Mip-NeRF :

A Multiscale Representation for Anti-Aliasing Neural Radiance Fields

Seungyeol Lee

CHAPTER

1. Purpose of Research
2. Limitations of Previous Research
3. Mip-NeRF
4. Key Components of Mip-NeRF
5. Experiments
6. Conclusions

Part 1 Purpose of Research

- Novel View Synthesis that prevents “blurring” and “aliasing”

Blurring

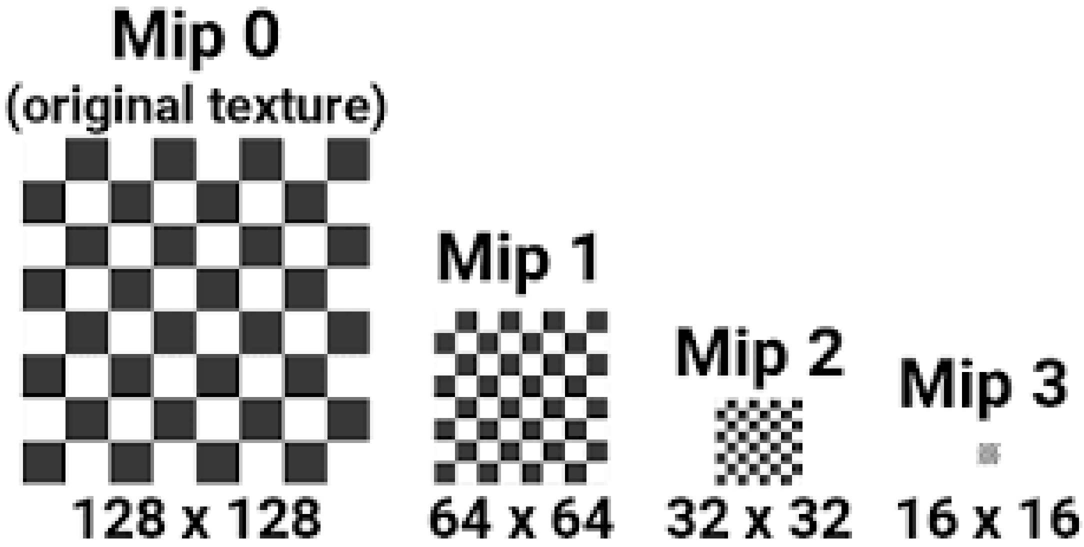
Purpose of Research

- **Novel View Synthesis that prevents “blurring” and “aliasing”**

Aliasing

Purpose of Research

- Mipmap



Limitations of Previous Research

1. Anti-aliasing in Rendering

(1) Supersampling-based techniques

- cast multiple rays per pixel while rendering to closer to the Nyquist frequency.

- can reduce aliasing, but expensive, as runtime generally scales linearly .

- typically used only in offline rendering contexts.

Part 2

Limitations of Previous Research

1. Anti-aliasing in Rendering

(2) Prefiltering-based techniques

- better suited for real-time rendering.

- correct scale can be used at a render time depending on the target sampling rate.

- tracing a cone instead of a ray through each pixel.

Part 2

Limitations of Previous Research

2. Scene Representations for View Synthesis

(1) Volumetric Representations

- Using gradient-based methods to optimize mesh geometry and topology is difficult.

  (due to discontinuities and local minima.)

- Volumetric representations have therefore become increasingly popular.

Limitations of Previous Research

2. Scene Representations for View Synthesis

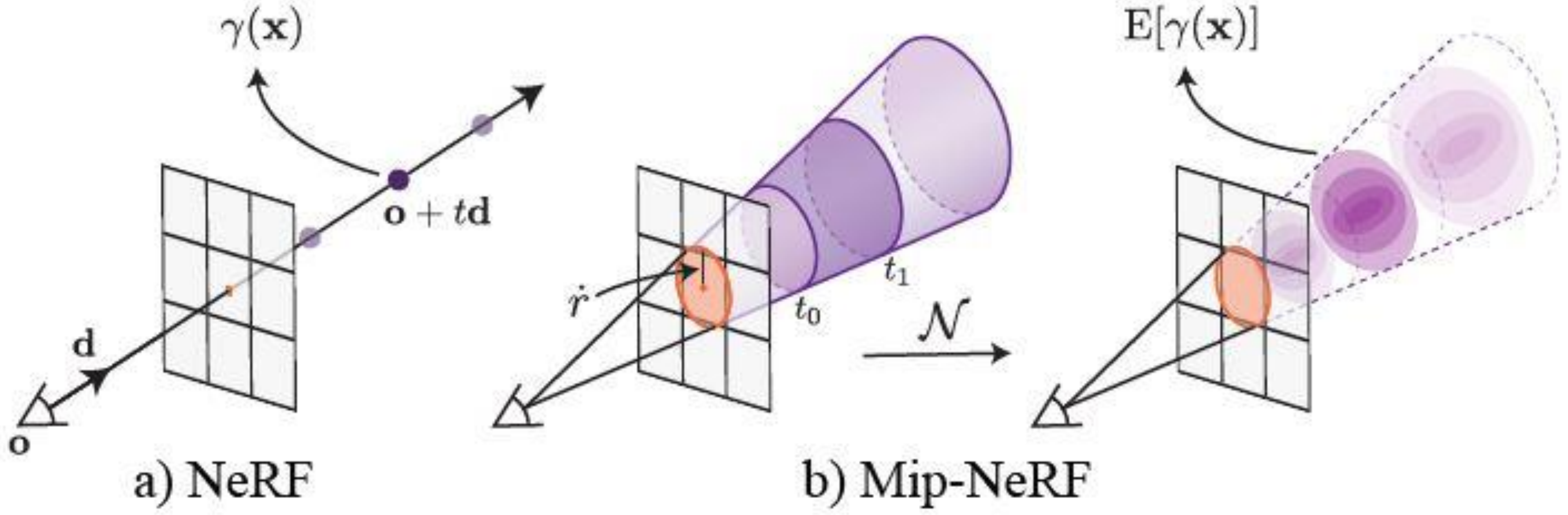
(2) Coordinate-based Neural Representations

