CVPR Nashville JUNE 11-15, 2025

ARC-NeRF: Area Ray Casting for Broader Unseen View Coverage in Few-shot Object Rendering

Seunghyeon Seo[1] Yeonjin Chang[1] Jayeon Yoo[1] Seungwoo Lee[1] Hojun Lee[1,2] Nojun Kwak[1]

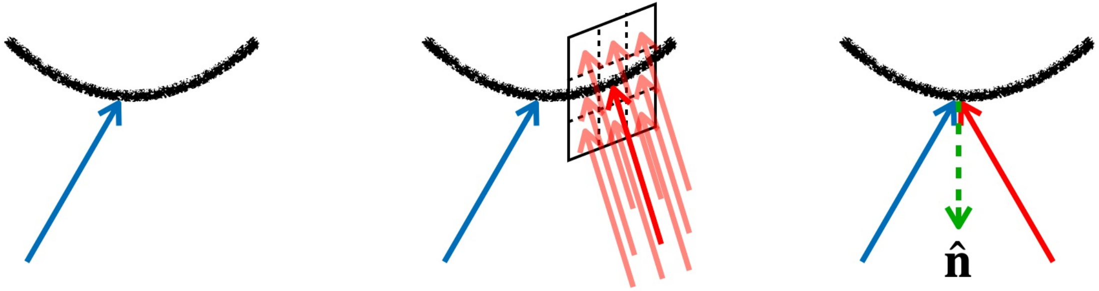
[1]Seoul National University [2]Xperty Corp.

Motivation

- NeRF suffers from a severe performance degradation with only a sparse set of inputs.

Motivation

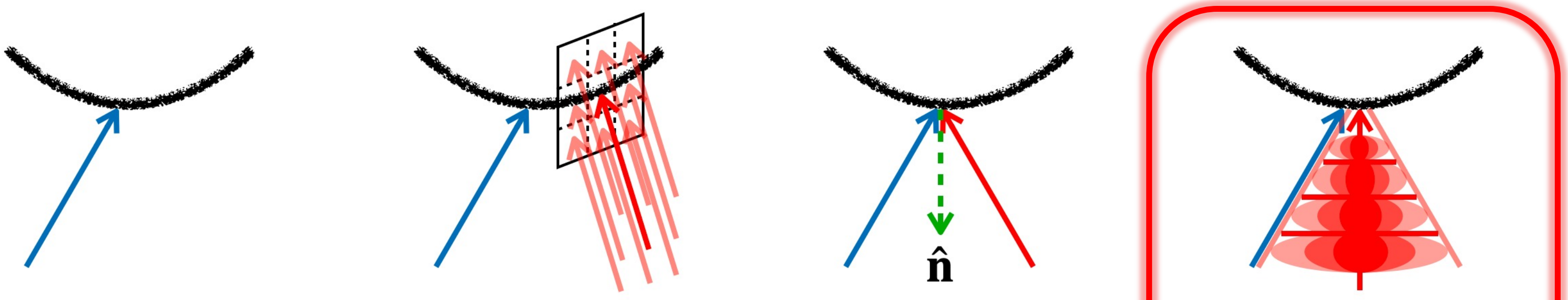
- Ray augmentation methods:
  > More augmented rays => higher training cost
  > Many potentially useful rays in unseen views remain unused.



(a) No aug. (b) Patch rendering with aug. rays (c) Flipped reflection rays

Motivation

- Ray augmentation methods:
  > More augmented rays => higher training cost
  > Many potentially useful rays in unseen views remain unused.
  >> *Area Ray* covers a broad area of unseen views.



