Point2Pix: Photo-Realistic Point Cloud Rendering via Neural Radiance Fields

Tao Hu, Xiaogang Xu, Shu Liu, Jiaya Jia

The Chinese University of Hong Kong, SmartMore

Content

- *Background and Motivation*
- *Our Approach*
- *Experiments*
- *Visualization*

Point2Pix

Background and Motivation

I. *Point Cloud Rendering is conducive to 3D visualization, navigation, and augmented reality;*

II. *Graphics-base rendering only generates image with holes;*

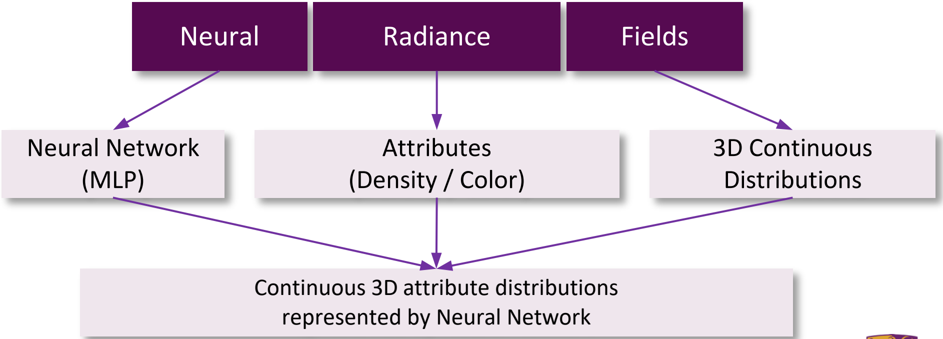
III. *Neural Radiance Fields (NeRF) can synthesis photo-realistic images thus our method combines point cloud with NeRF;*

IV. *Advantages of combining Point Cloud with NeRF, i.e., Point2Pix:*

- *Multi-scale NeRF to overcome hole artifacts*
- *Efficient Point Sampling for NeRF*
- *Generalization for Point Cloud Feature Extraction*

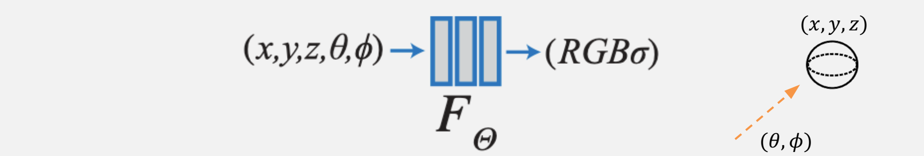
Efficient Neural Radiance Fields

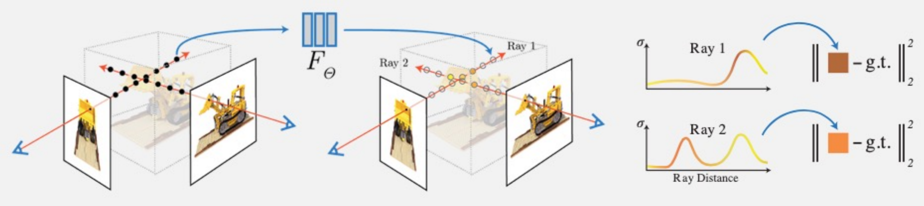
Novel 3D Representation: Neural Radiance Fields (NeRF) [1]



Neural Radiance Fields (NeRF)

Main Idea: Query all points' $RGB\sigma$ from an MLP for volume rendering





Point2Pix

Our Approach

I. Point-guided Sampling

We treat the queried point x_i as a valid sample then obtain the point feature, when it satisfied the following equation:

$$\|p_i - x_i\| \leq r$$

II. Multi-scale Radiance Fields

We extract 3D point feature from multiple scales and render to 2D Feature Maps:

$$(\sigma_i, f_i) = \Phi(F_i) = \Phi_i(F[x_i])$$

$$f = \sum w_i \cdot f_i$$

$$w_i = \exp(-\sum \sigma_i \delta_i)(1 - \exp(-\sigma_i \delta_i))$$

III. Fusion Decoding

We fuse multiple 2D feature maps to decode image

$$(\gamma, \beta) = Conv2D(f)$$

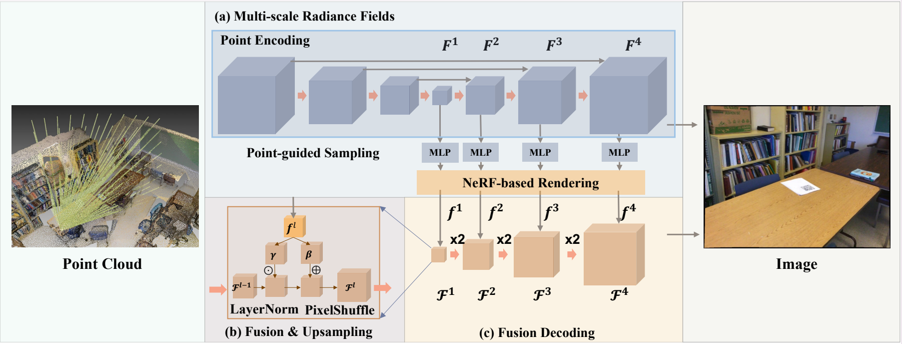
$$F \leftarrow \gamma \cdot LayerNorm(F) + \beta$$