- represents 3D scenes as continuous functions parameterized by MLPs (that map from a 3D coordinate to properties of the scene at that location)

- anti-aliased using supersampling, which exacerbates slow rendering procedure.

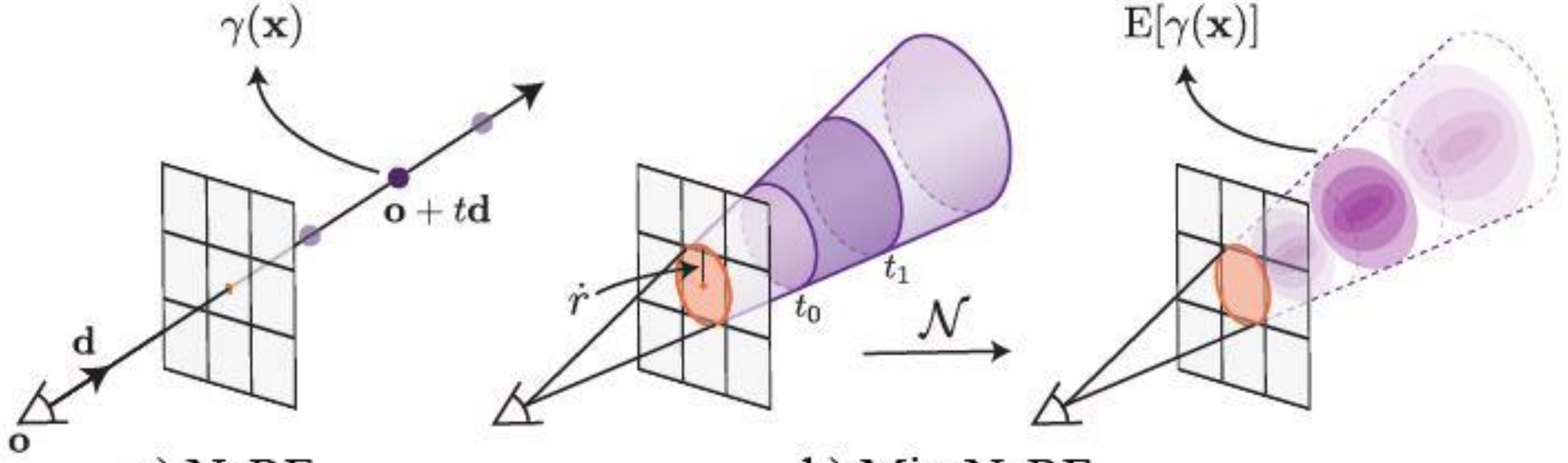
- anti-alised using multiscale representation based on sparse voxel octrees (but requires the scene geometry be known a priori.)

Part 3 Mip-NeRF



Must learn a prefiltered representation of the scene during training.

