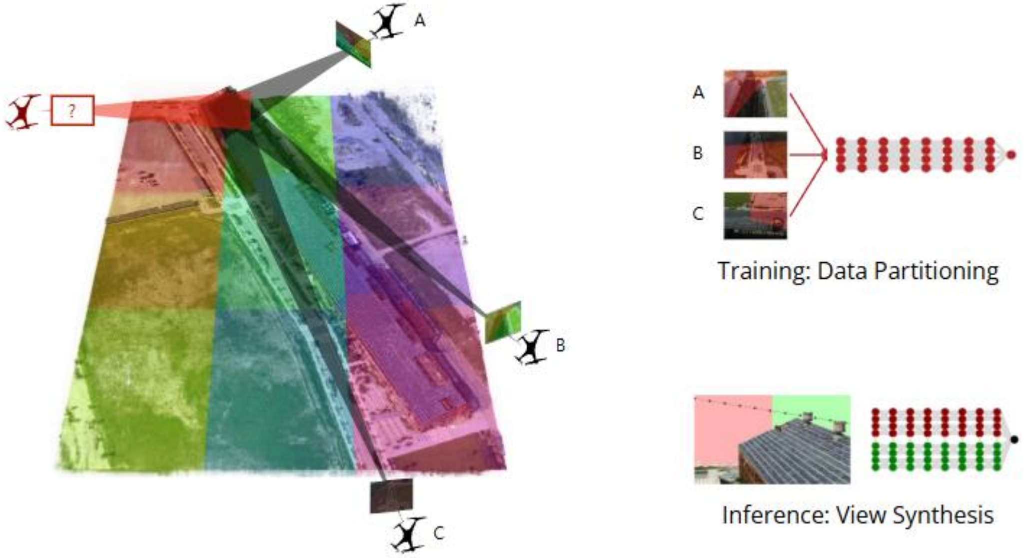
Mega-NeRF: Scalable Construction of Large-Scale NeRFs for Virtual Fly-Throughs

(CVPR 2022)



한국과학기술연구원(KIST)

CVIPL 학생연구원 **김연욱**

한국과학기술연구원
Korea Institute of Science and Technology

Contents

Author

Mega-NeRF:
Scalable Construction of Large-Scale NeRFs for Virtual Fly-Throughs

Haithem Turki[1] Deva Ramanan[1,2] Mahadev Satyanarayanan[1]
[1]Carnegie Mellon University [3]Argo AI

Deva Ramanan

팔로우

Professor, Robotics Institute, Carnegie Mellon University
cs.cmu.edu의 이메일 확인됨 - 홈페이지
Computer Vision Machine Learning

제목	인용	연도
Microsoft coco: Common objects in context TY Lin, M Maire, S Belongie, J Hays, P Perona, D Ramanan, P Dollár, ... European conference on computer vision, 740-755	63300	2014
Object detection with discriminatively trained part-based models PF Felzenszwalb, RB Girshick, D McAllester, D Ramanan IEEE transactions on pattern analysis and machine intelligence 32 (9), 1627-1645	13643	2009
A discriminatively trained, multiscale, deformable part model P Felzenszwalb, D McAllester, D Ramanan 2008 IEEE conference on computer vision and pattern recognition, 1-8	4164	2008
Depth-supervised nerf: Fewer views and faster training for free K Deng, A Liu, JY Zhu, D Ramanan Proceedings of the IEEE/CVF conference on computer vision and pattern ...	1141	2022

Depth-Supervised Nerf
나온 이후 나온 논문

Mega-NeRF:
Scalable Construction of Large-Scale NeRFs for Virtual Fly-Throughs

Haithem Turki[1] Deva Ramanan[1,2] Mahadev Satyanarayanan[1]
[1]Carnegie Mellon University [2]Argo AI

Haithem Turki

팔로우

Research Scientist, NVIDIA
nvidia.com의 이메일 확인됨 - 홈페이지

제목	인용	연도
Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs H Turki, D Ramanan, M Satyanarayanan Proceedings of the IEEE/CVF conference on computer vision and pattern …	488	2022
Suds: Scalable urban dynamic scenes H Turki, JY Zhang, F Ferroni, D Ramanan Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern …	152	2023
Learning specific-class segmentation from diverse data MP Kumar, H Turki, D Preston, D Koller 2011 International conference on computer vision, 1800-1807	112	2011

Mahadev Satyanarayanan

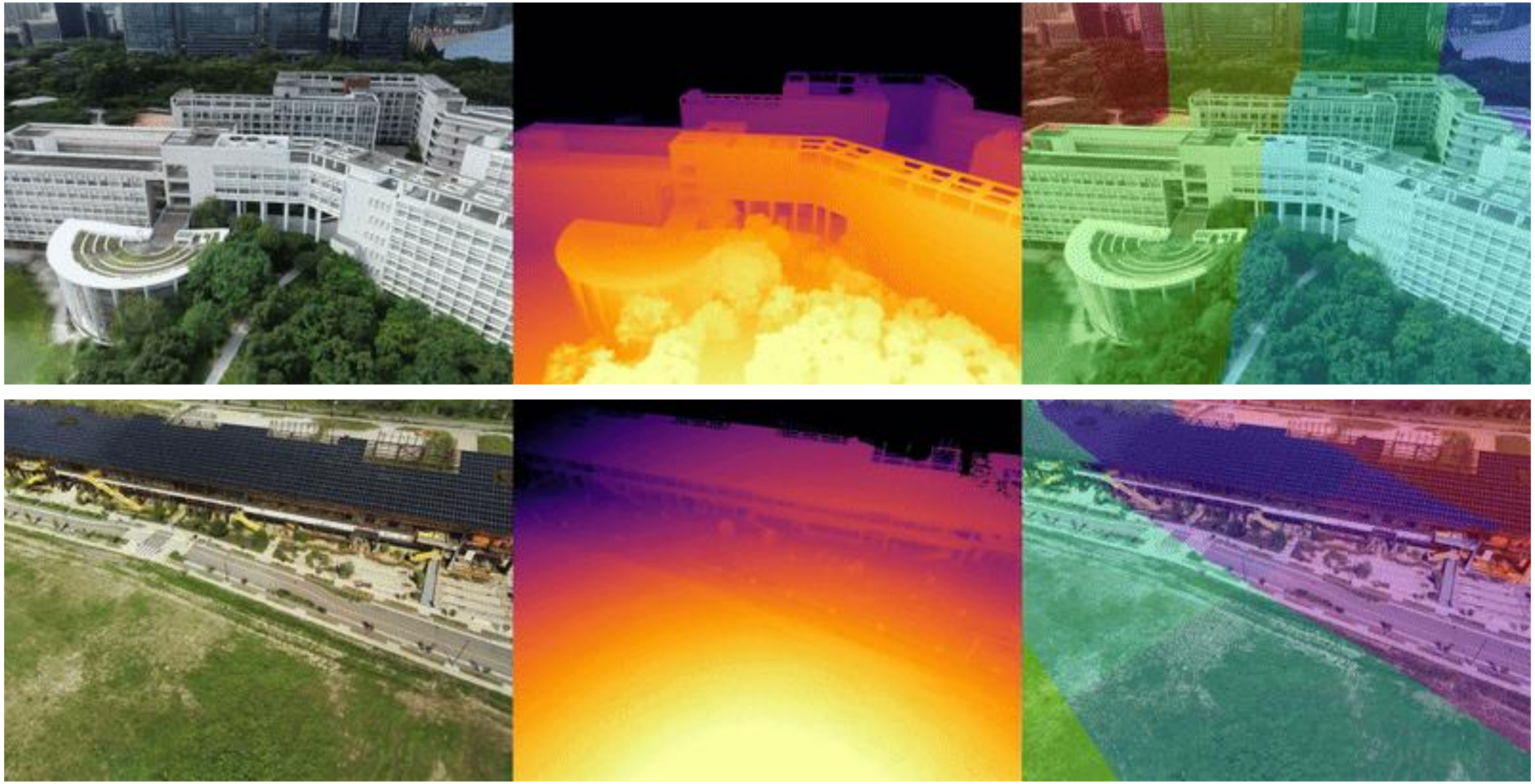
팔로우

Jaime Carbonell University Professor of Computer Science, Carnegie Mellon University
cs.cmu.edu의 이메일 확인됨 - 홈페이지
edge computing mobile computing Internet of Things pervasive computing distributed systems

제목	인용	연도
The case for vm-based cloudlets in mobile computing M Satyanarayanan, P Bahl, R Caceres, N Davies IEEE pervasive Computing 8 (4), 14-23	4994	2009
Pervasive computing: Vision and challenges M Satyanarayanan IEEE Personal communications 8 (4), 10-17	4015	2002
The emergence of edge computing M Satyanarayanan Computer 50 (1), 30-39	3245	2017

