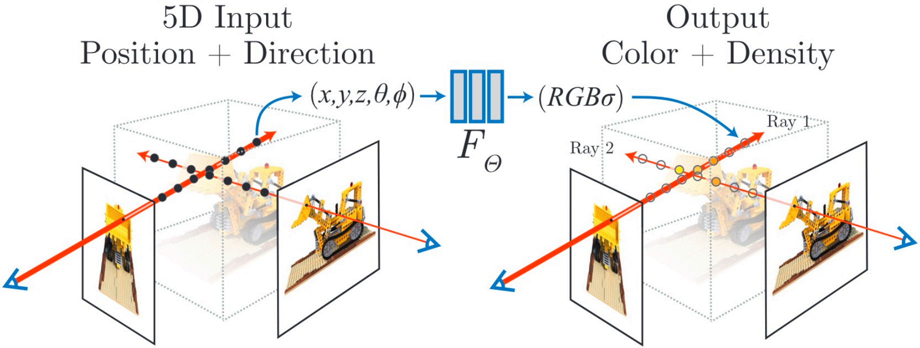
Method	Device	Training Time	Memory (MB)	PSNR	Initial LR
NeRF	NVIDIA V100	1-2 days	6MB	31.01	5e-4
DVGO	NVIDIA RTX2080 Ti	15 mins	612.1	31.95	0.1
Plenoxel	NVIDIA Titan RTX	11 mins	778.1	31.71	30
Instant NGP	NVIDIA RTX 3090	**5 mins**	**50.4**	**33.18**	0.01
Instant NGP's NeRF	NVIDIA RTX 3090	**5 mins**	–	30.06	–
TensoRF	NVIDIA V100	17 mins	71.8	**33.14**	0.02

- Parameter sharing methods are more **memory efficient**
- Larger **learning rates** of local tensors lead to faster convergence
- Good **Implementation** is another key to fast training, e.g., Instant NGP's NeRF
- Empty space skipping is also commonly used

Outline

- Real-time rendering of neural radiance fields
- Fast training of neural scene representation
- 3D-aware generative model

Why 3D-Aware Generative Models?



- NeRF optimizes the MLP for a **single scene**
- Not easy to create non-existed scenes

Why 3D-Aware Generative Models?

- Existing 2D GANs are able to generate high fidelity, novel contents
- However, there is no **3D controllability**, e.g., control over camera poses

3D-Aware GANs

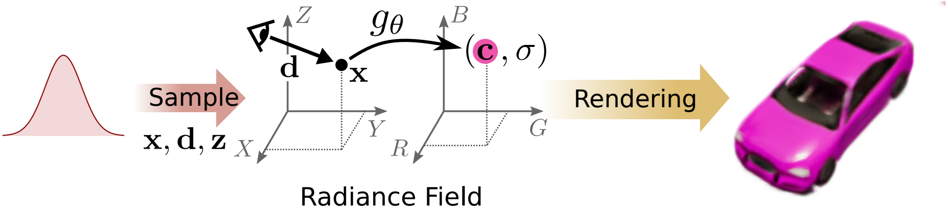
Naïve Solution: Replace the 2D generator as a conditional radiance field

Challenge: Image-level supervision, **no per-pixel loss**; computationally expensive

Idea: Reducing FLOPs of one forward pass of the generator $H \times W \times K \times L_r$

- **Reducing $H \times W$**
  - Patch-based discriminator
  - 2D-CNN based upsampling network
- **Reducing K**
  - Sampling on a few isosurfaces
  - Empty space skipping
- **Reducing L_r**
  - Replace (a part of) large MLP with 2D/3D CNN

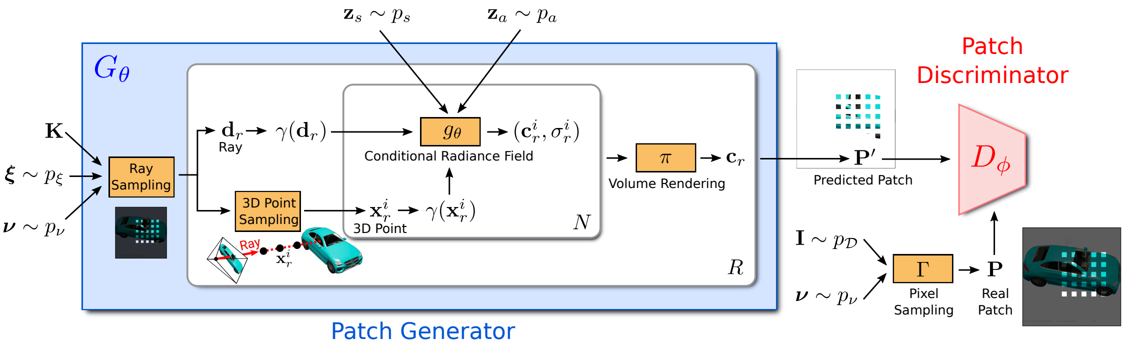
GRAF: Generative Radiance Fields



- **Generative model** for radiance fields
- Trained from **unstructured** and **unposed** 2D image collections

