Learning Accurate and Parsimonious Point Cloud Representations from Images

Wonseok Oh | December 8, 2023
EECS 453 Principals of Machine Learning | University of Michigan

Contents

Introduction

Motivation

Methods

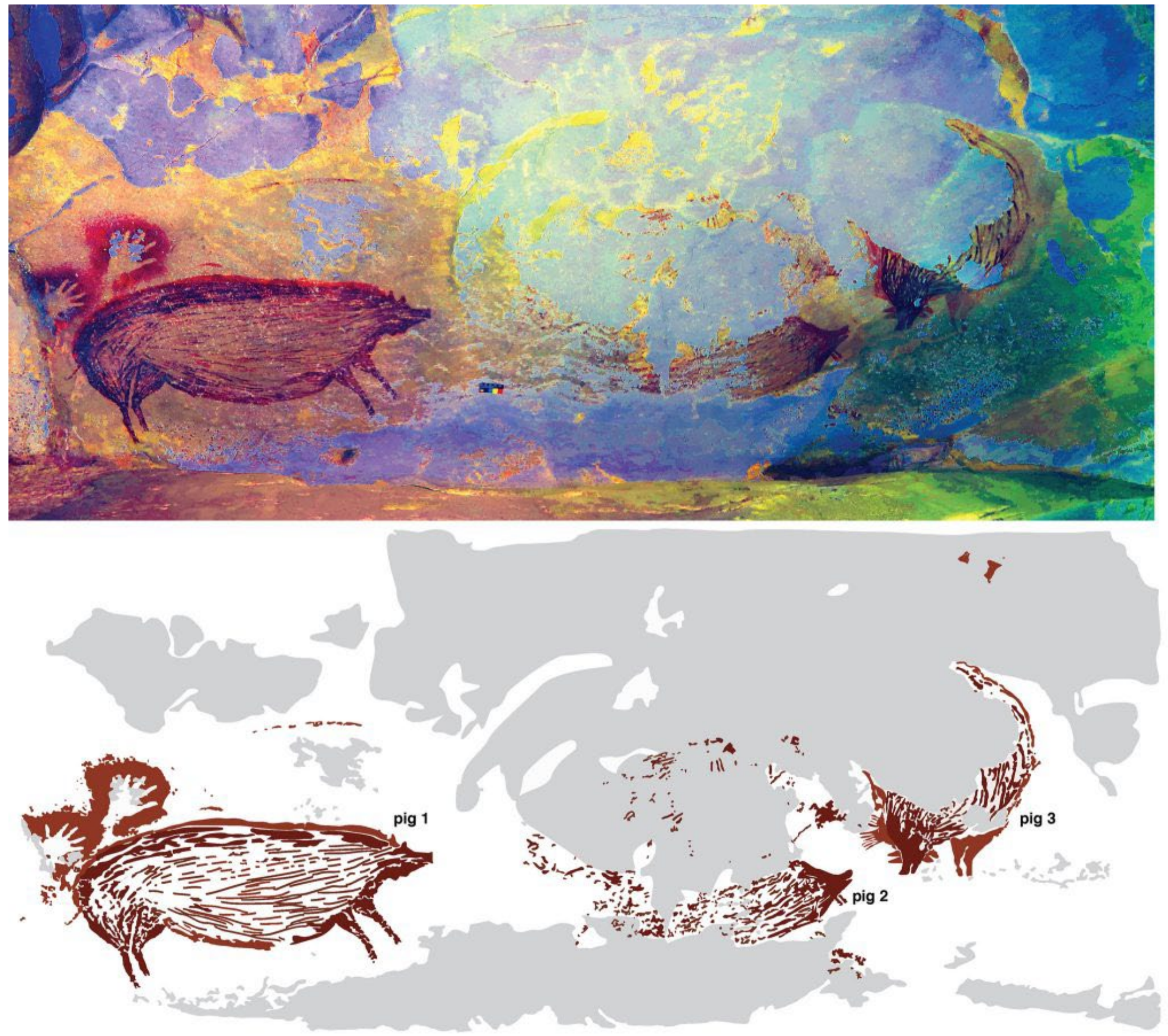
Experimentation

Results & Future work

Conclusion

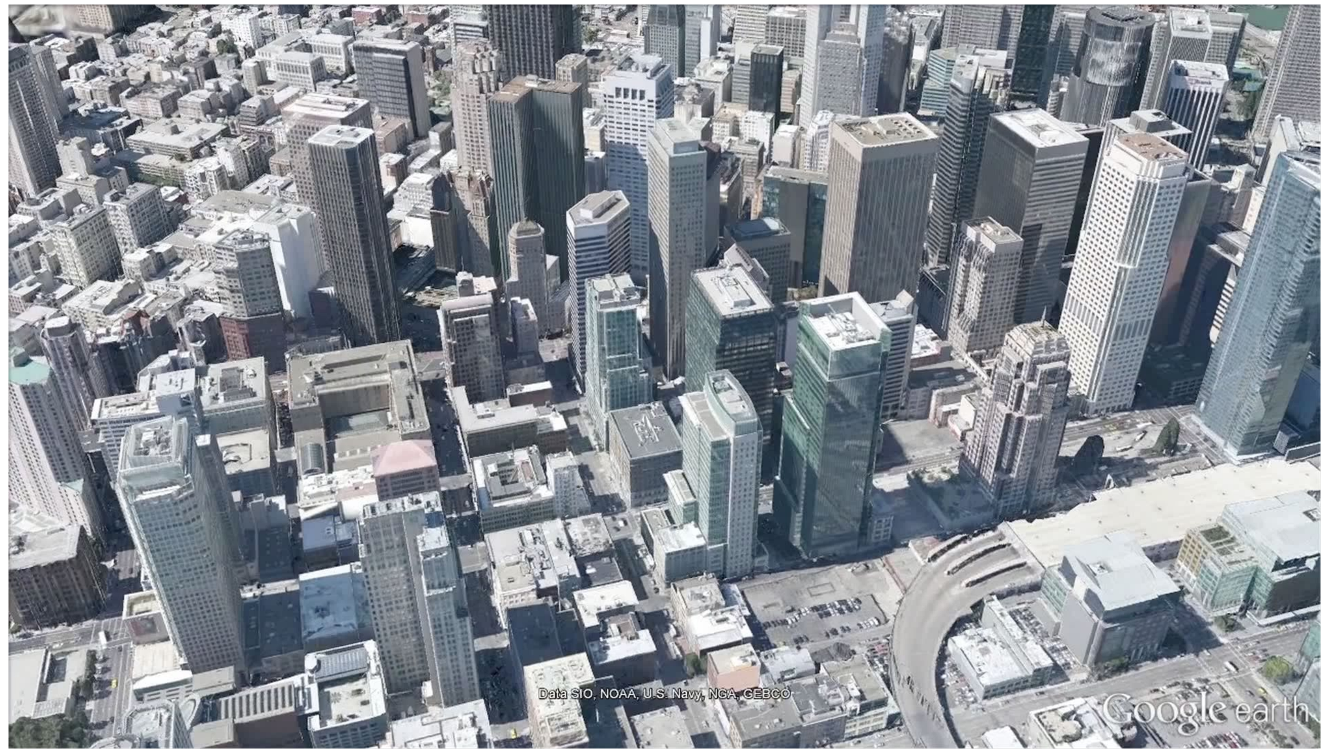
Introduction

Problem Statement (How far is 3D?)