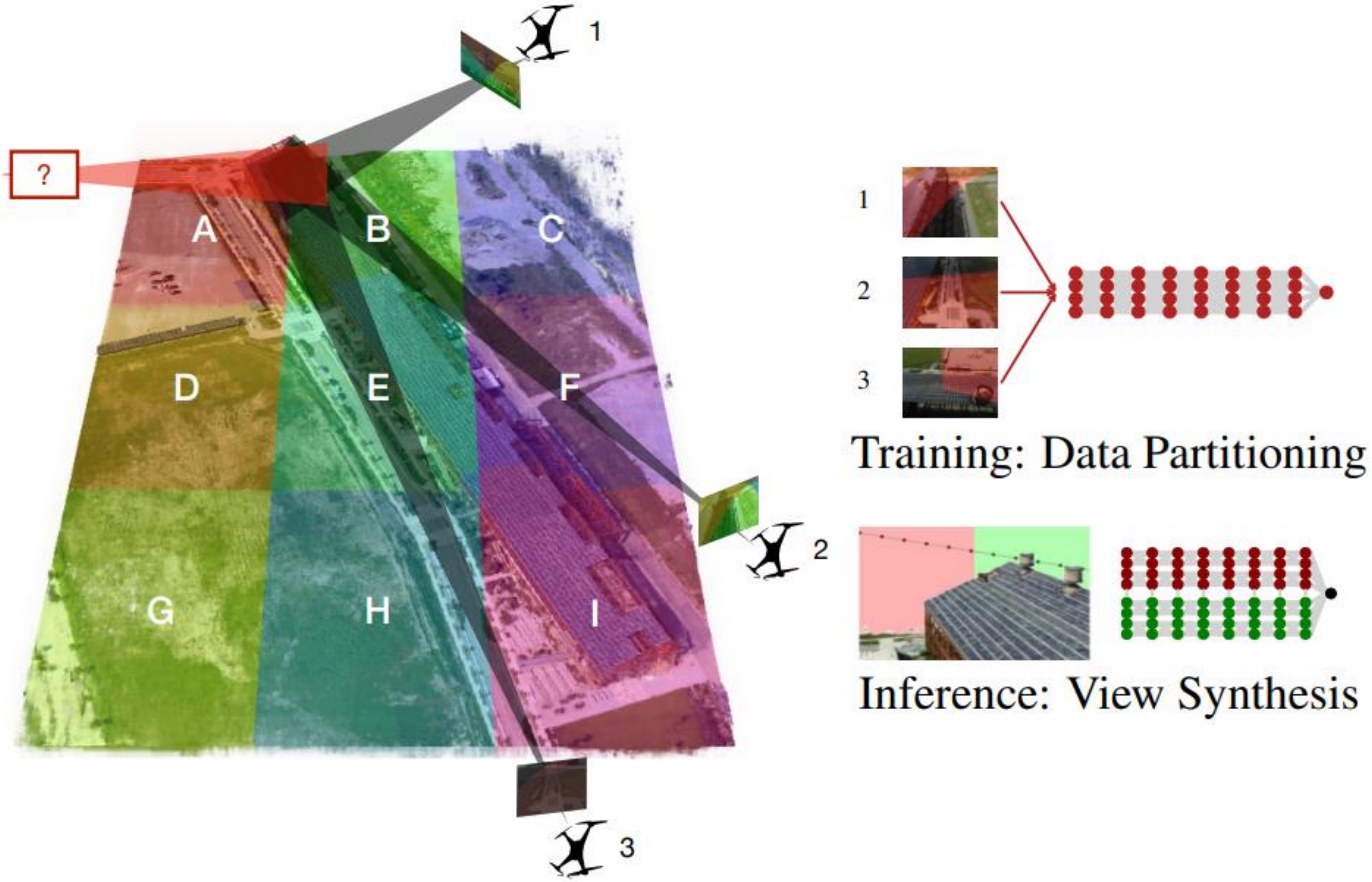
OverView

OverView



Reconstructed Image Depth Map Submodule Visualization

OverView



1
?
A
B
C
D
E
F
G
H
I
2
3
1
2
3
Training: Data Partitioning
Inference: View Synthesis

Introduction

Introduction

Search and rescue (SAR) :

search for and provision of aid to people who are in distress or imminent danger.

기존 Drone based SAR 문제점:

- 배터리 부족
- 2D "Birds-eye-view" map만 생성
  --> 디테일 정보 부족

2D "Birds-eye-view" map

Introduction

Search and rescue (SAR) :

search for and provision of aid to people who are in distress or imminent danger.

기존 Drone based SAR 문제점:

- 배터리 부족
- 2D "Birds-eye-view" map만 생성
  --> 디테일 정보 부족

NeRF로 해결!

2D "Birds-eye-view" map

Introduction

- NeRF의 굉장히 많은 후속 연구
- 그러나, 대부분은 실내에서 찍은 single-object scenes에 대한 연구
- NeRF의 가장 큰 데이터셋 Tanks and Temples Dataset의 평균 촬영 공간의 넓이는 Scene 당 463m^2

Introduction

- 저자가 원하는 것은 Urban-Scale environments NeRF
- Urban-Scale environments의 Dataset은 order-of-magnitude more pixels을 포함한다.

	Resolution	# Images	# Pixels/Rays	Scene Captured / Image
Synthetic NeRF - Chair	400 x 400	400	256,000,000	0.271
Synthetic NeRF - Drums	400 x 400	400	256,000,000	0.302
Synthetic NeRF - Ficus	400 x 400	400	256,000,000	0.582
Synthetic NeRF - Hotdog	400 x 400	400	256,000,000	0.375
Synthetic NeRF - Lego	400 x 400	400	256,000,000	0.205
Synthetic NeRF - Materials	400 x 400	400	256,000,000	0.379
Synthetic NeRF - Mic	400 x 400	400	256,000,000	0.518
Synthetic NeRF - Ship	400 x 400	400	256,000,000	0.483
T&T - Barn	1920 x 1080	384	796,262,400	0.135
T&T - Caterpillar	1920 x 1080	368	763,084,800	0.216
T&T - Family	1920 x 1080	152	315,187,200	0.284
T&T - Ignatius	1920 x 1080	263	545,356,800	0.476
T&T - Truck	1920 x 1080	250	518,400,000	0.225
Mill 19 - Building	4608 x 3456	1940	30,894,981,120	0.062
Mill 19 - Rubble	4608 x 3456	1678	26,722,566,144	0.050
Quad 6k	1708 x 1329	5147	11,574,265,679	0.010
UrbanScene3D - Residence	5472 x 3648	2582	51,541,512,192	0.059
UrbanScene3D - Sci-Art	4864 x 3648	3019	53,568,749,568	0.088
UrbanScene3D - Campus	5472 x 3648	5871	117,196,056,576	0.028

Mega-Nerf Dataset

Introduction

- Firstly, applications such as search-and-rescue are time-sensitive.
- Secondly, our datasets are orders of magnitude larger than previously evaluated datasets.
- Finally, existing real-time NeRF renderers are ill-suited for large-scale scenes

Introduction

- Firstly, applications such as search-and-rescue are time-sensitive.

- Secondly, our datasets are orders of magnitude larger than previously evaluated datasets.

해결방법 :

거대한 전체 장면을 여러 조각(Cell)으로 나누어 각 조각(Cell)을 submodules(NeRF)이 전담
+
submodules 병렬로 훈련

Introduction

해결방법 :

spatial locality라는 것을 사용

(설명은 뒤쪽에...)

- Finally, existing real-time NeRF renderers are ill-suited for large-scale scenes

Related Work

Related Work

DeRF

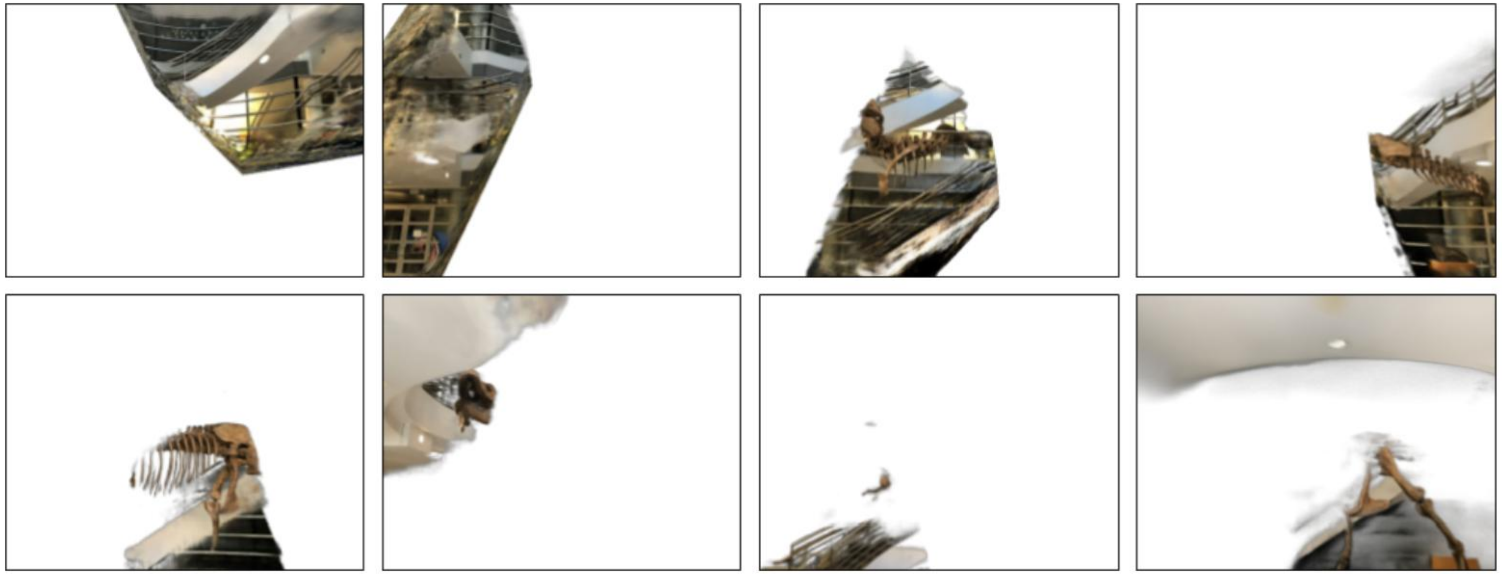


Figure 4. **Decomposed radiance fields** – We visualize each of the rendering heads individually. Note that as each head is rendered *only* the weights of *one* neural network head needs to be loaded, hence resulting in optimal cache coherency while accessing GPU memory.