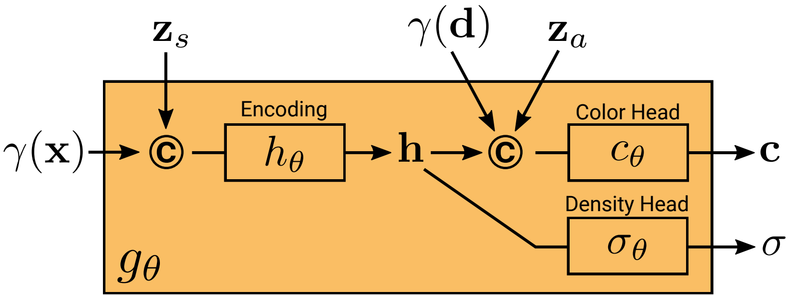
Schwarz*, Liao*, Niemeyer, Geiger: GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. NeurIPS, 2020

Reducing $H \times W$: Patch-based discriminator



- **Conditional** neural radiance fields supervised by **adversarial loss**
- Sample **camera poses** and **image patches** of size 32×32 pixels

GRAF: Generator



- $\mathbf{z}_s$ for shape， $\mathbf{z}_a$ for appearance
- Automatically disentangled $\mathbf{z}_s$ and $\mathbf{z}_a$

RGB

Depth

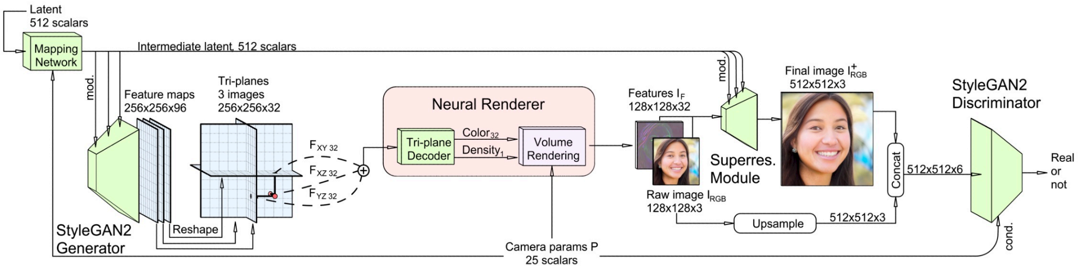
256x256

Reducing $H \times W$: StyleNeRF

- Low-res. volume rendering + **2D upsampling neural renderer**
- High fidelity, harder to preserve multi-view consistency

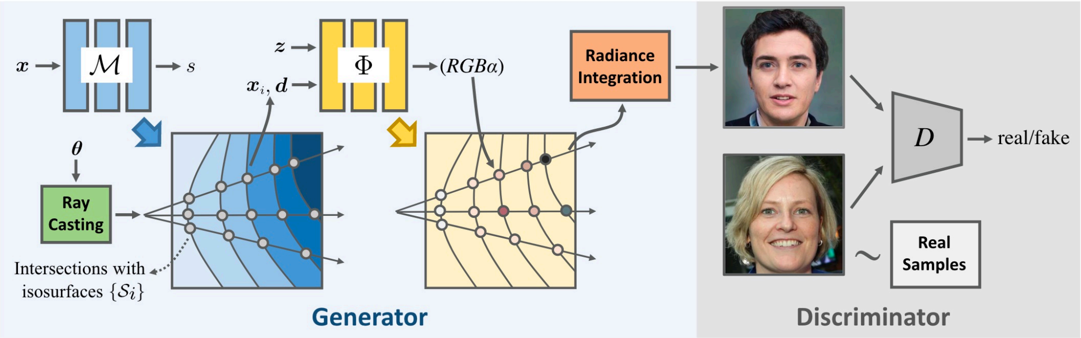
Gu, Liu, Wang, Theobalt. StyleNeRF: A Style-based 3D-Aware Generator for High-resolution Image Synthesis. ICLR 2022.