Mip-NeRF



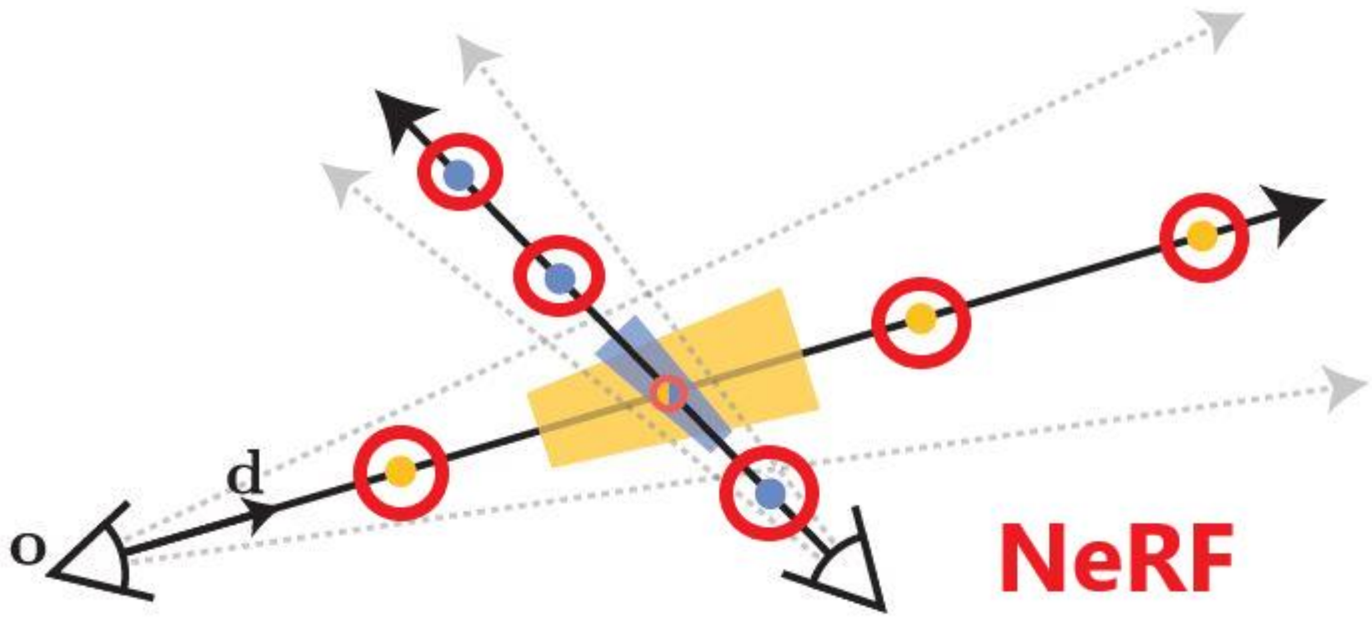
a) NeRF b) Mip-NeRF

Notion of scale is continuous instead of discrete.