Related Work

DeRF

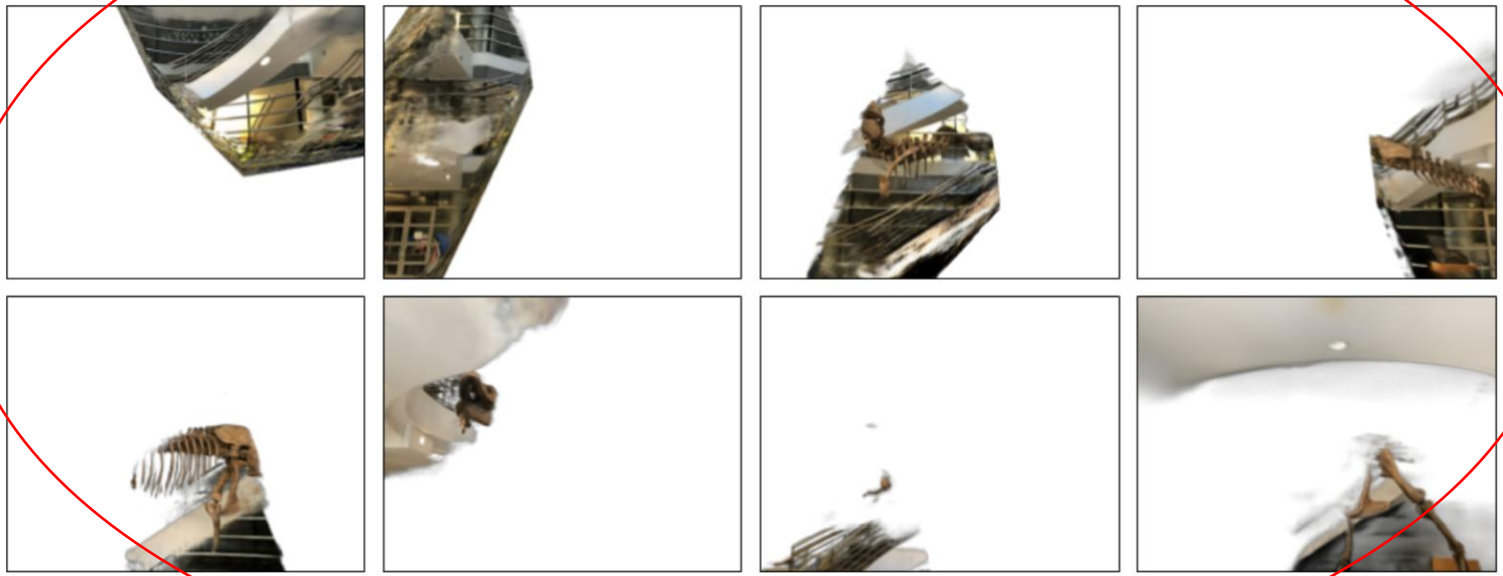


Figure 4. **Decomposed radiance fields** – We visualize each of the rendering heads individually. Note that as each head is rendered *only* the weights of *one* neural network head needs to be loaded, hence resulting in optimal cache coherency while accessing GPU memory.

Related Work

DeRF

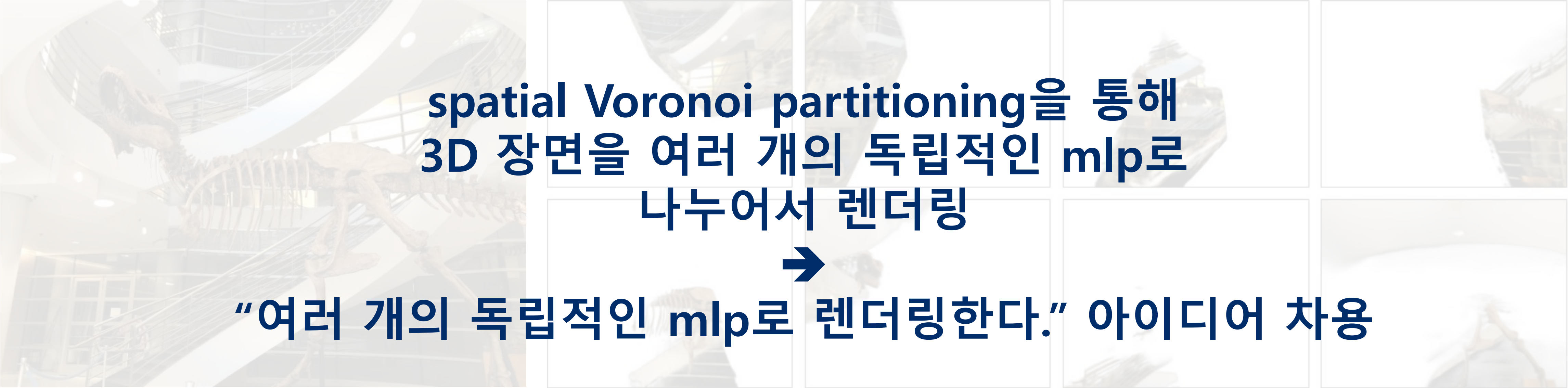
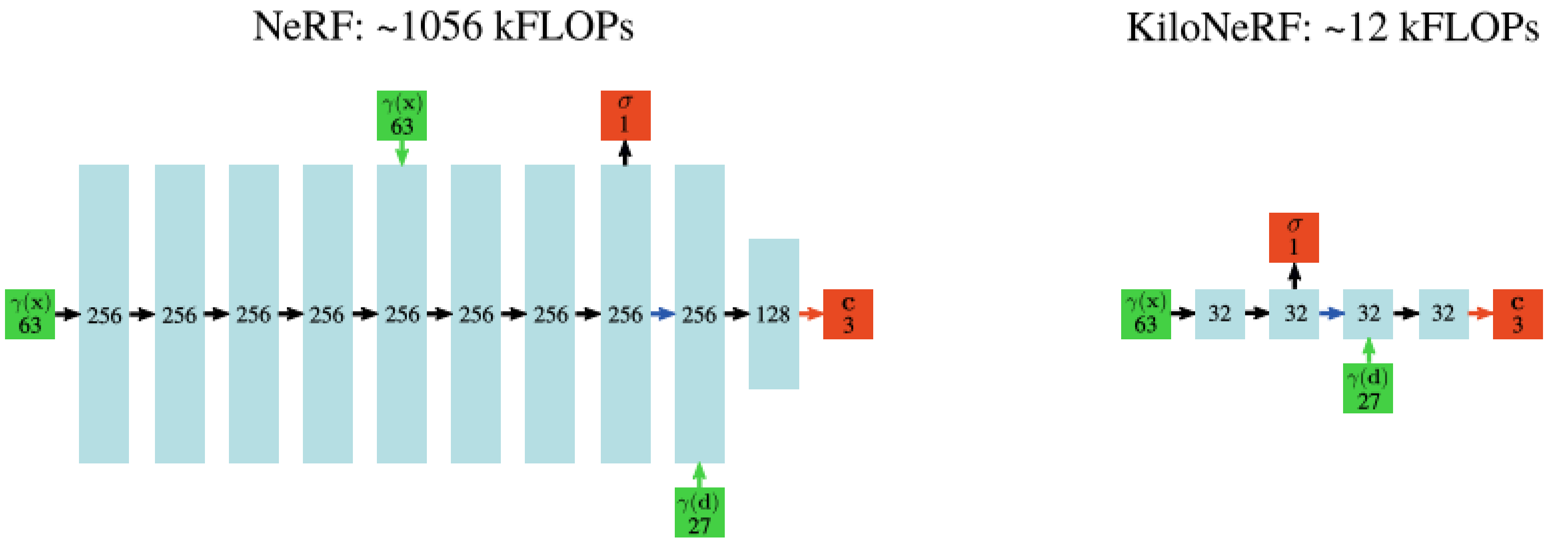


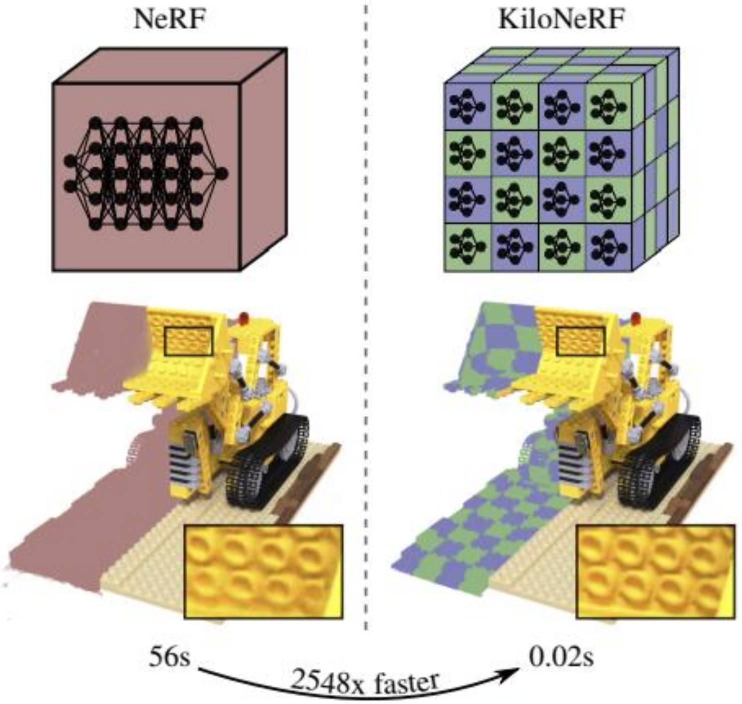
Figure 4. **Decomposed radiance fields** – We visualize each of the rendering heads individually. Note that as each head is rendered *only* the weights of *one* neural network head needs to be loaded, hence resulting in optimal cache coherency while accessing GPU memory.

spatial Voronoi partitioning을 통해
3D 장면을 여러 개의 독립적인 mlp로
나누어서 렌더링
➔
"여러 개의 독립적인 mlp로 렌더링한다." 아이디어 차용