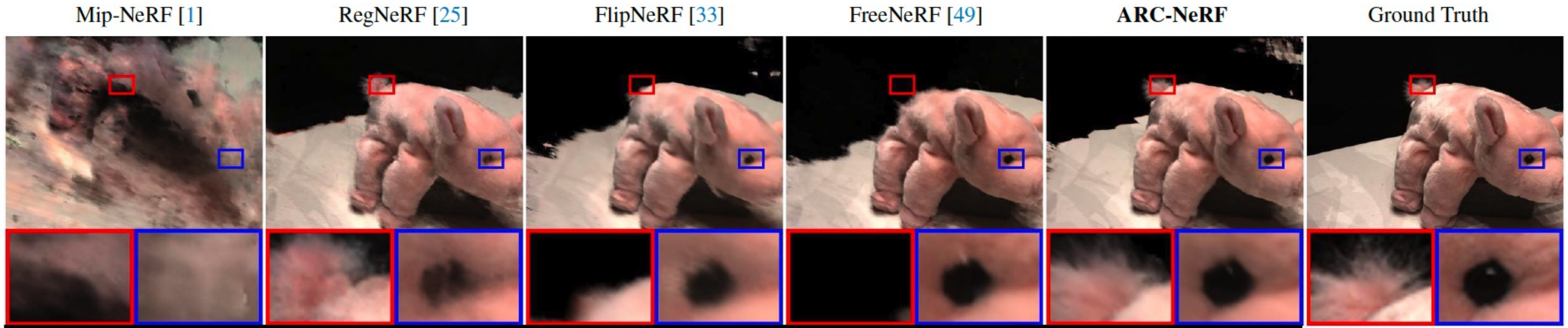
(a) No aug. (b) Patch rendering with aug. rays (c) Flipped reflection rays (d) Area Ray (Ours)

Motivation

- ***Area Ray*:**

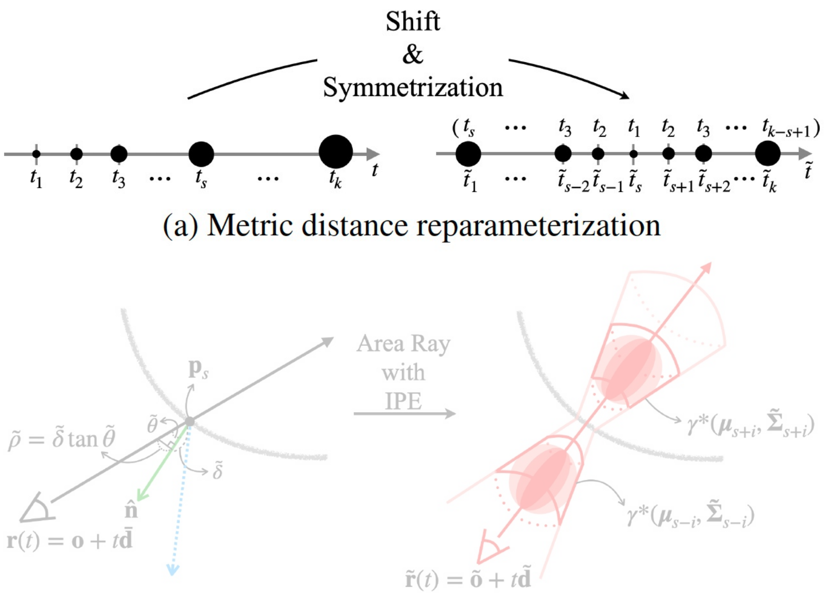
  > Adaptive high-frequency regularization

  > Achieves better training efficiency than brute-force ray augmentation



Mip-NeRF [1]	RegNeRF [25]	FlipNeRF [33]	FreeNeRF [49]	ARC-NeRF
✗	✓ Patch rendering with aug. rays	✓ Flipped reflection rays	✗	✓ Area Ray
✗	✗	✗	✓ Masking curriculum	✓ Adaptive reg. with IPE