Part 4

Key Components of Mip-NeRF

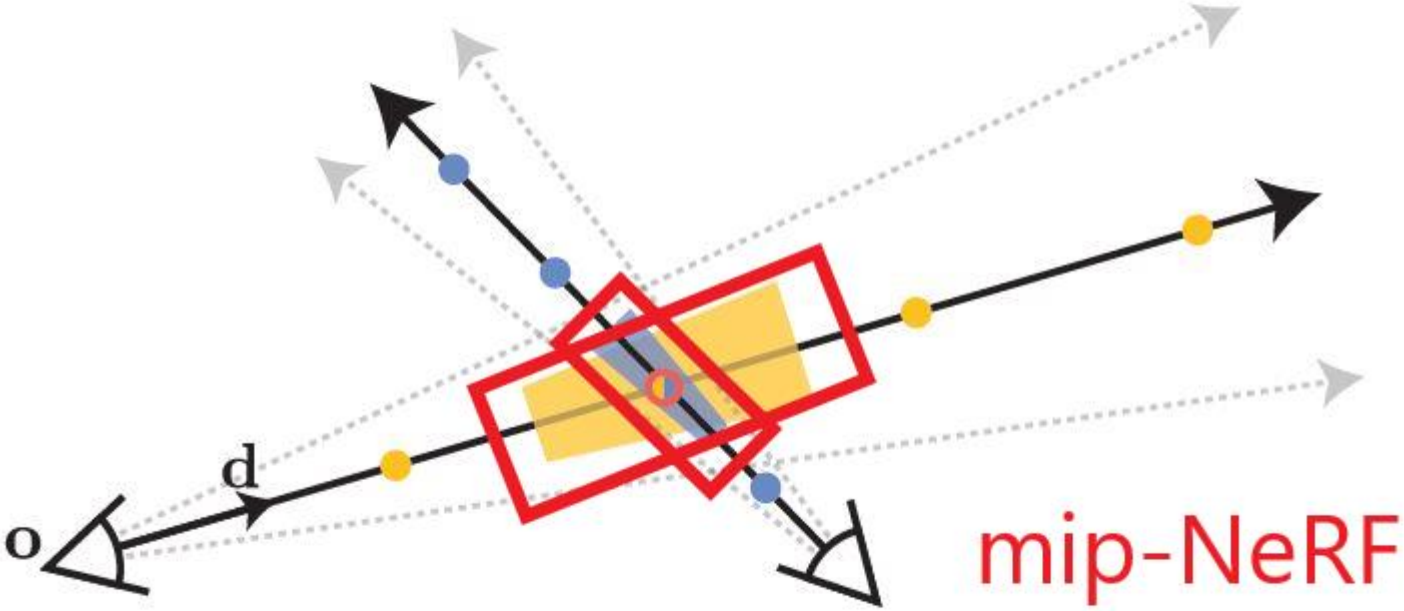
1. Ray Tracing - Cone



NeRF casts a single infinitesimally narrow ray per pixel, resulting in aliasing

Key Components of Mip-NeRF

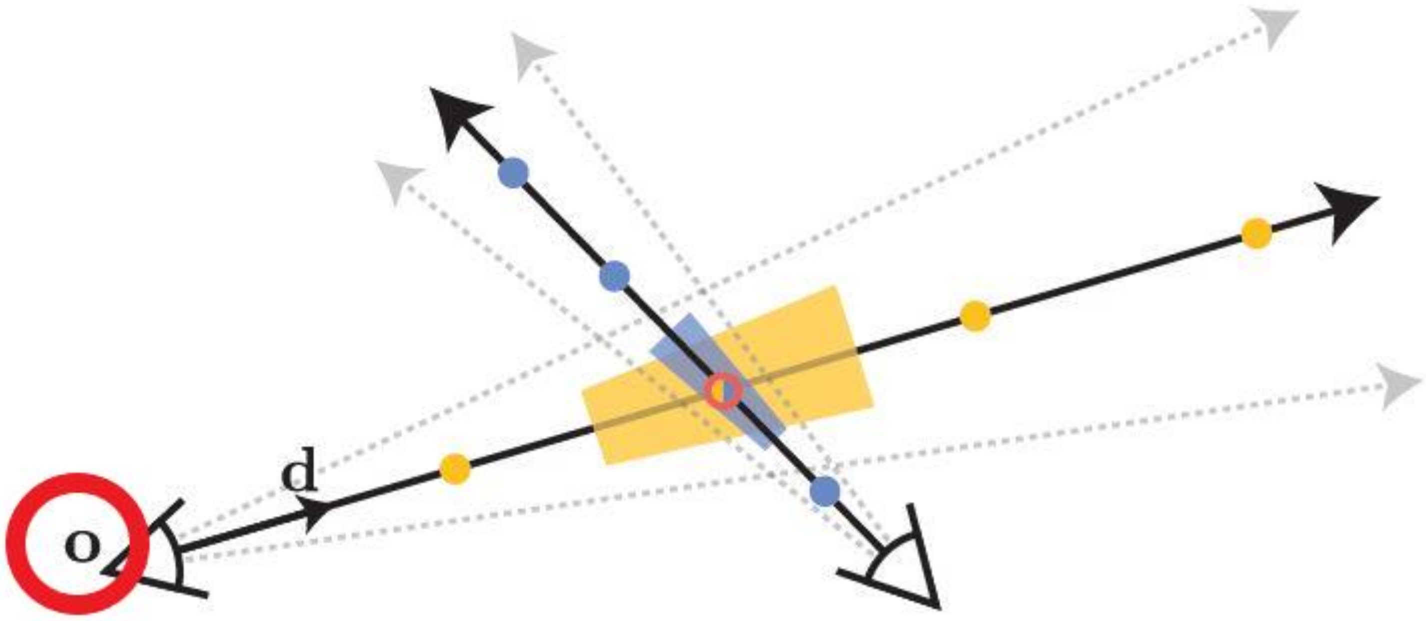
1. Ray Tracing - Cone



Mip-NeRF casts a cone from each pixel, resulting in anti-aliasing.