Related Work

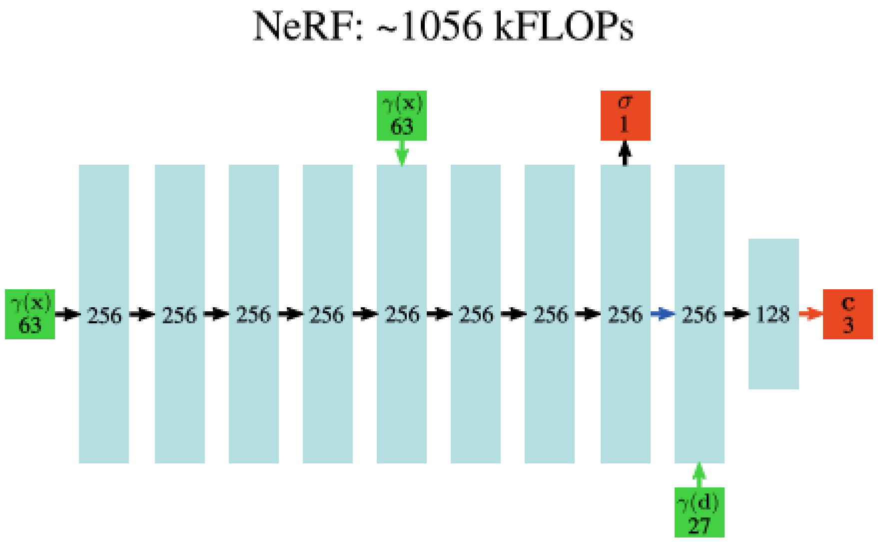
KiloNeRF

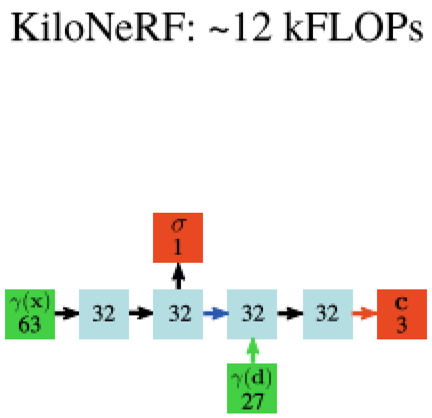


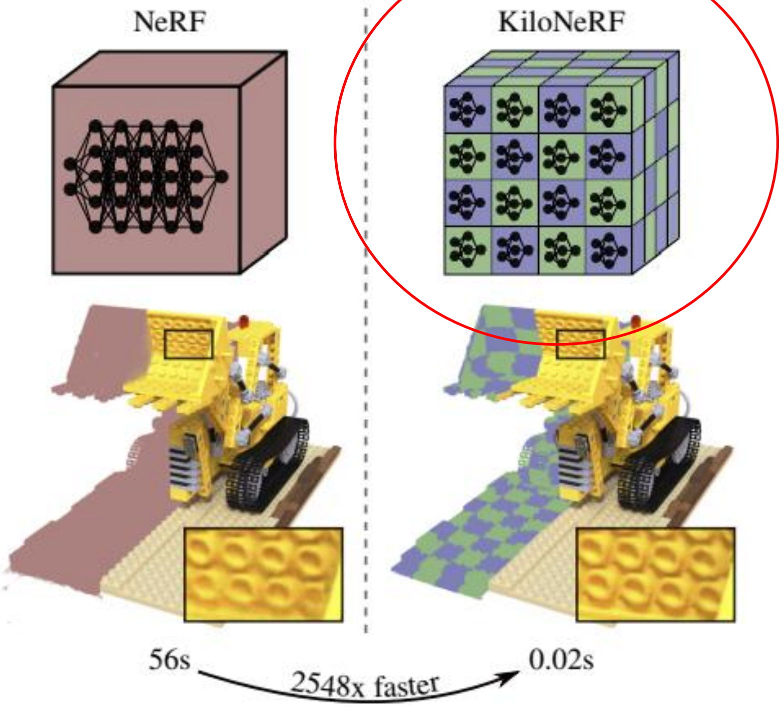


Related Work

KiloNeRF







Related Work

KiloNeRF

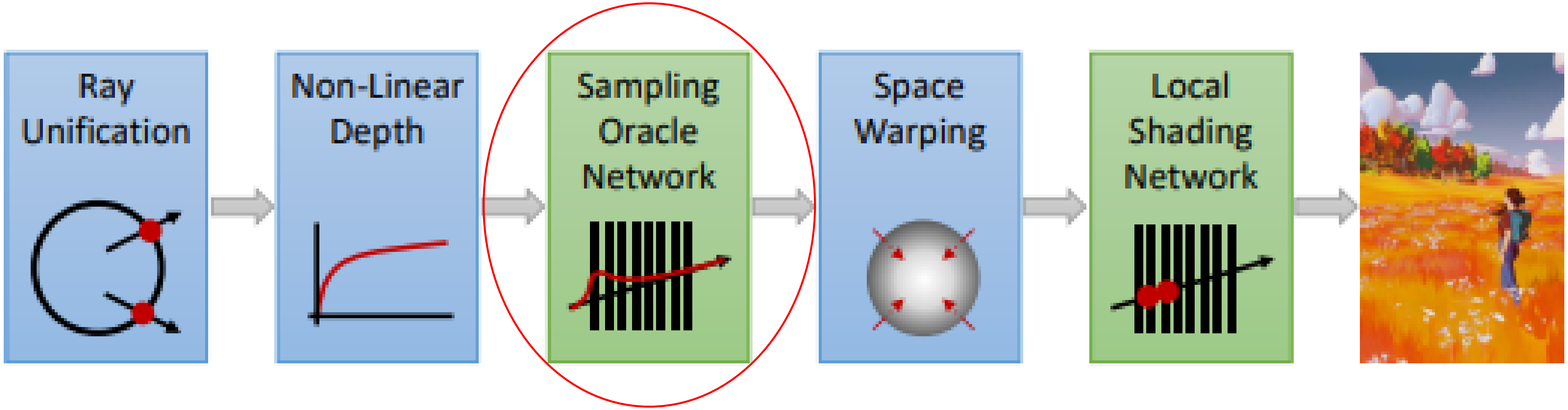
Scene을 수천개로 나누어서 렌더링 한다.

➔

"여러 개의 독립적인 mlp로 나누고 병렬로 학습시킨다."

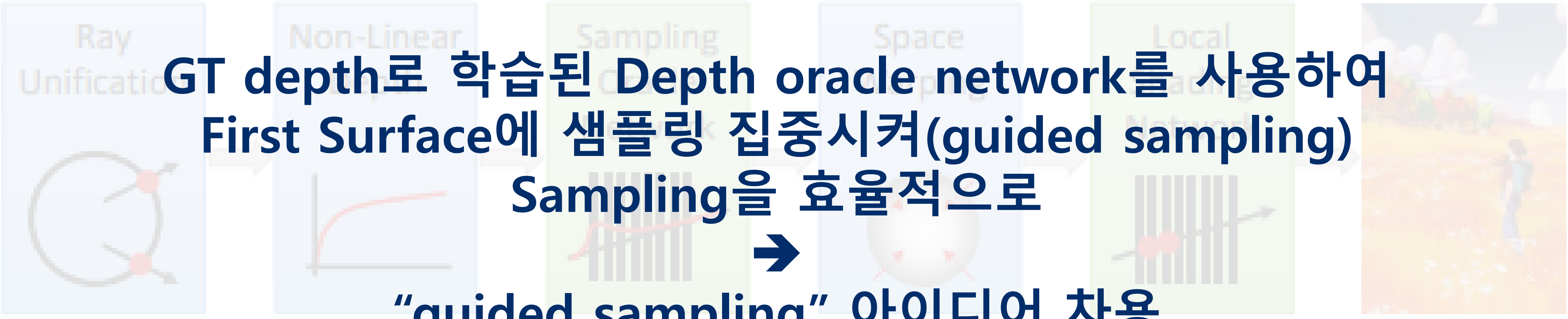
Related Work

DONeRF



Related Work

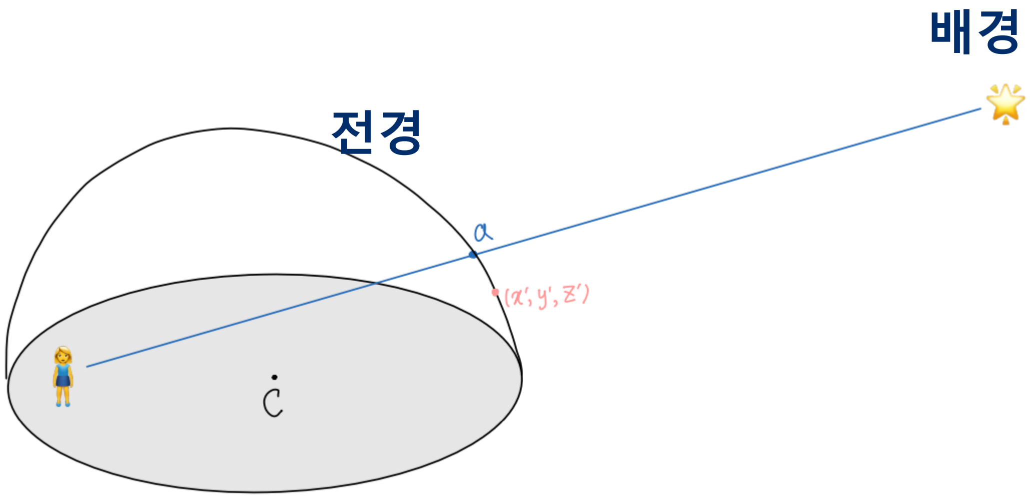
DONeRF

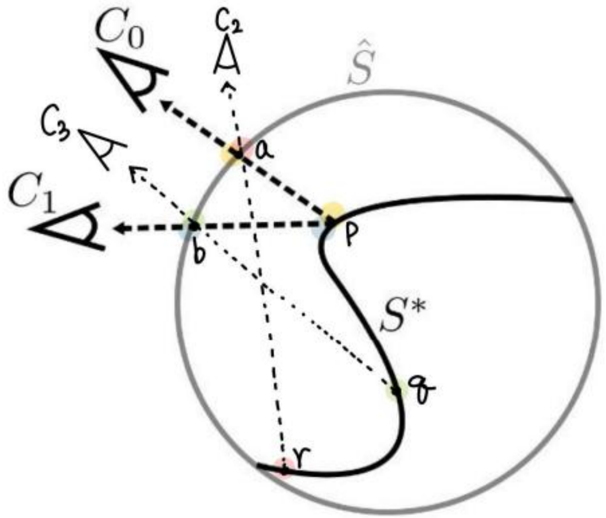


GT depth로 학습된 Depth oracle network를 사용하여
First Surface에 샘플링 집중시켜(guided sampling)
Sampling을 효율적으로
➔
“guided sampling” 아이디어 차용