Area Ray

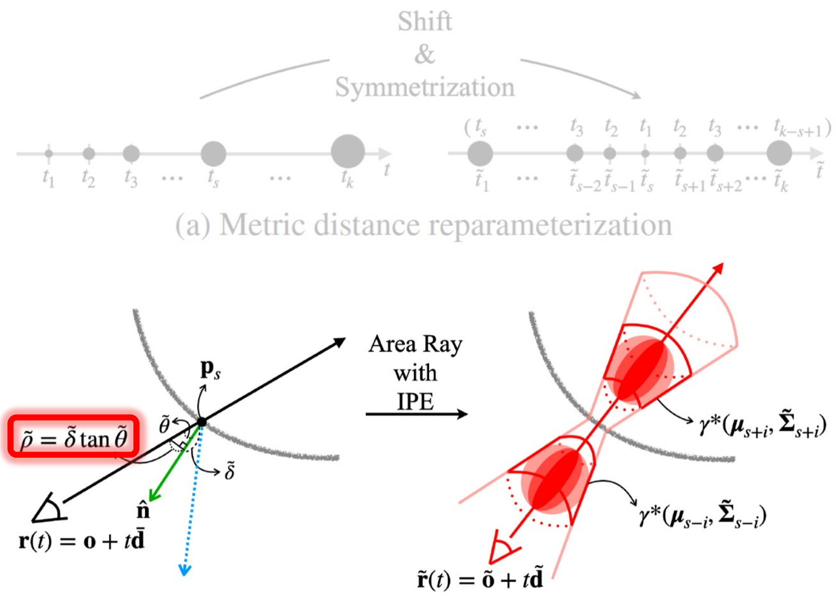


(a) Metric distance reparameterization

(b) Area Ray featurization by IPE

- Reparameterize the metric distance t as $\tilde{t}$ to derive the variance $\tilde{\sigma}_{\rho}^{2}$.

Area Ray



(a) Metric distance reparameterization

(b) Area Ray featurization by IPE

- Reparameterize the metric distance t as $\tilde{t}$ to derive the variance $\tilde{\sigma}_\rho^2$.
- Derive a base radius $\tilde{\rho}$ of the Area Ray using the trigonometric function.

MIPALaboratory
Machine Intelligence
& Pattern Analysis

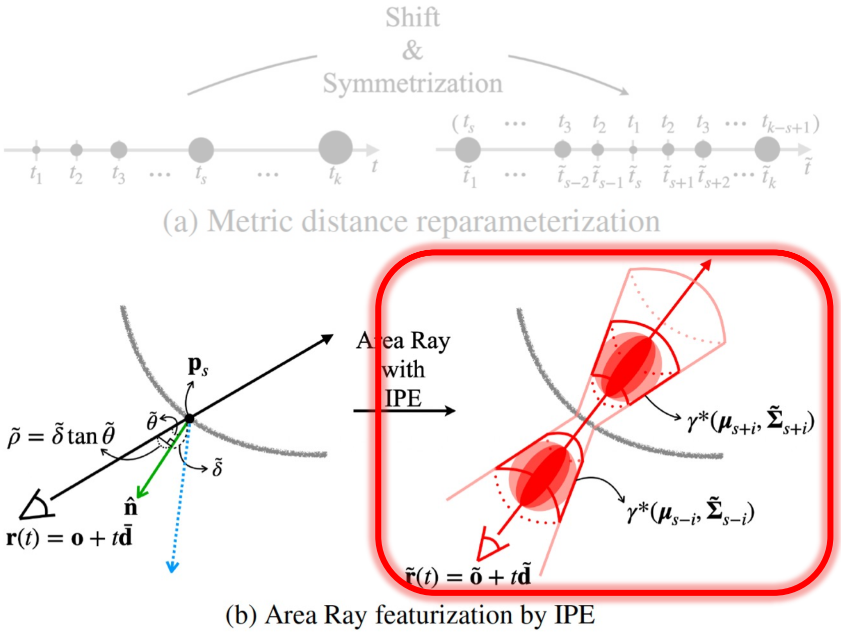
Area Ray

- Area Ray:

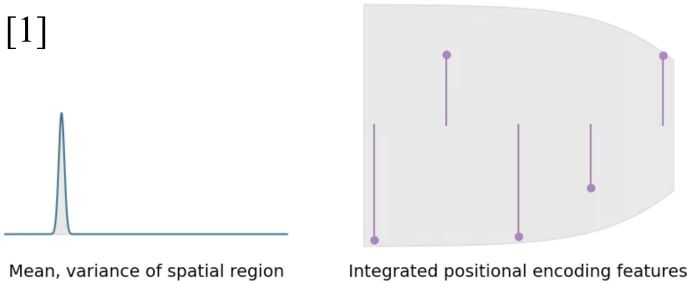
$$\tilde{\mathbf{r}}(t) = \tilde{\mathbf{o}} + t\tilde{\mathbf{d}},$$

where $\tilde{\mathbf{d}} = -\hat{\mathbf{n}}$ and $\tilde{\mathbf{o}} = \mathbf{p}_s - t_s\tilde{\mathbf{d}}$.