IV. Loss Function

$$\ell = \lambda_{pc}\ell_{pc} + \lambda_{nr}\ell_{nr} + \lambda_{per}\ell_{pr}$$

Point2Pix

Our Approach: Overview



Our Approach

I. Point-guided Sampling

We treat the queried point x_i as a valid sample then obtain the point feature, when it satisfied the following equation:

$$\|p_i - x_i\| \leq r$$

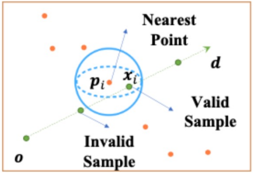


Figure 2. The proposed point-guided sampling. For any queried point $\mathbf{x}_i$, we find its nearest point $\mathbf{p}_i$ in the point cloud. If $\mathbf{x}_i$ is located in the ball area (with radius r) of $\mathbf{p}_i$, it is a valid sample. Invalid samples are omitted to improve sampling efficiency.

Point2Pix

Our Approach

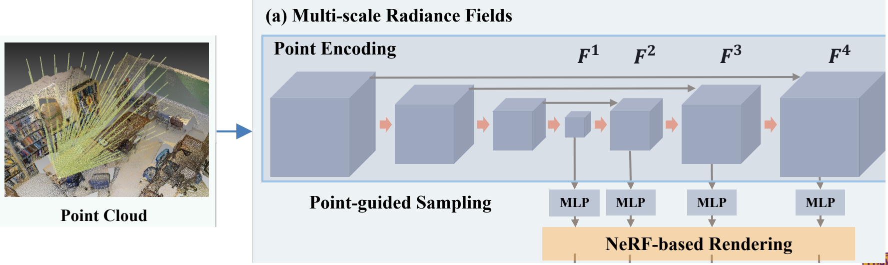
II. Multi-scale Radiance Fields

We extract 3D point feature from multiple scales and render to 2D Feature Maps:

$$(\sigma_i, f_i) = \Phi(F_i) = \Phi_i(F[x_i])$$

$$f = \sum w_i \cdot f_i$$

$$w_i = \exp(-\sum \sigma_i \delta_i)(1 - \exp(-\sigma_i \delta_i))$$



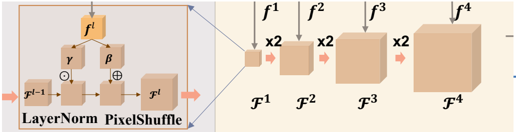
Our Approach

III. Fusion Decoding

We fuse multiple 2D feature maps to decode image

$$(\gamma, \beta) = Conv2D(f)$$

$$F \leftarrow \gamma \cdot LayerNorm(F) + \beta$$



(b) Fusion & Upsampling (c) Fusion Decoding

Point2Pix

Our Approach

IV. Loss Function

$$\ell = \lambda_{pc}\ell_{pc} + \lambda_{nr}\ell_{nr} + \lambda_{per}\ell_{pr}$$

Point Cloud Loss: Point Cloud provides ground-truth density and color

$$\ell_{pc} = \sum_{k=1}^{K} \|\hat{c}_k - c_k\|_2^2 + \frac{1}{D} max(0, D - \sigma_k)$$

Neural Rendering Loss: Image Reconstruction Loss

$$\ell_{nr} = \left\|\hat{I} - I\right\|_2^2$$

Neural Rendering Loss: Image Reconstruction Loss

$$\ell_{per} = \left\|\phi_l(\hat{I}) - \phi(I)\right\|_2^2$$

Experiments

I. Quantitively Comparison on ScanNet and ArkitScene dataset

Dataset	ScanNet [9]			ARKitScenes [3]		
Metrics	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR↑	SSIM↑	LPIPS↓
Pytorch3D [38]	13.62	0.528	0.779	15.21	0.581	0.756
Pix2PixHD [47]	15.59	0.601	0.611	15.94	0.636	0.605
NPCR [10]	16.22	0.659	0.574	16.84	0.661	0.518
NPBG++ [11]	16.81	0.671	0.585	17.23	0.692	0.511
ADOP [41]	16.83	**0.699**	0.577	17.32	0.707	**0.495**
Point-NeRF [51]	**17.53**	0.685	**0.517**	**17.61**	**0.715**	0.508
Point2Pix (Ours)	**18.47**	**0.723**	**0.484**	**18.84**	**0.734**	**0.471**

Table 1. Comparing our method with different point renderers on the ScanNet [9] and ARkitScenes [3] datasets. There is no finetuning process in this experiment, which demonstrates the generalization in novel scenes.