Related Work

NeRF++

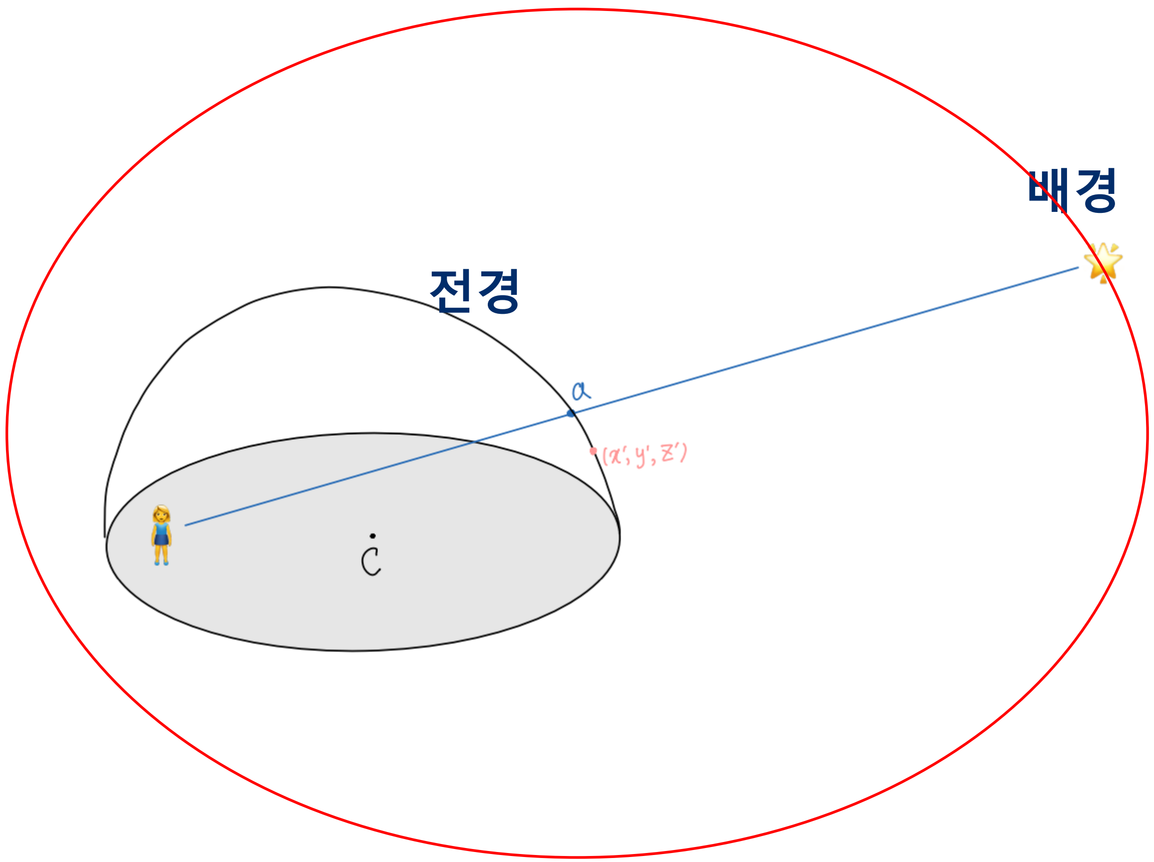


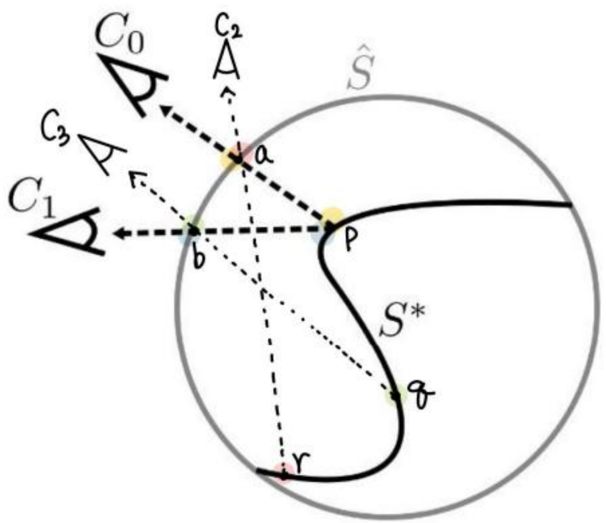


Shape-Radiance Ambiguity

Related Work

NeRF++

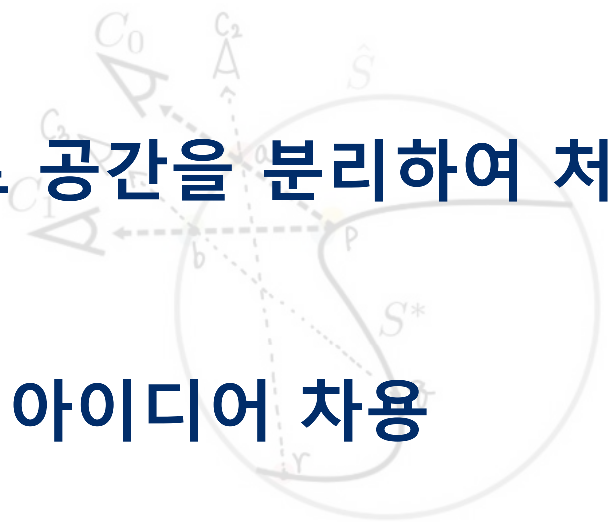




Shape-Radiance Ambiguity

Related Work

NeRF++



Shape-Radiance Ambiguity

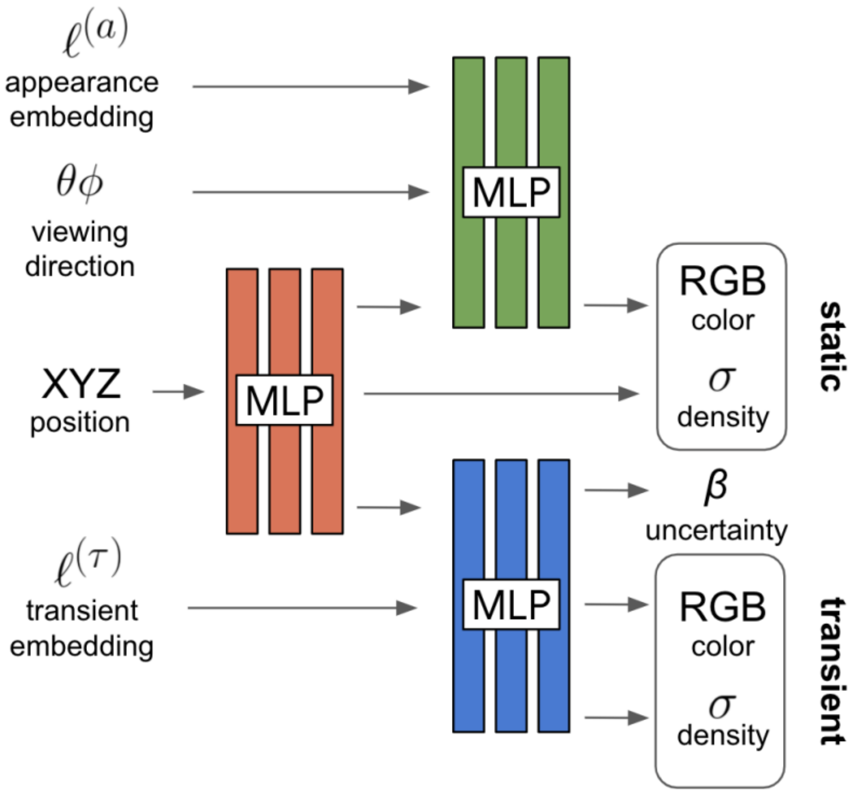
배경이 학습이 잘 안돼서 **전경과 배경으로 공간을 분리하여 처리**

➔

"전경 배경 공간 분리" 아이디어 차용

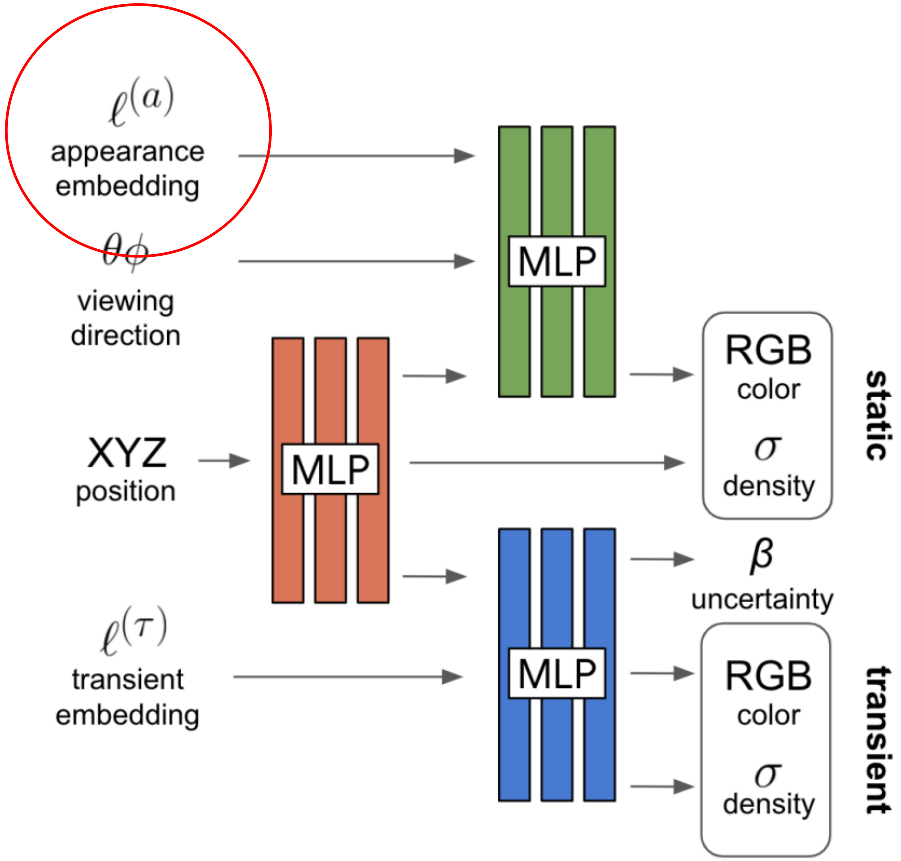
Related Work

NeRF in the Wild



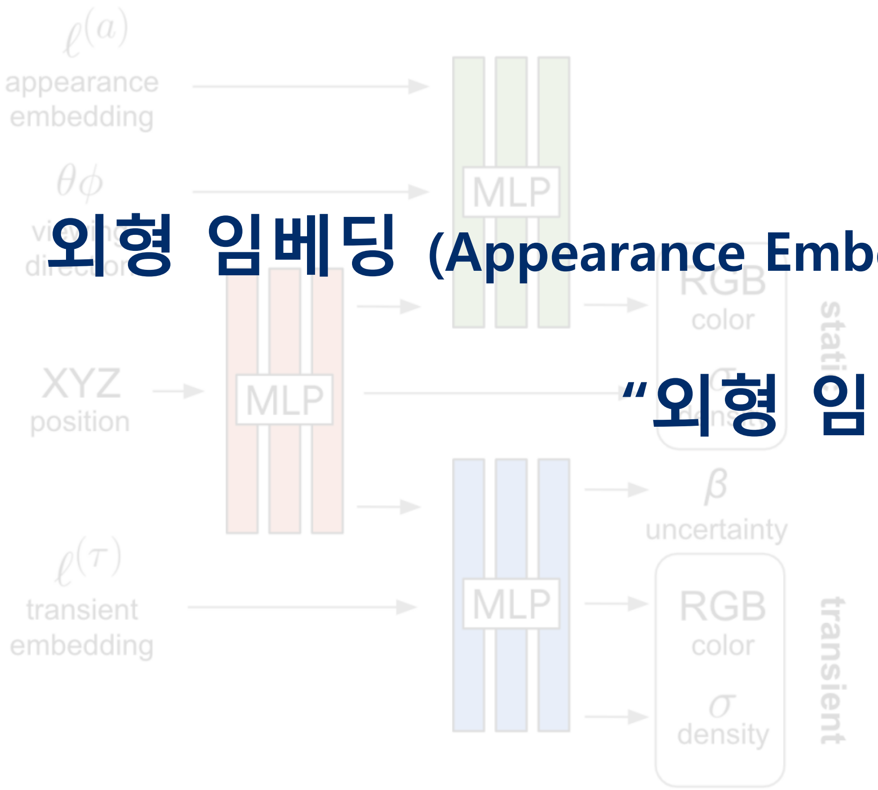
Related Work

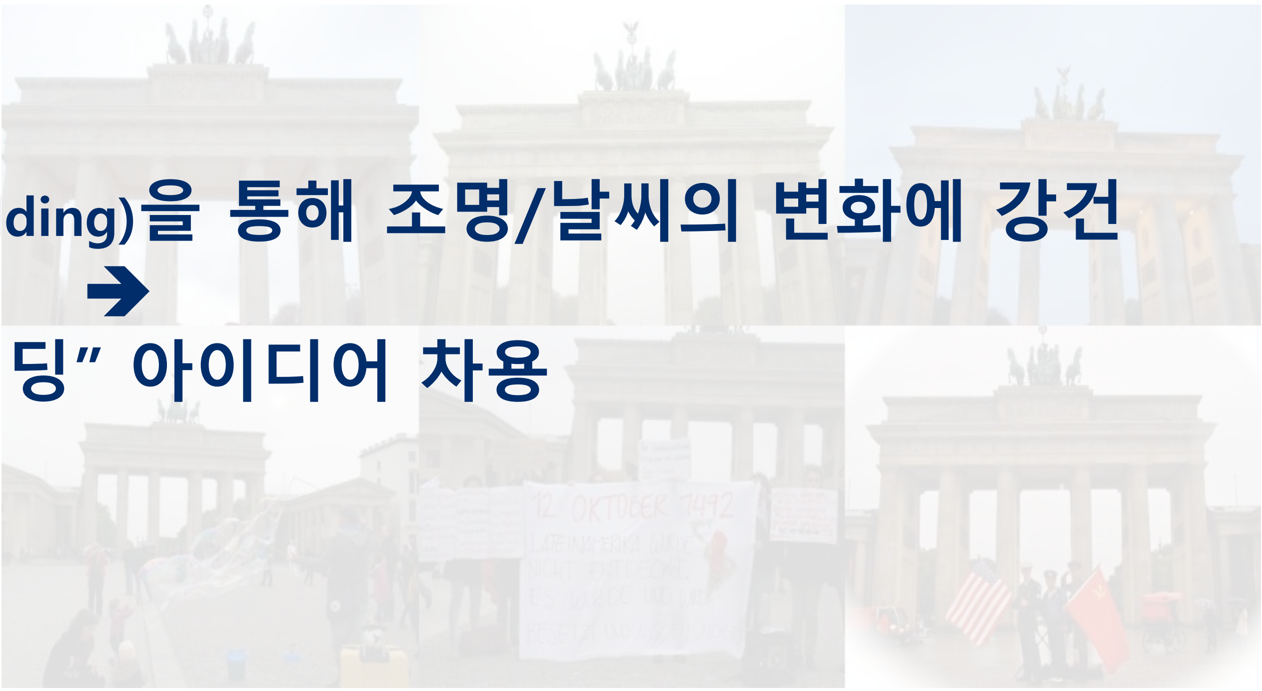
NeRF in the Wild



Related Work

NeRF in the Wild





외형 임베딩 (Appearance Embedding)을 통해 조명/날씨의 변화에 강건