Earliest cave painting (45,500 years old)
Sulawesi, Indonesia

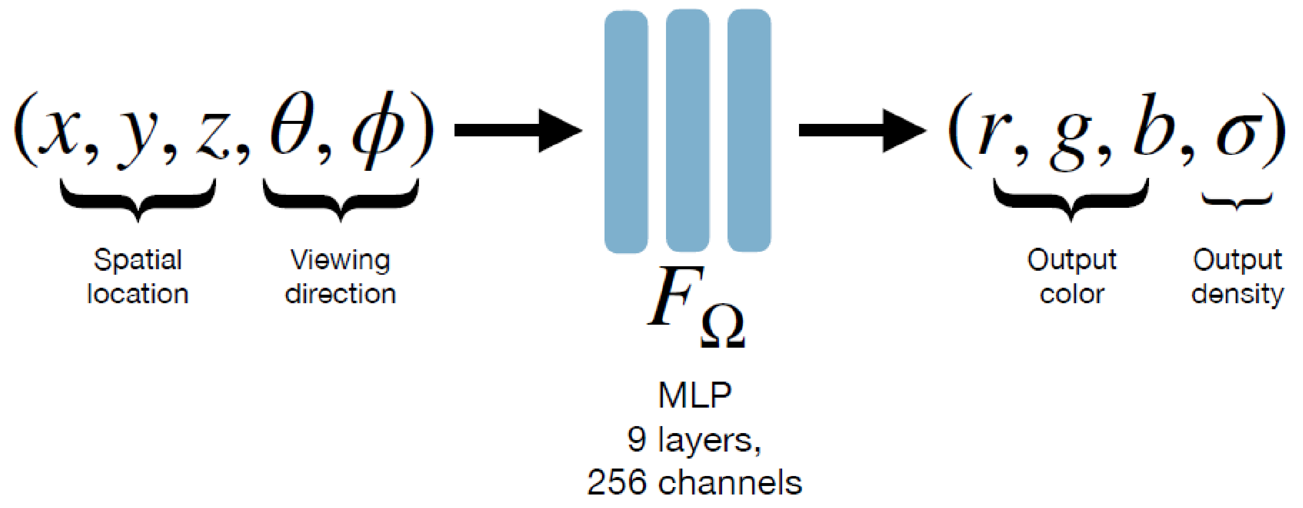
Monet's Cathedral series:
study of light 1893-1894