Method	Time	PSNR(↑)	SSIM (↑)	LPIPS(↓)
Point-NeRF [51]	0 mins	17.53	0.685	0.517
Point2Pix (Ours)	0 mins	**18.47**	**0.723**	**0.484**
NeRF [29]	∼30 hours	21.33	0.788	0.355
NSVF [23]	∼40 hours	22.47	0.791	0.337
PlenOctrees [54]	∼30 hours	22.02	0.795	0.341
Instant-NGP [30]	20 mins	21.94	0.775	0.363
Plenoxels [53]	20 mins	22.35	0.780	0.346
Point-NeRF [51]	20 mins	22.55	0.792	0.336
Point2Pix (Ours)	20 mins	**23.02**	**0.815**	**0.318**

Table 2. Comparing our method with NeRF-based methods on the ScanNet dataset [9]. "Time" means the average finetuning time for all scenes.

Point2Pix

Visualization

I. Qualitative Comparison on ScanNet and ArkitScene dataset

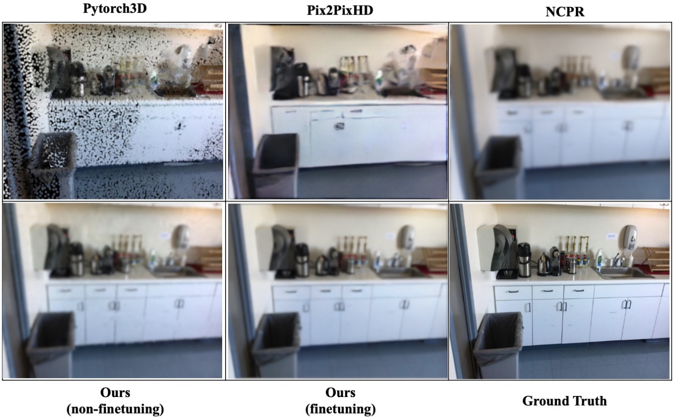


Figure 3. Qualitative comparison between different point renderers on the ScanNet [9].

Visualization

I. *Qualitative Comparison on ScanNet and ArkitScene dataset*

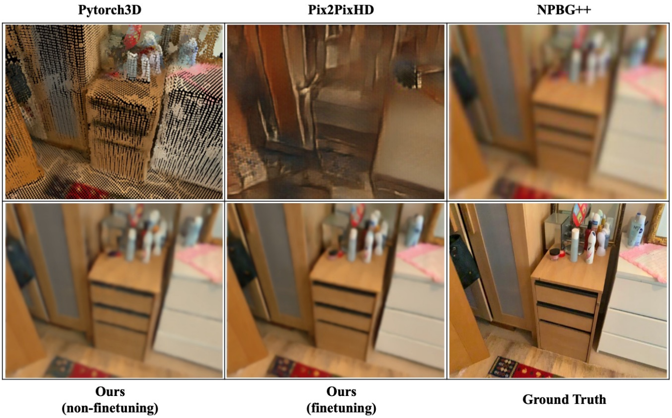
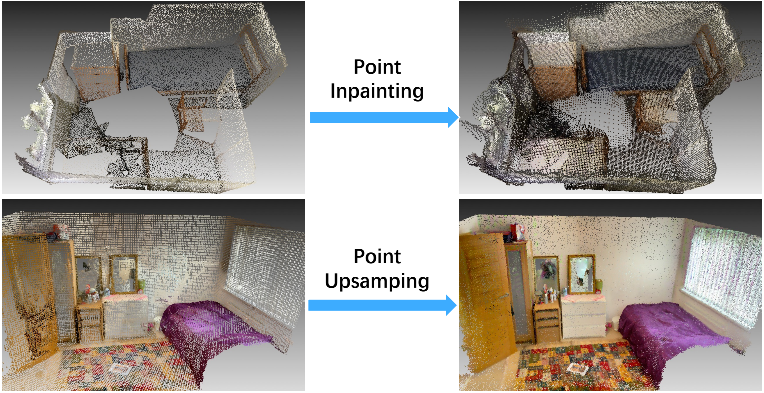


Figure 4. Qualitative comparison between different point renderers and NeRF-based methods on the ArkitScenes [3] dataset.

Visualization

I. Qualitative Comparison on Point Cloud Inpainting and Upsampling



Raw Point Ploud **Point2Pix (Ours)**

Point2Pix: Photo-Realistic Point Cloud Rendering via Neural Radiance Fields

Thank you!