- High-frequency details are more tightly regularized in pixels with lower photo-consistency.
=> Adaptive high-frequency reg.



[1]



[1] Barron, Jonathan T., et al. "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields." ICCV 2021.

Luminance Consistency Regularization

- Luminance maps offer a “free lunch” training signal that can be easily extracted from RGB images.

Luminance Consistency Regularization

- Luminance maps offer a "free lunch" training signal that can be easily extracted from RGB images.

1) Compute the GT relative luminance of a target pixel:

$$y_{\mathrm{GT}} = \sum_{\bar{c}}^{\{\bar{r},\bar{g},\bar{b}\}} \lambda_{\bar{c}} \bar{c}$$

where $\bar{c} = c_{\mathrm{GT}}^{2.2}$ is the linear RGB value obtained from gamma-compressed input via a simple power transformation.

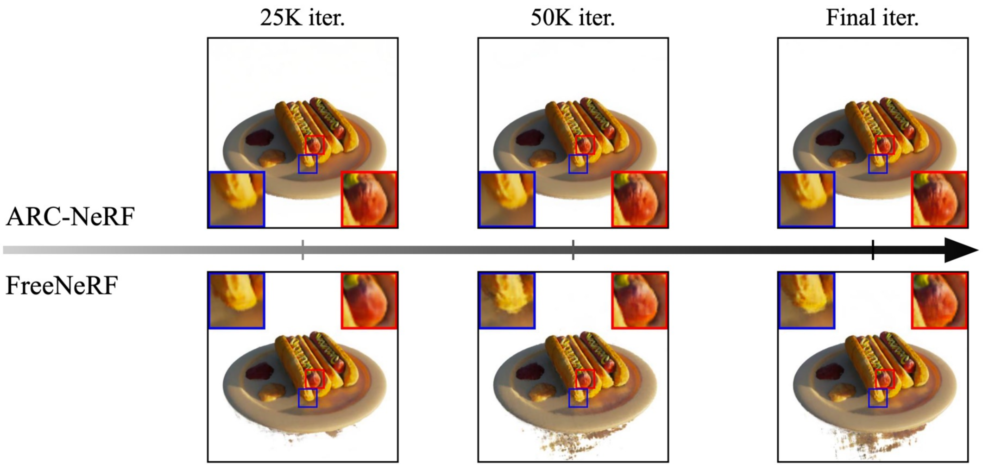
Luminance Consistency Regularization

- Luminance maps offer a “free lunch” training signal that can be easily extracted from RGB images.

2) Predict the per-sample luminance y_i along a ray and aggregate the final luminance $\hat{y}$ by volume rendering:

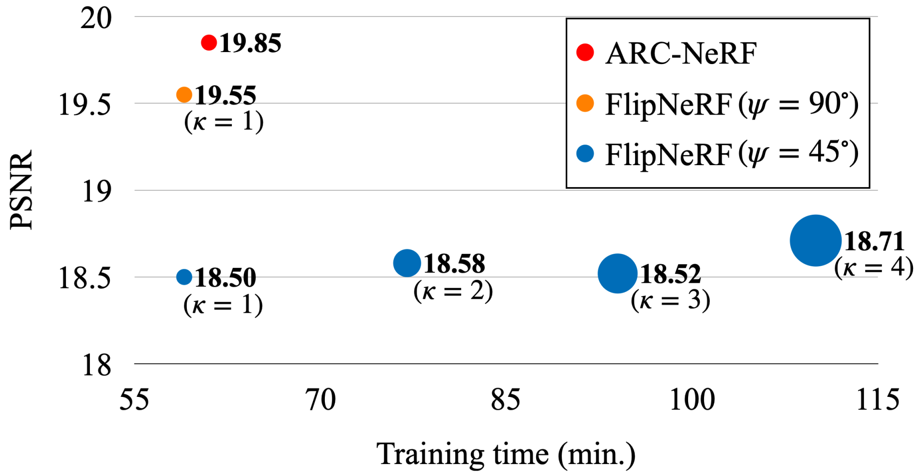
$$\hat{y}(\mathbf{r}) = \sum_{i=1}^{N} w_i y_i$$