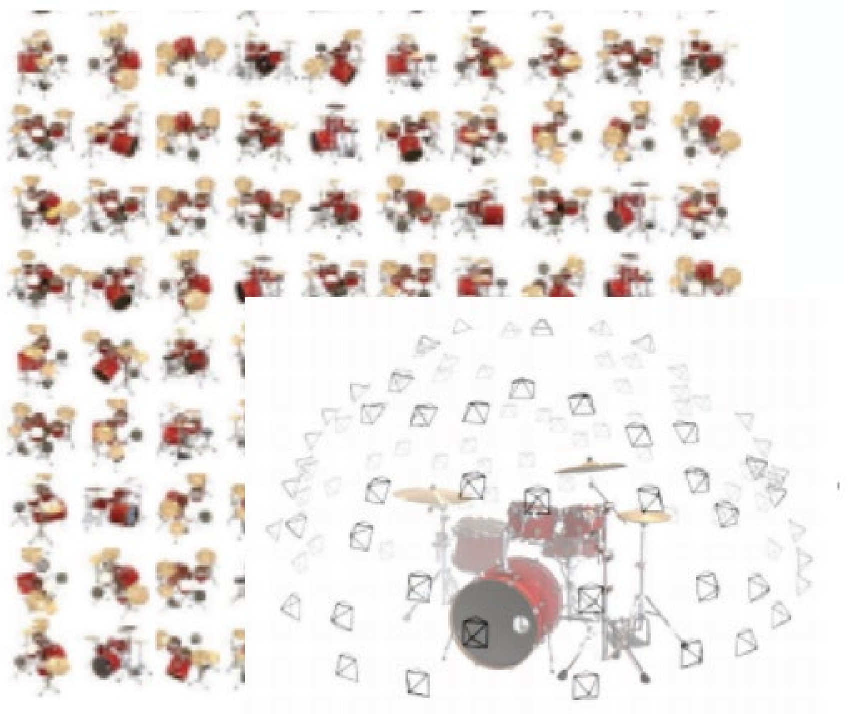
Google Earth 2016~

Motivation

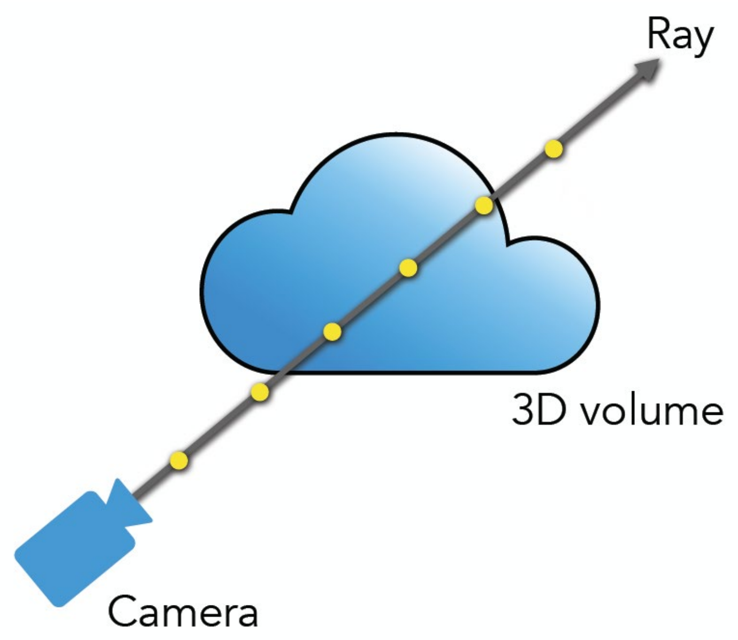
Nerf & point cloud diffusion



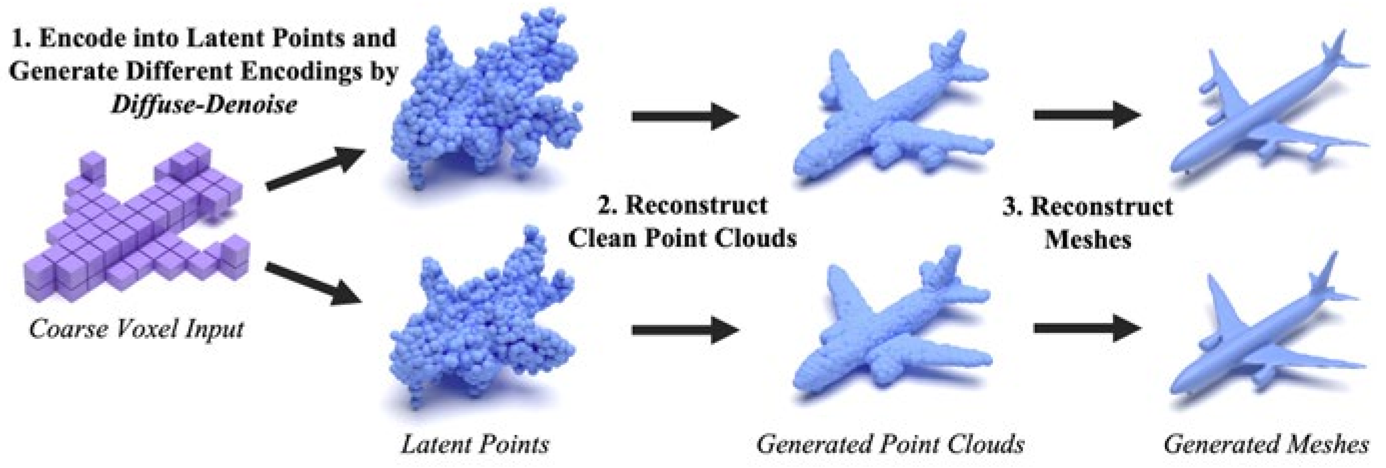
1. Neural Volumetric
3D Scene Representation



3. **Optimization** via Analysis-by-Synthesis
Objective: Synthesize all training views