Key Components of Mip-NeRF

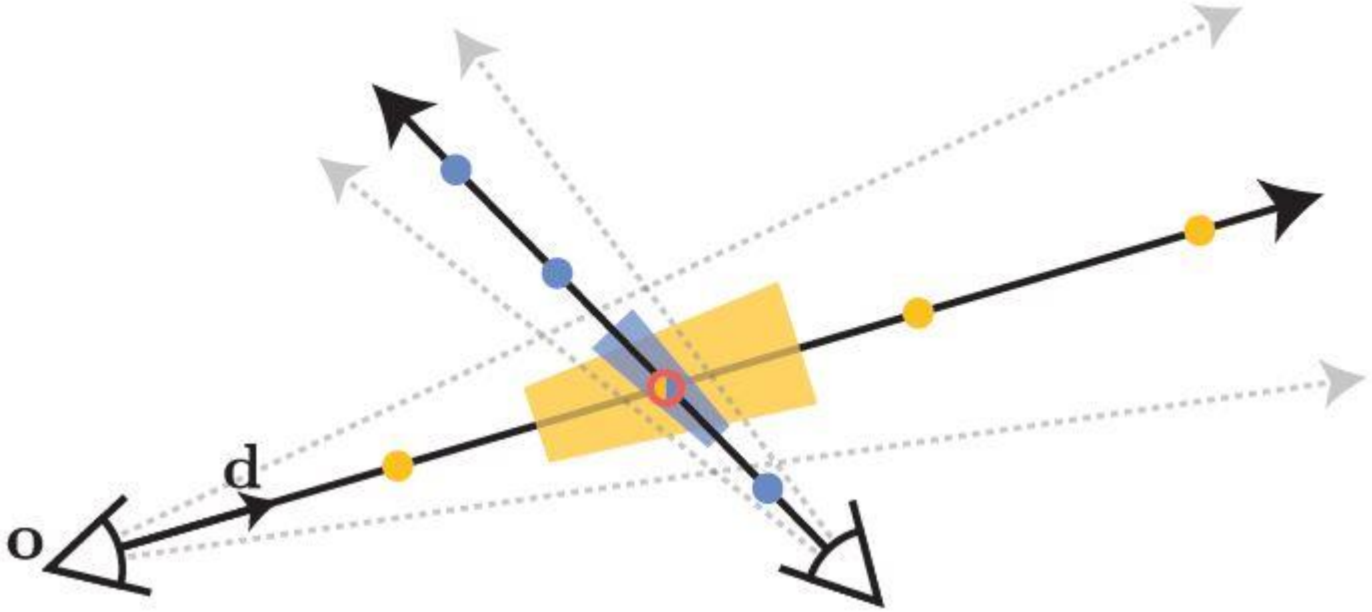
1. Ray Tracing - Cone



O : the camera's center of projection

Part 4 **Key Components of Mip-NeRF**

1. Ray Tracing - Cone



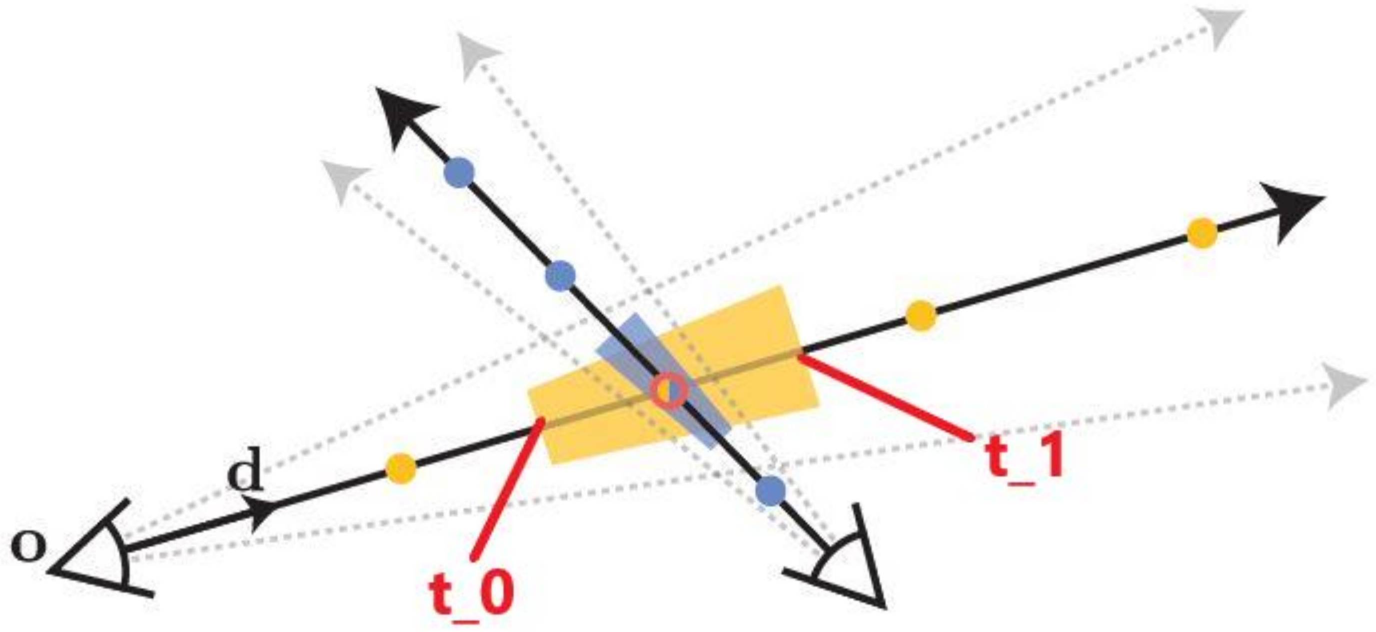
$\dot{r}$ **The radius of the cone at the image plane**

(set to the width of the pixel in world coordinates)

(yields a cone that matches the variance of the pixel's footprint)

Key Components of Mip-NeRF

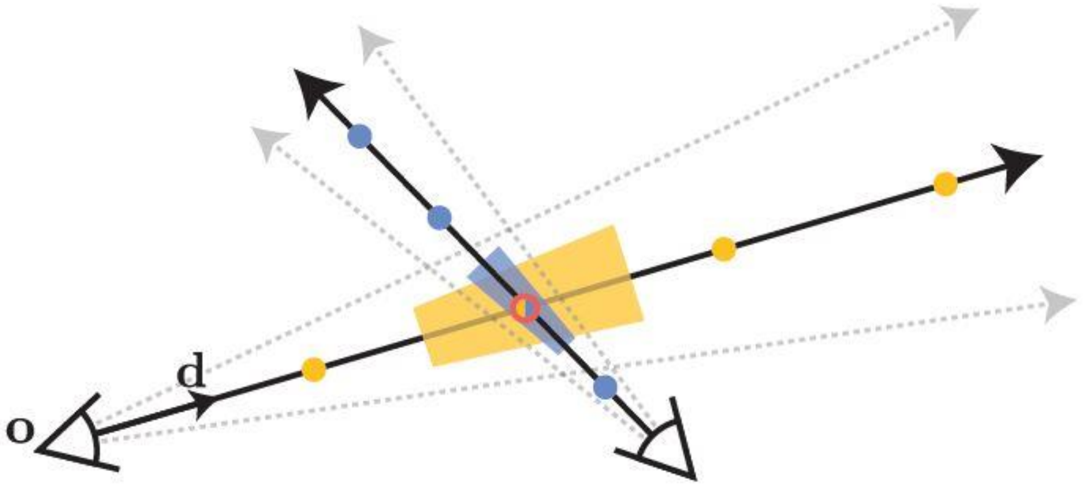
1. Ray Tracing - Cone



$[t_0, t_1]$ **A conical frustum between t values**

Key Components of Mip-NeRF

1. Ray Tracing - Cone



$$\mathrm{F}(\mathbf{x}, \mathbf{o}, \mathbf{d}, \dot{r}, t_0, t_1) = \mathbb{1}\left\{\left(t_0 < \frac{\mathbf{d}^{\mathrm{T}}(\mathbf{x}-\mathbf{o})}{\|\mathbf{d}\|_2^2} < t_1\right) \wedge \left(\frac{\mathbf{d}^{\mathrm{T}}(\mathbf{x}-\mathbf{o})}{\|\mathbf{d}\|_2 \|\mathbf{x}-\mathbf{o}\|_2} > \frac{1}{\sqrt{1+(\dot{r}/\|\mathbf{d}\|_2)^2}}\right)\right\},$$

Key Components of Mip-NeRF

2. Integrated Positional Encoding

- We approximate the conical frustum with a multivariate Gaussian, which allows for an efficient approximation to the desired feature.

- We must compute the mean and variance of multivariate Gaussian.

- Then, we derive the IPE (Integrated Positional Encoding), which is the expectation of a positionally-encoded coordinate distributed according to the aforementioned Gaussian.