Analysis



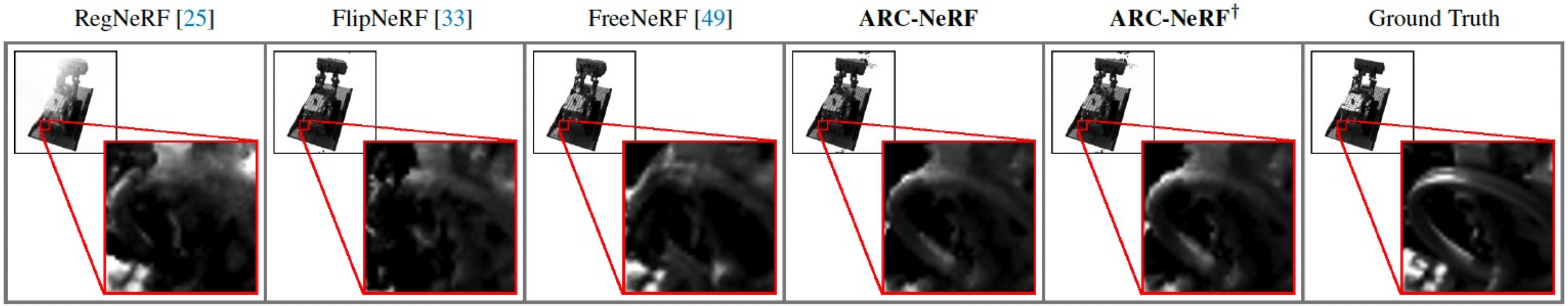
- FreeNeRF: hard masking of high-frequencies in early training
- Ours: adaptive frequency regularization based on ray–pixel consistency (angle-based)
  → Sharper details at just 25K iters, even better than fully trained FreeNeRF

Analysis



- Single Area Ray with adaptive regularization outperforms multicasting multiple augmented rays across the same region.
- ARC-NeRF improves rendering quality and training efficiency without the cost of processing extra rays.

Analysis



- ARC-NeRF produces sharper relative luminance maps compared to other baselines.
- Luminance estimation provides additional supervision to blending weights via shared rendering pipeline.

	PSNR ↑	SSIM ↑	LPIPS ↓	Average ↓
RegNeRF [25]	5.71	0.780	0.291	0.368
FlipNeRF [33]	16.49	0.878	0.080	0.092
FreeNeRF [49]	15.63	0.869	0.091	0.113
ARC-NeRF	<u>16.87</u>	<u>0.886</u>	<u>0.074</u>	<u>0.088</u>
ARC-NeRF†	**17.18**	**0.891**	**0.062**	**0.079**

Analysis

	Area Ray	$\mathcal{L}_{\text{lum.}}$	PSNR ↑	SSIM ↑	LPIPS ↓	Avg. ↓
FlipNeRF [33]			<u>19.55</u>	0.767	0.180	0.101
(1)	✓		19.51	**0.774**	<u>0.147</u>	<u>0.097</u>
(2)		✓	18.44	0.747	0.201	0.119
(3)	✓	✓	**19.85**	<u>0.773</u>	**0.146**	**0.096**

- Replacing FlipNeRF's reflection rays with Area Ray improves SSIM and LPIPS.
- Using only $L_{\text{lum.}}$ without Area Ray leads to degraded performance.
  > View-dependent nature of reflection rays conflicts with $L_{\text{lum.}}$, causing mismatch.
- Combining Area Ray with $L_{\text{lum.}}$ yields best results, especially in PSNR with sharper fine details.