2. Differentiable Volumetric
Rendering Function

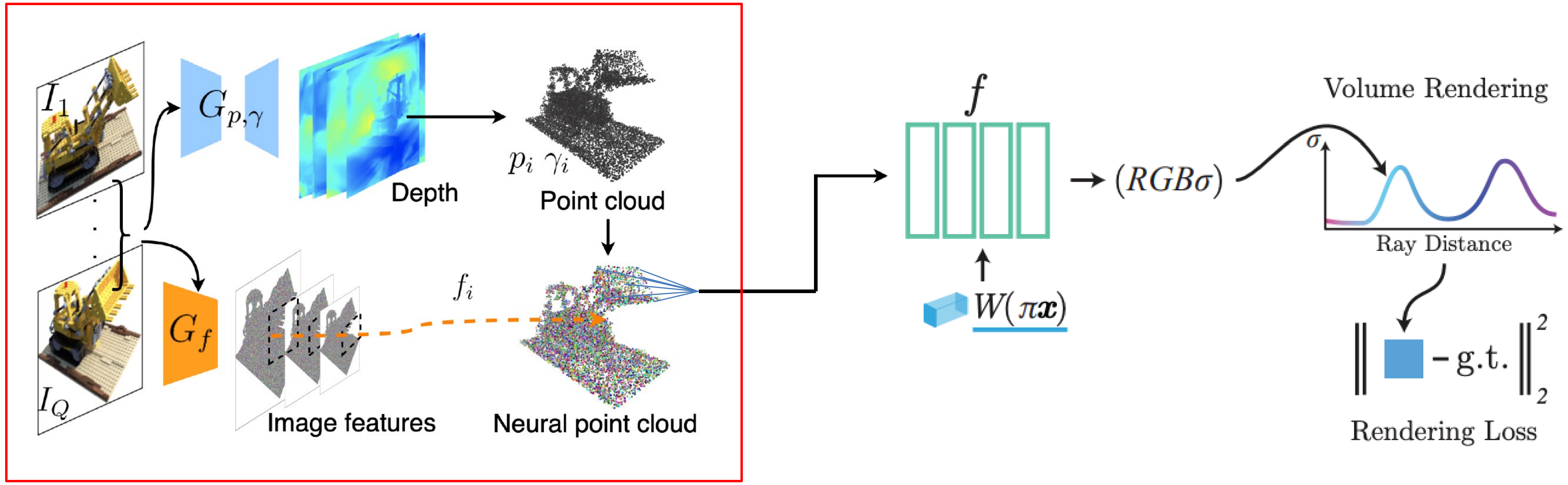


4. Voxel guided synthesis with LION.
Trained with **diffusion**-denoise

NeRF[B. Mildenhall et al., ACM 2021]
LION [X. Zheng et al., CVPR 2022]

Methods

Intermediate Neural point cloud

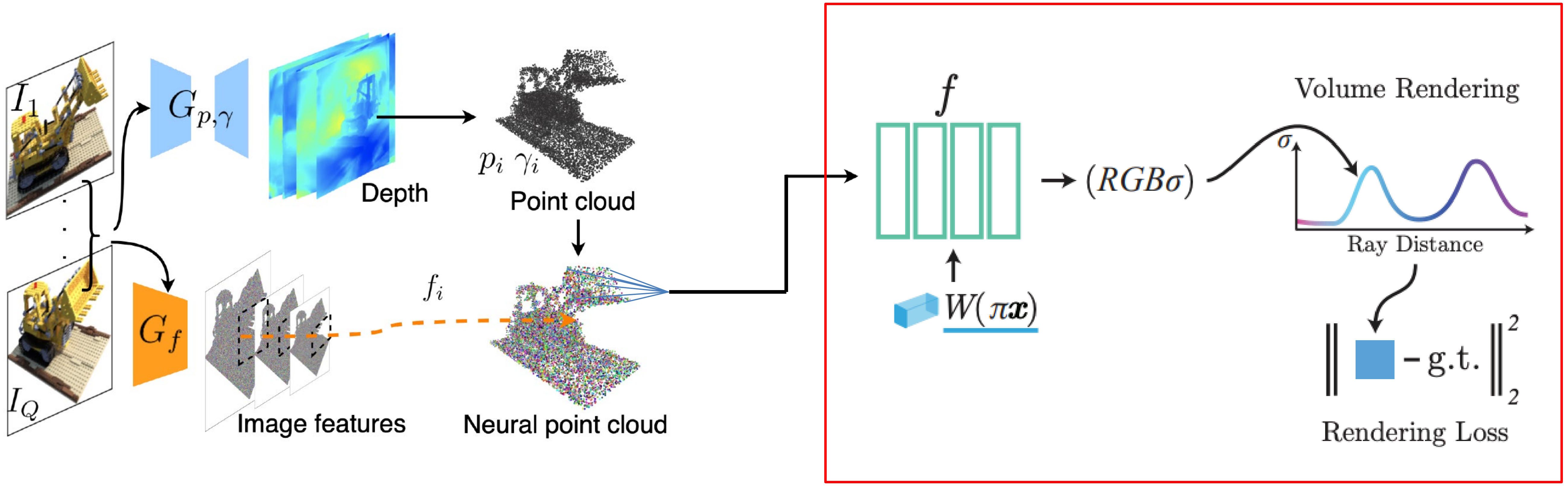


(a) Neural Generation using **Point Cloud**

- Our model takes into account multi-view images and creates **depth** for every perspective by harnessing the capability of 3D CNNs ***G_p*** that employs a cost-volume approach.
- Extracts **2D characteristics** from the provided images utilizing a specific 2D CNN, ***G_f***.
- Following the consolidation of the **depth map**, our model yields a radiance field based on **points.**
- In this field, each respective point is uniquely identified by its spatial position ***p_i***, an associated confidence factor (i), and features from the image that have not been projected ***f_i.***

Methods

Rendering and Optimization



(b) Novel **View** Synthesis

- Perform differentiable ray marching, computing shading in close proximity to the neural point cloud (for instance x_a , x_b , x_c).
- At each specific shading location, our method utilize to aggregate features from its **K** closest neural point neighbors, with the subsequent computation of the **radiance 'r'** and **volume density**.
- The radiance 'r' is then successively accumulated using the **volume density**.
- This entire operation is seamlessly trainable from end-to-end, and the point-based radiance field can be fine-tuned in alignment with the **rendering loss**.