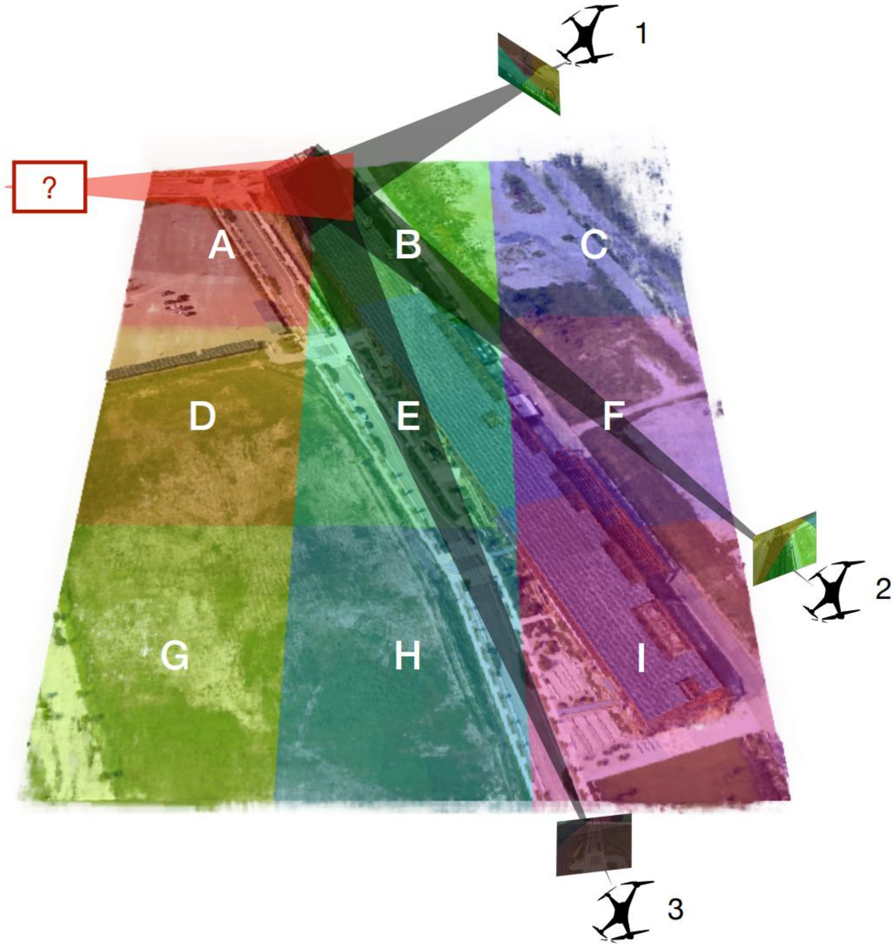
➔

“외형 임베딩” 아이디어 차용

Method

Method - architecture

- 2D grid decomposition - efficiency
- Each cell has their own f^n (NeRF == submodule)
- Each cell's centroids $\mathbf{n}_{\in \mathcal{N}} = (n_x, n_y, n_z)$
- **Input**: position, direction, appearance embedding vector $l^{(a)}$
- **Output**: opacity, color



Method - architecture

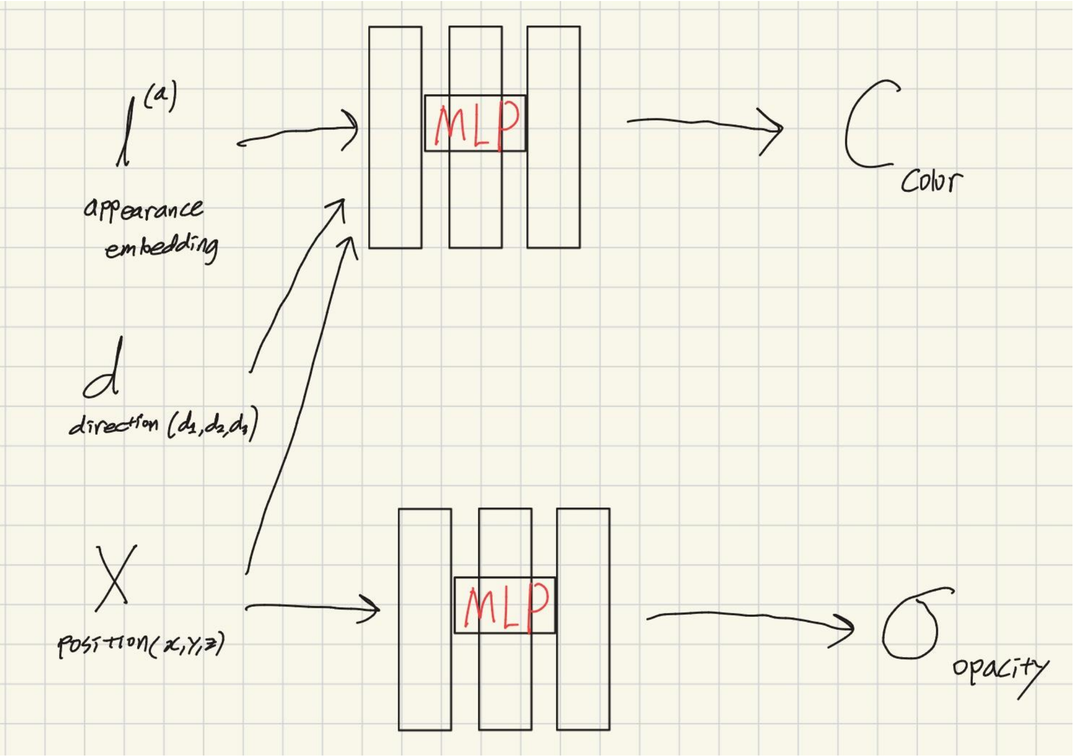
$$f^{\mathbf{n}}(\mathbf{x}) = \sigma \quad (1)$$

$$f^{\mathbf{n}}(\mathbf{x}, \mathbf{d}, l^{(a)}) = \mathbf{c} \quad (2)$$

$$\text{where } \mathbf{n} = \underset{n \in \mathcal{N}}{\operatorname{argmin}} \left\| n - \mathbf{x} \right\|^2 \quad (3)$$