Results

Realistic Synthetic 360

	PSNR ↑ 4-view	PSNR ↑ 8-view	SSIM ↑ 4-view	SSIM ↑ 8-view	LPIPS ↓ 4-view	LPIPS ↓ 8-view	Avg. ↓ 4-view	Avg. ↓ 8-view
Mip-NeRF [1]	8.70	13.31	0.792	0.848	0.250	0.176	0.285	0.188
DietNeRF [9]	10.86	16.08	0.814	0.870	0.194	0.113	0.223	0.123
InfoNeRF [15]	13.65	16.74	0.834	0.865	0.134	0.094	0.139	0.095
RegNeRF [25]	7.24	13.47	0.795	0.856	0.292	0.158	0.318	0.177
MixNeRF [34]	16.13	19.31	0.863	0.902	0.099	0.058	0.101	0.065
FreeNeRF [49]	15.71	18.99	0.857	0.894	0.103	0.064	0.114	0.072
FlipNeRF [33]	16.47	19.54	0.866	0.903	0.091	0.057	0.095	0.062
ARC-NeRF	**16.86**	**20.29**	**0.873**	**0.910**	**0.084**	**0.052**	**0.091**	**0.057**

Shiny Blender 4-view

	PSNR ↑	SSIM ↑	LPIPS ↓	Average ↓
Ref-NeRF [41]	17.10	0.821	0.190	0.142
FreeNeRF [49]	16.99	0.828	0.157	0.131
FlipNeRF [33]	18.14	0.847	**0.141**	0.109
ARC-NeRF	**18.68**	**0.851**	**0.141**	**0.107**

DTU 3-view

	PSNR ↑	SSIM ↑	LPIPS ↓	Avg. ↓
Mip-NeRF [1]	8.68	0.571	0.353	0.323
3DGS [14]	14.18	0.628	0.301	0.191
Pre-training.				
PixelNeRF [50]	16.82	0.695	0.270	0.147
PixelNeRF† [50]	18.95	0.710	0.269	0.125
SRF [5]	15.32	0.671	0.304	0.171
SRF† [5]	15.68	0.698	0.281	0.162
MVSNeRF [4]	18.63	0.769	0.197	0.113
MVSNeRF† [4]	18.54	0.769	0.197	0.113
Regularization.				
DietNeRF [9]	11.85	0.633	0.314	0.243
RegNeRF [25]	18.89	0.745	0.190	0.112
MixNeRF [34]	18.95	0.744	0.203	0.113
SimpleNeRF [35]	16.25	0.751	0.249	0.143
DiffusioNeRF [46]	16.20	0.698	0.160	0.128
SparseNeRF [42]	19.55	0.769	0.201	0.102
FreeNeRF‡ [49]	19.92	0.781	0.125	0.086
FreeNeRF [49]	19.23	0.769	0.149	0.103
FlipNeRF [33]	19.55	0.767	0.180	0.101
SparseGS [47]	18.89	0.702	0.229	0.117
ARC-NeRF	**19.85**	**0.773**	**0.146**	**0.096**

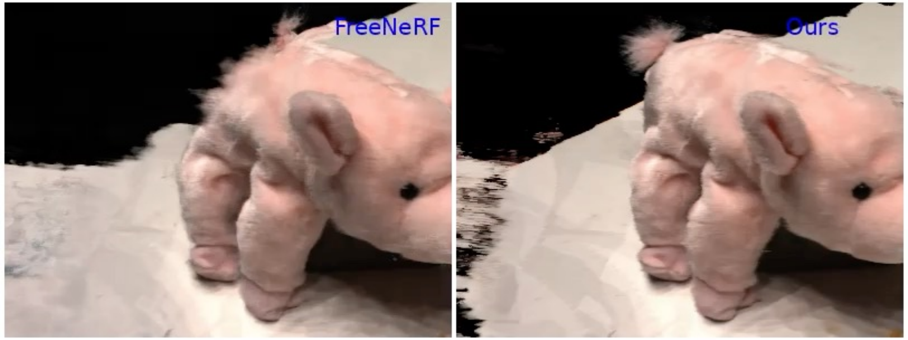
MIPALaboratory
Machine Intelligence
& Pattern Analysis

Results

Realistic Synthetic 360 4-view

Realistic Synthetic 360 4-view

Results



3-view; Notable improvement in the detail of the tail.

6-view; Apple surface textures are more stably reconstructed across changing views.

9-view; Brick textures are also more consistently reproduced.

Thank you!