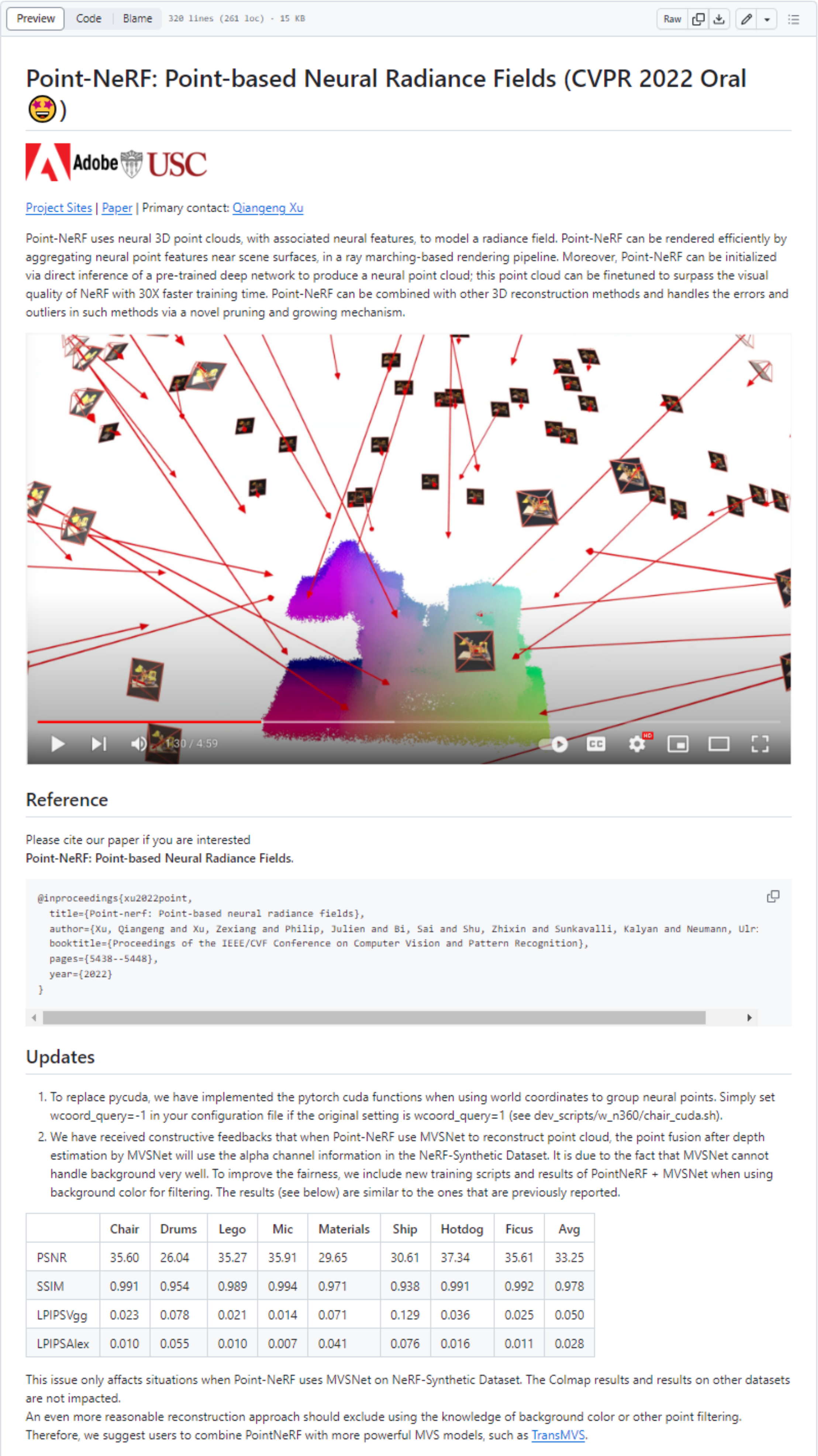
Experimentation

For this project, I started with the base code as a point-nerf.