Reducing $H \times W$ and L_r: EG3D



- L_r: Replace per-point large MLP with tri-plane 2D generator + small MLP
- $H \times W$: Uses 2D upsampling neural renderer

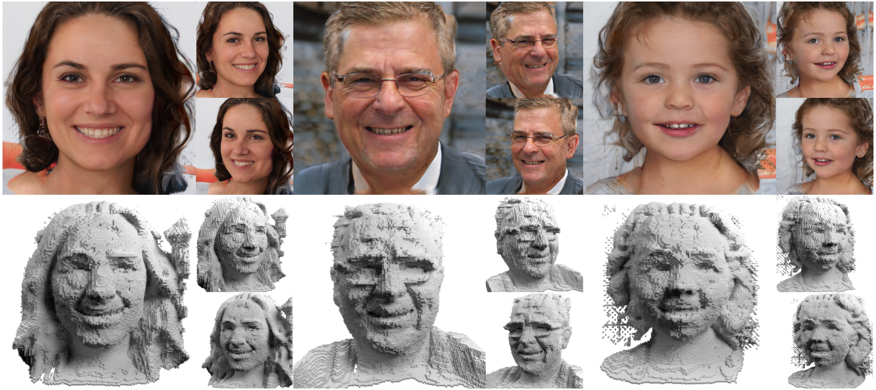
Reducing K: GRAM



- No 2D CNN, evaluate on 24 or 48 isosurfaces

Deng, Yang, Xiang, Tong. GRAM: Generative Radiance Manifolds for 3D-Aware Image Generation. CVPR 2022.

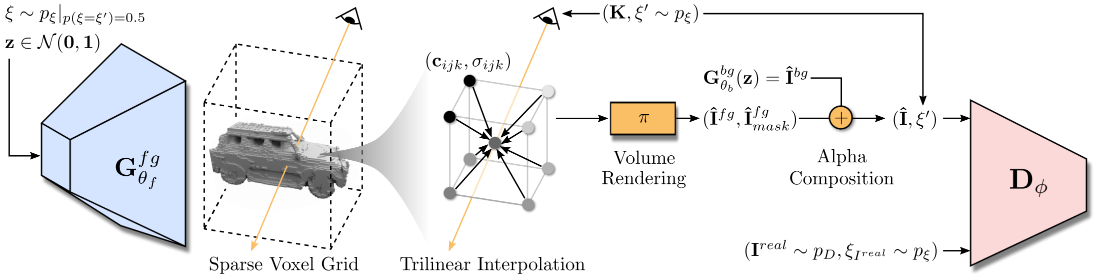
Reducing K and L_r: VoxGRAF



- Inspired by DVGO and Plenoxel, represent scene as 3D **sparse** voxel grids
- Fast rendering during inference， one forward pass to generate the voxel grids

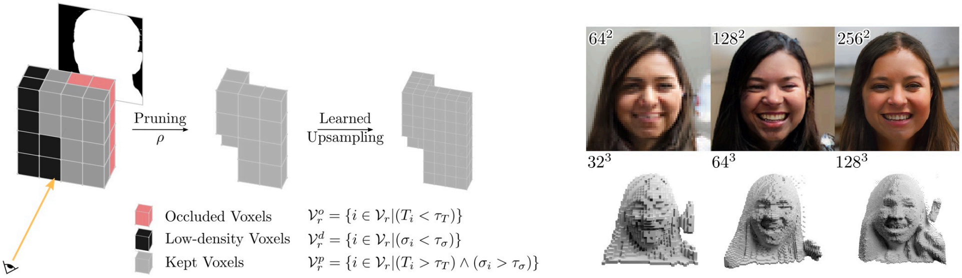
Schwarz, Sauer, Niemeyer, Liao, Geiger. VoxGRAF: Fast 3D-Aware Image Synthesis with Sparse Voxel Grids. Arxiv 2022

VoxGRAF



- Learn sharp surface via regularization
- Disentangle foreground and background

VoxGRAF



(a) Pruning (b) Progressive Growing

- Density based pruning
- Progressive Growing for the resolution of voxel grids and 2D image