Key Components of Mip-NeRF

2. Integrated Positional Encoding

$$\begin{aligned}\gamma(\boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \mathrm{E}_{\mathbf{x}\sim\mathcal{N}(\boldsymbol{\mu}_\gamma, \boldsymbol{\Sigma}_\gamma)}[\gamma(\mathbf{x})] \\ &= \begin{bmatrix} \sin(\boldsymbol{\mu}_\gamma) \circ \exp(-({}^1\!/\!{}_2)\,\mathrm{diag}(\boldsymbol{\Sigma}_\gamma)) \\ \cos(\boldsymbol{\mu}_\gamma) \circ \exp(-({}^1\!/\!{}_2)\,\mathrm{diag}(\boldsymbol{\Sigma}_\gamma)) \end{bmatrix}\end{aligned}$$

IPE feature

as the expected sines and cosines of the mean and the diagonal of the covariance matrix

※ Please refer to the paper for the detailed derivation process of the equation.

Part 4 # Key Components of Mip-NeRF

2. Integrated Positional Encoding

$$\gamma(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathrm{E}_{\mathbf{x}\sim\mathcal{N}(\boldsymbol{\mu}_\gamma, \boldsymbol{\Sigma}_\gamma)}[\gamma(\mathbf{x})]$$
$$= \begin{bmatrix} \sin(\boldsymbol{\mu}_\gamma) \circ \exp(-(1/2)\,\mathrm{diag}(\boldsymbol{\Sigma}_\gamma)) \\ \cos(\boldsymbol{\mu}_\gamma) \circ \exp(-(1/2)\,\mathrm{diag}(\boldsymbol{\Sigma}_\gamma)) \end{bmatrix}$$

$$\mathrm{diag}(\boldsymbol{\Sigma}_\gamma) = \left[\mathrm{diag}(\boldsymbol{\Sigma}), 4\,\mathrm{diag}(\boldsymbol{\Sigma}), \ldots, 4^{L-1}\,\mathrm{diag}(\boldsymbol{\Sigma})\right]^{\mathrm{T}}$$

- $\boldsymbol{\Sigma}_\gamma$ is prohibitively expensive to compute due its relative size
- So, we directly compute the diagonal of $\boldsymbol{\Sigma}_\gamma$.

Key Components of Mip-NeRF

2. Integrated Positional Encoding

$$\gamma(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathrm{E}_{\mathbf{x}\sim\mathcal{N}(\boldsymbol{\mu}_\gamma, \boldsymbol{\Sigma}_\gamma)}[\gamma(\mathbf{x})]$$
$$= \begin{bmatrix} \sin(\boldsymbol{\mu}_\gamma) \circ \exp(-(1/2)\,\mathrm{diag}(\boldsymbol{\Sigma}_\gamma)) \\ \cos(\boldsymbol{\mu}_\gamma) \circ \exp(-(1/2)\,\mathrm{diag}(\boldsymbol{\Sigma}_\gamma)) \end{bmatrix}$$

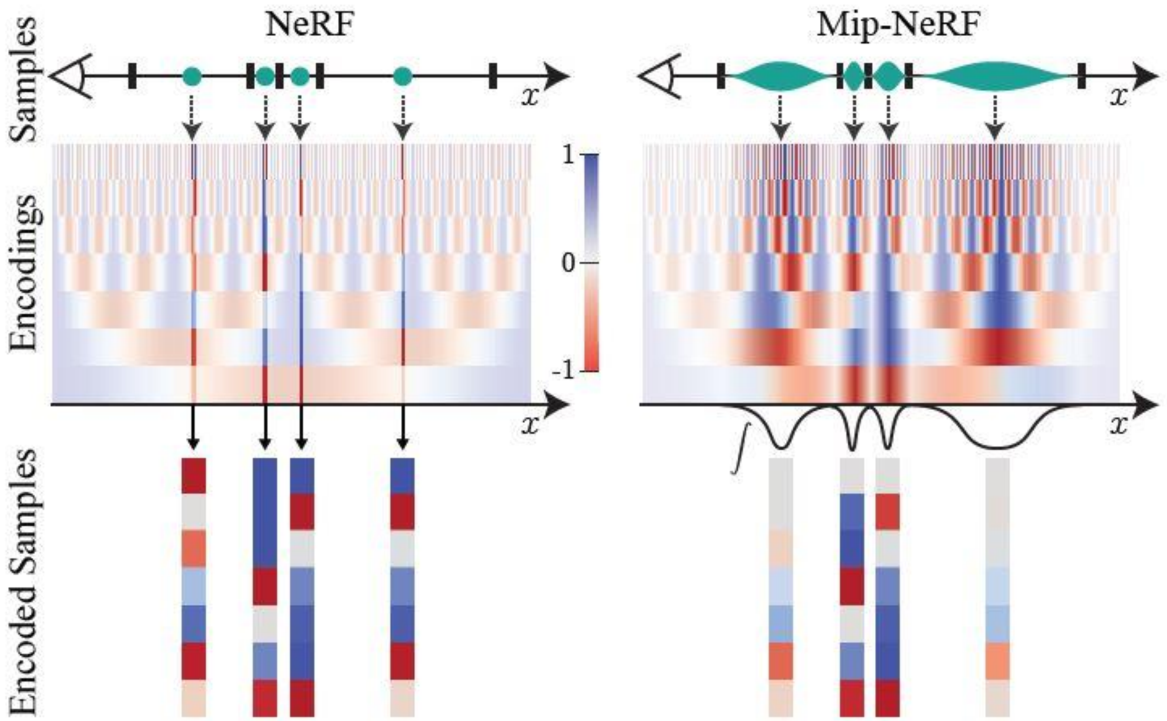
$$\mathrm{diag}(\boldsymbol{\Sigma}_\gamma) = \left[\underline{\mathrm{diag}(\boldsymbol{\Sigma})}, 4\,\underline{\mathrm{diag}(\boldsymbol{\Sigma})}, \ldots, 4^{L-1}\,\underline{\mathrm{diag}(\boldsymbol{\Sigma})}\right]^{\mathrm{T}}$$

$$\underline{\mathrm{diag}(\boldsymbol{\Sigma})} = \sigma_t^2(\mathbf{d}\circ\mathbf{d}) + \sigma_r^2\left(\mathbf{1} - \frac{\mathbf{d}\circ\mathbf{d}}{\|\mathbf{d}\|_2^2}\right)$$