1. Part of this was to train it on various datasets.
   - Typical datasets I used are the nerf synthetic dataset.

2. More options to enhance the learning
   - Fine-tuning hyperparameters
   - Improving preprocessing operations
   - Effectively utilizing the Great Lakes GPU in combination with various datasets

	Chair	Drums	Lego	Mic	Materials	Ship	Hotdog	Ficus	Avg
PSNR	35.60	26.04	35.27	35.91	29.65	30.61	37.34	35.61	33.25
SSIM	0.991	0.954	0.989	0.994	0.971	0.938	0.991	0.992	0.978
LPIPSVgg	0.023	0.078	0.021	0.014	0.071	0.129	0.036	0.025	0.050
LPIPSAlex	0.010	0.055	0.010	0.007	0.041	0.076	0.016	0.011	0.028



Preview Code Blame 320 lines (261 loc) · 15 KB Raw

Point-NeRF: Point-based Neural Radiance Fields (CVPR 2022 Oral 🤩)

Adobe USC

Project Sites | Paper | Primary contact: Qiangeng Xu

Point-NeRF uses neural 3D point clouds, with associated neural features, to model a radiance field. Point-NeRF can be rendered efficiently by aggregating neural point features near scene surfaces, in a ray marching-based rendering pipeline. Moreover, Point-NeRF can be initialized via direct inference of a pre-trained deep network to produce a neural point cloud; this point cloud can be finetuned to surpass the visual quality of NeRF with 30X faster training time. Point-NeRF can be combined with other 3D reconstruction methods and handles the errors and outliers in such methods via a novel pruning and growing mechanism.

1:30 / 4:59

Reference

Please cite our paper if you are interested
Point-NeRF: Point-based Neural Radiance Fields.

```
@inproceedings{xu2022point,
  title={Point-nerf: Point-based neural radiance fields},
  author={Xu, Qiangeng and Xu, Zexiang and Philip, Julien and Bi, Sai and Shu, Zhixin and Sunkavalli, Kalyan and Neumann, Ulr:
  booktitle={Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition},
  pages={5438--5448},
  year={2022}
}
```

Updates

1. To replace pycuda, we have implemented the pytorch cuda functions when using world coordinates to group neural points. Simply set wcoord_query=-1 in your configuration file if the original setting is wcoord_query=1 (see dev_scripts/w_n360/chair_cuda.sh).
2. We have received constructive feedbacks that when Point-NeRF use MVSNet to reconstruct point cloud, the point fusion after depth estimation by MVSNet will use the alpha channel information in the NeRF-Synthetic Dataset. It is due to the fact that MVSNet cannot handle background very well. To improve the fairness, we include new training scripts and results of PointNeRF + MVSNet when using background color for filtering. The results (see below) are similar to the ones that are previously reported.

	Chair	Drums	Lego	Mic	Materials	Ship	Hotdog	Ficus	Avg
PSNR	35.60	26.04	35.27	35.91	29.65	30.61	37.34	35.61	33.25
SSIM	0.991	0.954	0.989	0.994	0.971	0.938	0.991	0.992	0.978
LPIPSVgg	0.023	0.078	0.021	0.014	0.071	0.129	0.036	0.025	0.050
LPIPSAlex	0.010	0.055	0.010	0.007	0.041	0.076	0.016	0.011	0.028

This issue only affacts situations when Point-NeRF uses MVSNet on NeRF-Synthetic Dataset. The Colmap results and results on other datasets are not impacted.
An even more reasonable reconstruction approach should exclude using the knowledge of background color or other point filtering. Therefore, we suggest users to combine PointNeRF with more powerful MVS models, such as TransMVS.

Images from Point-NeRF directory

Results & Future work

Real image | Depth image | Real image | Depth image

PSNR	LPIPSVgg
SSIM	LPIPSAlex

Reconstructed image

Reconstructed Depth

Conclusion

- Combining two of the state-of-art ideas to create something new was quite difficult.
- Along the way, various CNN networks appeared that needed to be trained separately.
- The optimization we learned in class also comes into play in the rendering process, which converts a 2D image into 3D.
- In the end, 3D reconstruction is all about how you find the missing information.