어떤 서브모듈을 사용할지 결정

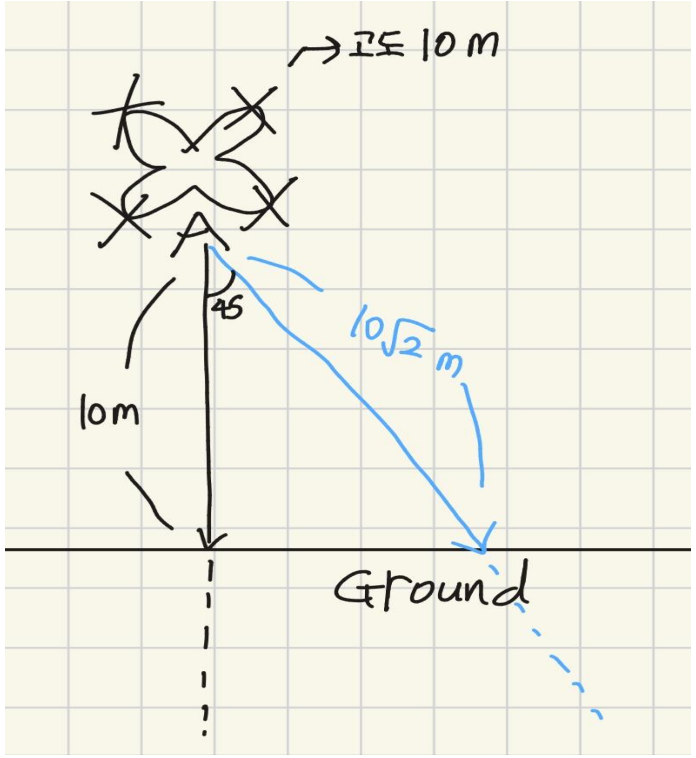
$$\mathbf{n}_{\in \mathcal{N}} = (n_x, n_y, n_z)$$



Mega-NeRF Architecture

Method - architecture

- Terminate rays near ground level.
- Subdivide the scene into a foreground and a background



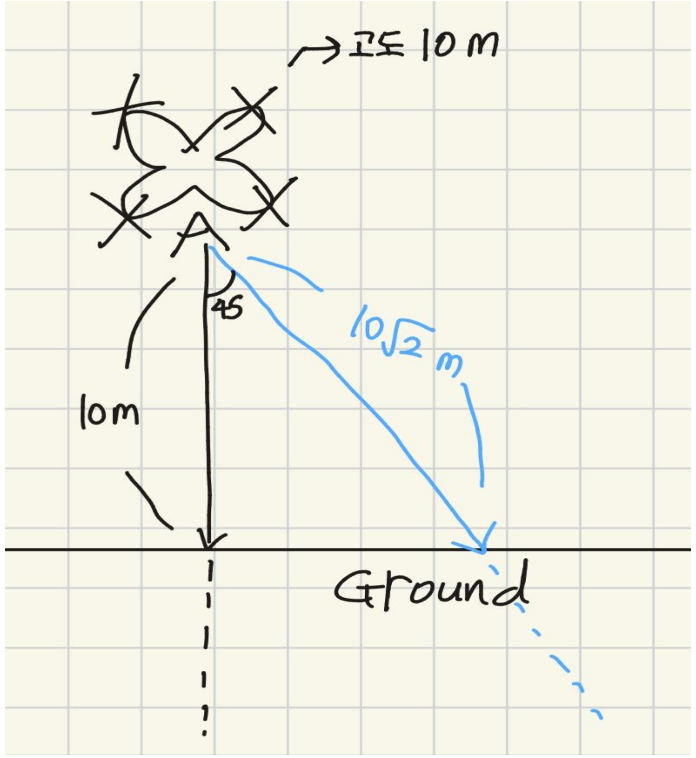
구

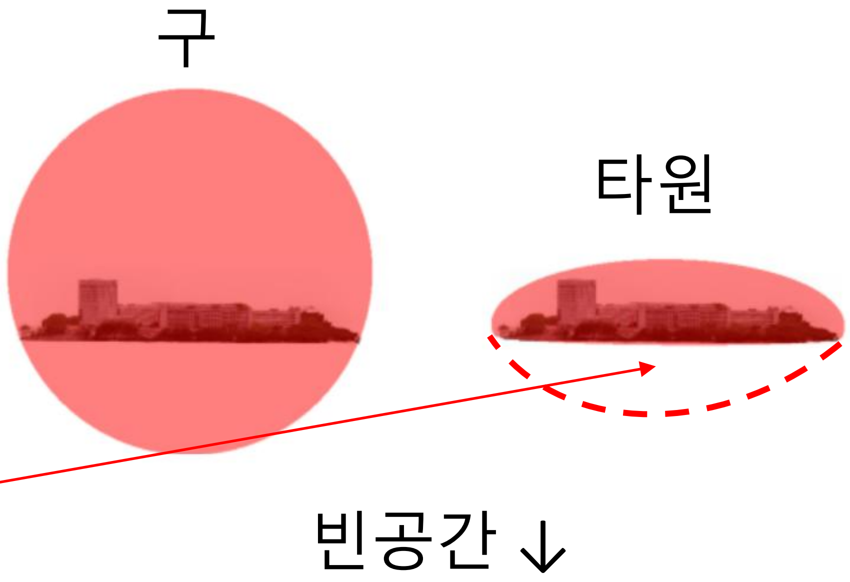
타원

빈공간 ↓

Method - architecture

- Terminate rays near ground level.
- Subdivide the scene into a foreground and a background



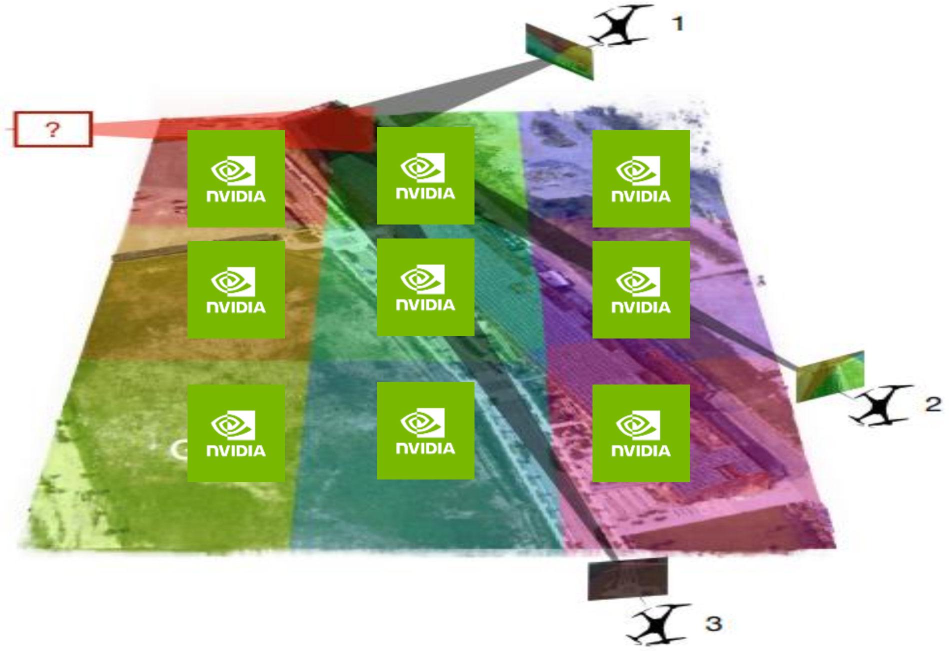


Method - Training

- Train each in parallel with no inter-module communication
- Add pixels to the trainset for only those spatial cells it intersects
- small overlap factor between cells (15% in our experiments)
- Spatial Data Pruning

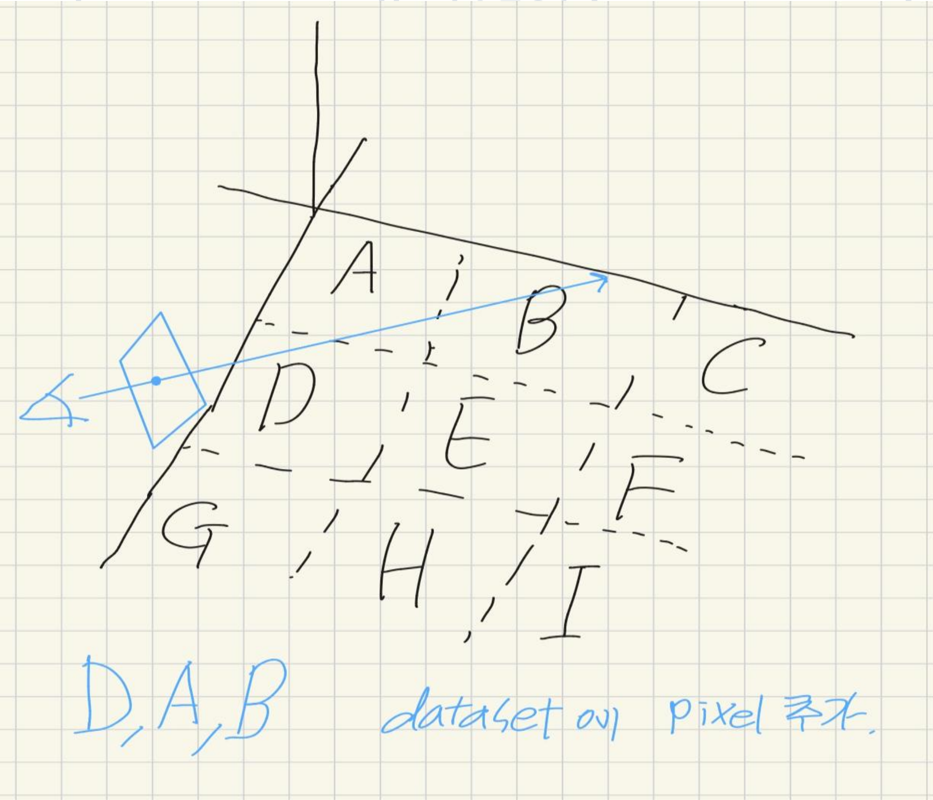
Method - Training

- Train each in parallel with no inter-module communication
- Add pixels to the trainset for only those spatial cells it intersects
- small overlap factor between cells (15% in our experiments)
- Spatial Data Pruning