- σ_t^2 : the variance along the ray
- σ_r^2 : the variance perpendicular to the ray
- $\circ$: element-wise multiplication

Key Components of Mip-NeRF

2. Integrated Positional Encoding

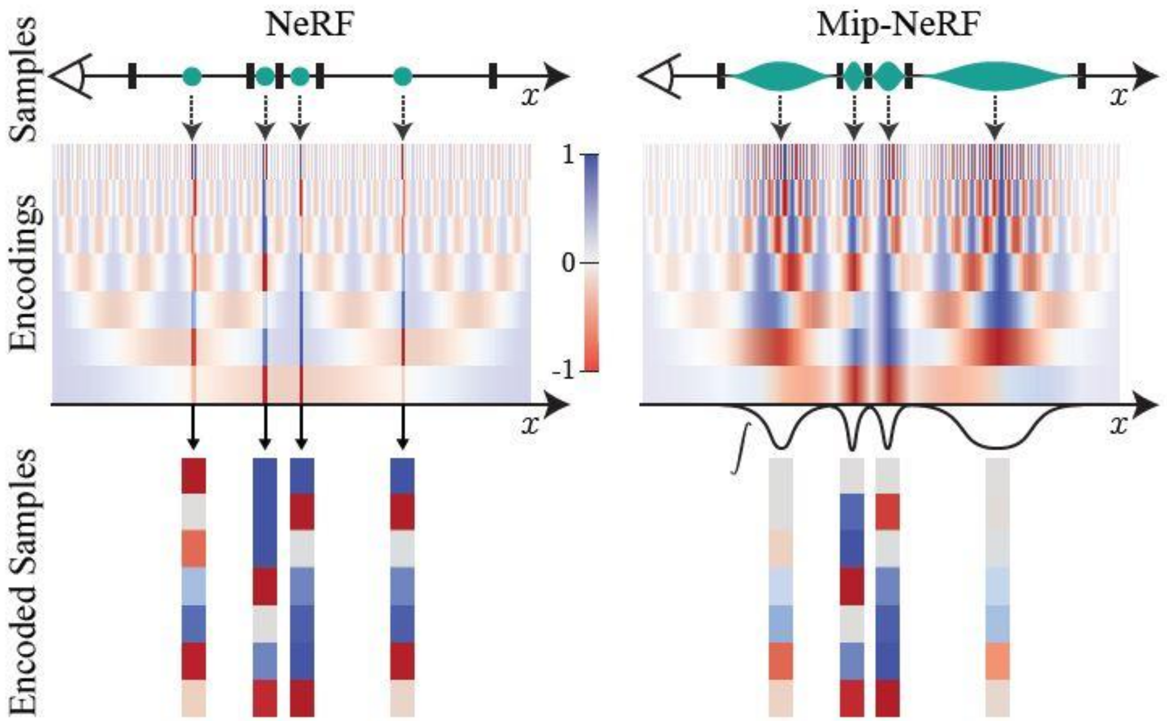


PE preserves all frequencies up to some manually-tuned hyperparameter L.