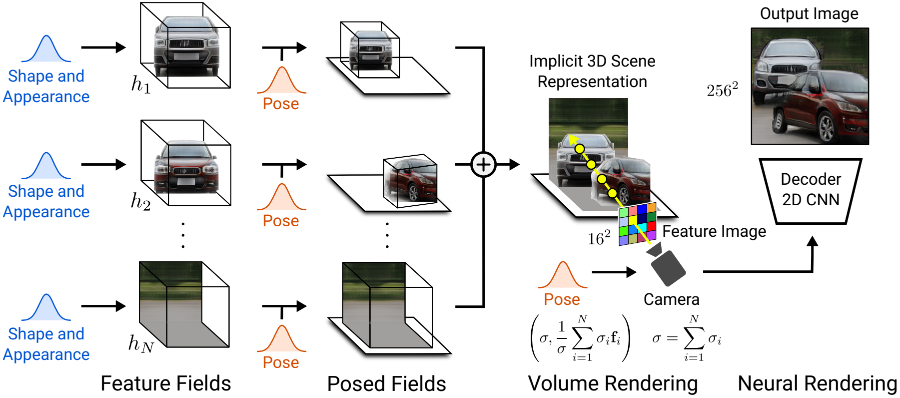
VoxGRAF

Schwarz, Sauer, Niemeyer, Liao, Geiger. VoxGRAF: Fast 3D-Aware Image Synthesis with Sparse Voxel Grids. Arxiv 2022

VoxGRAF

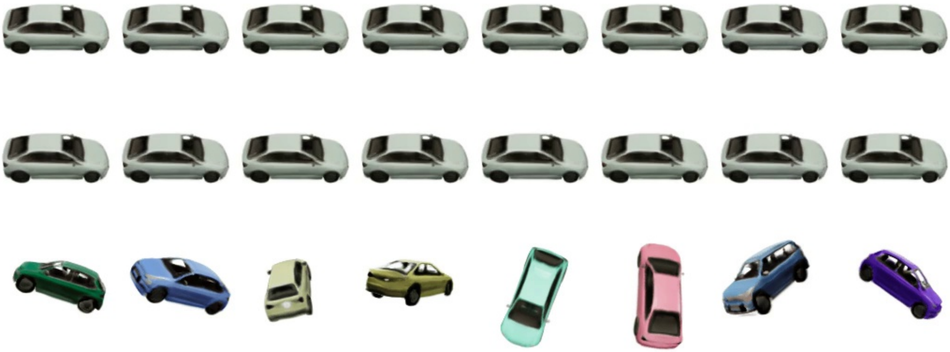
Schwarz, Sauer, Niemeyer, Liao, Geiger. VoxGRAF: Fast 3D-Aware Image Synthesis with Sparse Voxel Grids. Arxiv 2022

What's more? GIRAFFE: Compositional Generation



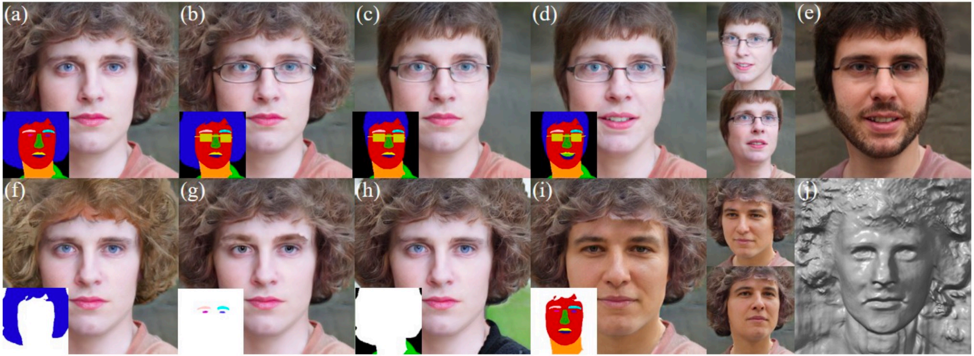
Niemeyer and Geiger: GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields. CVPR, 2021

What's more? Category-level 6DoF Pose Estimation



Guo, Zhong, Xiong, Liu, Wang, Liao. A Visual Navigation Perspective for Category-Level Object Pose Estimation. ECCV 2022.

What's more? Semantic Editing



Sun, Wang, Shi, Wang, Wang, Liu. IDE-3D: Interactive Disentangled Editing for High-Resolution 3D-aware Portrait Synthesis. Arxiv 2022.

Conclusions

- 3D-aware generative methods share similar ideas to fast inference and training
- 2D neural renderer leads to better FID
- Many exciting applications

	FFHQ $R_I = 256^2$	AFHQ $R_I = 256^2$	Carla $R_I = 128^2$
GIRAFFE	31.5	16.1	–
StyleNeRF	8.0	–	–
EG3D	**4.8**	**3.9**	–
GRAF	71	121	41
π-GAN	85	47	29.2
GOF	69.2	54.1	29.3
GRAM	17.9	18.5	26.3
VoxGRAF	**14.4**	**9.6**	**11.3**

Thank you!

https://yiyiliao.github.io/

https://zju3dv.github.io/inr_tutorial/