Part 4

Key Components of Mip-NeRF

2. Integrated Positional Encoding

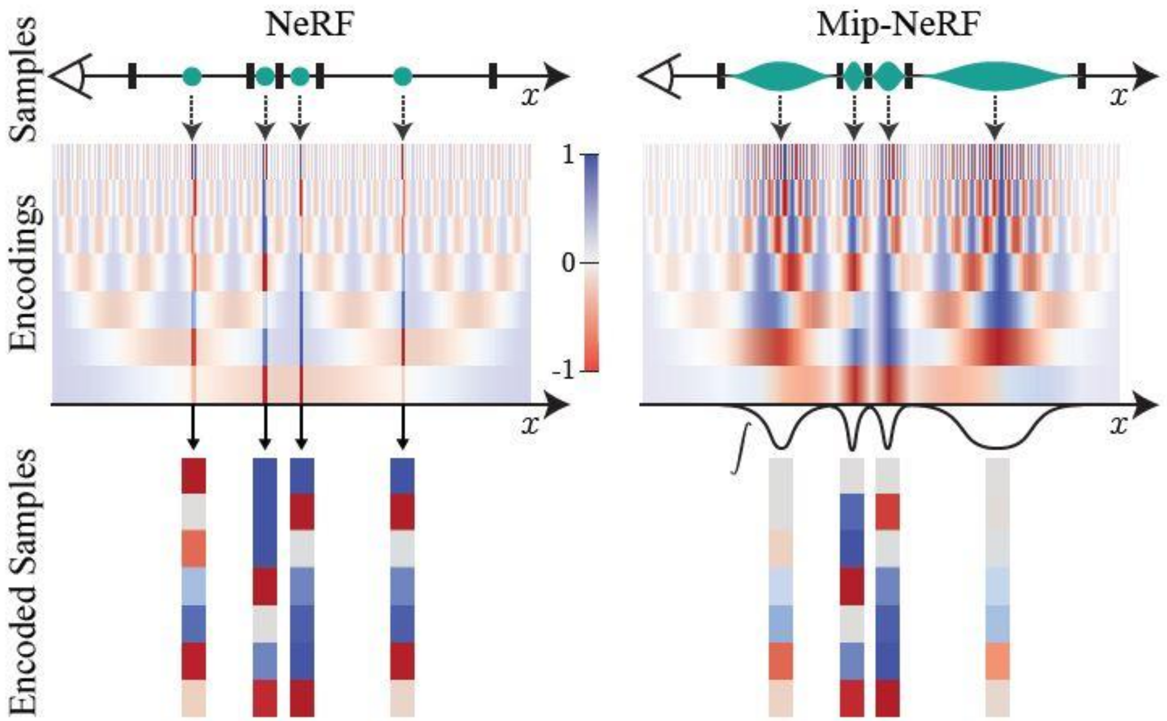


IPE preserves frequencies that are constant over an interval,

removes frequencies that vary over an interval softly.

Key Components of Mip-NeRF

2. Integrated Positional Encoding



IPE features are effectively anti-aliased positional encoding features that smoothly encode the size and shape of a volume of space.

Key Components of Mip-NeRF

3. Single Network

- Model size is cut in half.

- Renderings are more accurate.

- Sampling is more efficient.

Key Components of Mip-NeRF

3. Single Network

[Optimization Problem]

$$\min_{\Theta} \sum_{\mathbf{r}\in\mathcal{R}} \left(\lambda\|\mathbf{C}^*(\mathbf{r})-\mathbf{C}(\mathbf{r};\Theta,\mathbf{t}^c)\|_2^2+\|\mathbf{C}^*(\mathbf{r})-\mathbf{C}(\mathbf{r};\Theta,\mathbf{t}^f)\|_2^2\right)$$

- Coarse loss must be balanced against the fine loss.

- In Mip-NeRF, λ is set to 0.1. (In NeRF, λ is 1.)

Experiments

Evaluation Metrics

- PSNR, SSIM, and LPIPS

- Also use an “average” error metric that summarizes 3 metrics.

$$\mathrm{MSE} = 10^{-\mathrm{PSNR}/10}$$

$$\sqrt{1 - \mathrm{SSIM}}$$

$$\mathrm{LPIPS}$$

Experiments

Multiscale Blender Dataset

- Straightforward modification to NeRF’s Blender Dataset

- Designed to probe aliasing and scale-space reasoning

- Ablation study on Misc, Single MLP, Area Loss, and IPE

Part 5

Experiments

Multiscale Blender Dataset

	PSNR ↑				SSIM ↑				LPIPS ↓						
	Full Res.	1/2 Res.	1/4 Res.	1/8 Res.	Full Res.	1/2 Res.	1/4 Res.	1/8 Res.	Full Res.	1/2 Res.	1/4 Res.	1/8 Res	Avg. ↓	Time (hours)	# Params
NeRF (Jax Impl.) [11, 30]	31.196	30.647	26.252	22.533	0.9498	0.9560	0.9299	0.8709	0.0546	0.0342	0.0428	0.0750	0.0288	3.05 ± 0.04	1,191K
NeRF + Area Loss	27.224	29.578	29.445	25.039	0.9113	0.9394	0.9524	0.9176	0.1041	0.0677	0.0406	0.0469	0.0305	3.03 ± 0.03	1,191K
NeRF + Area, Centered Pixels	29.893	32.118	33.399	29.463	0.9376	0.9590	0.9728	0.9620	0.0747	0.0405	0.0245	0.0398	0.0191	3.02 ± 0.05	1,191K
NeRF + Area, Center, Misc.	29.900	32.127	33.404	29.470	0.9378	0.9592	0.9730	0.9622	0.0743	0.0402	0.0243	0.0394	0.0190	2.94 ± 0.02	1,191K
Mip-NeRF	32.629	34.336	35.471	35.602	0.9579	0.9703	0.9786	0.9833	0.0469	0.0260	0.0168	0.0120	0.0114	2.84 ± 0.01	612K
Mip-NeRF w/o Misc.	32.610	34.333	35.497	35.638	0.9577	0.9703	0.9787	0.9834	0.0470	0.0259	0.0167	0.0120	0.0114	2.82 ± 0.03	612K
Mip-NeRF w/o Single MLP	32.401	34.131	35.462	35.967	0.9566	0.9693	0.9780	0.9834	0.0479	0.0268	0.0169	0.0116	0.0115	3.40 ± 0.01	1,191K
Mip-NeRF w/o Area Loss	33.059	34.280	33.866	30.714	0.9605	0.9704	0.9747	0.9679	0.0427	0.0256	0.0213	0.0308	0.0139	2.82 ± 0.01	612K
Mip-NeRF w/o IPE	29.876	32.160	33.679	29.647	0.9384	0.9602	0.9742	0.9633	0.0742	0.0393	0.0226	0.0378	0.0186	2.79 ± 0.01	612K

- Misc : whether to add small changes that slightly improve the stability of training
- Single MLP : whether to use NeRF's training scheme or Mip-NeRF's scheme
- Area Loss : whether to add the loss scaling by pixel area
- IPE : whether to use positional encoding or integrated positional encoding

Part 6
Conclusions

- Model : multiscale NeRF-like model that addresses the inherent aliasing of NeRF

- Ray casting : cones

- Encoding : positions and sizes of conical frustums (IPE)

- Training : single neural network that models the scene at multiple scales