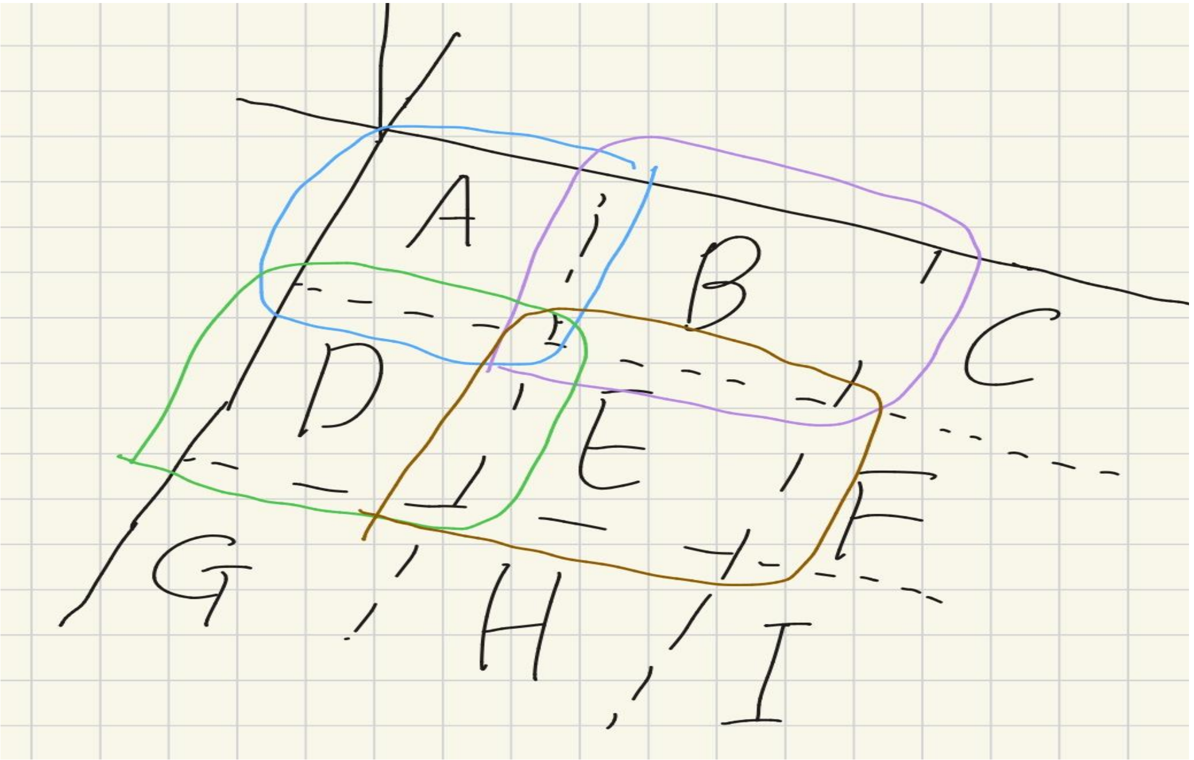
Method - Training

- Train each in parallel with no inter-module communication
- Add pixels to the trainset for only those spatial cells it intersects
- small overlap fact ments)
- Spatial Data Pruni



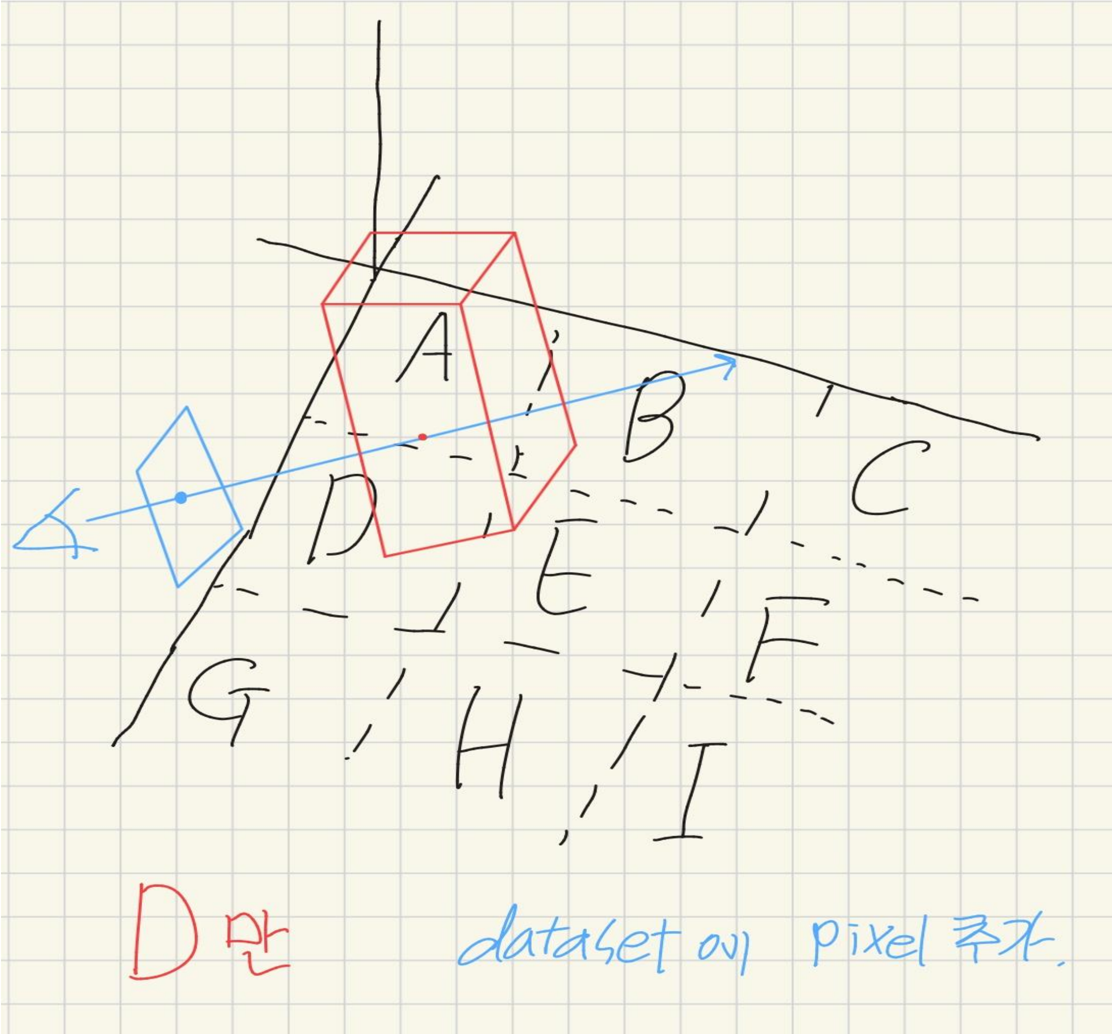
Method - Training

- Train each in parallel with no inter-module communication
- Add pixels to the trainset for only those spatial cells it intersects
- small overlap factor between cells (15% in our experiments)
- Spatial Data Pruning



Method - Training

- Train each in parallel with
- Add pixels to the trainset
- small overlap factor betw
- Spatial Data Pruning

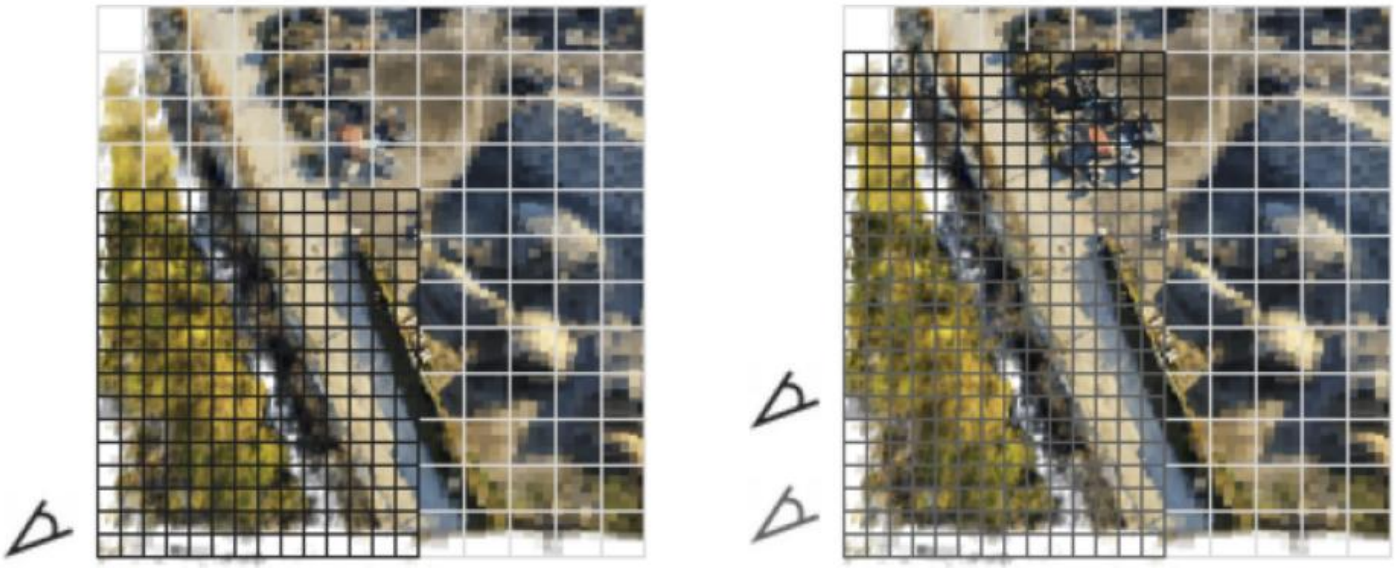


Method - Rendering

- Dynamic Octree
- Temporal coherence of interactive flythroughs
- Caching – LRU(Least Recently Used)
- Guided sampling

Method - Rendering

- Dynamic Octree
- Temporal coherence of interactive flythroughs
- Caching – LRU(Least Recently Used)
- Guided sampling

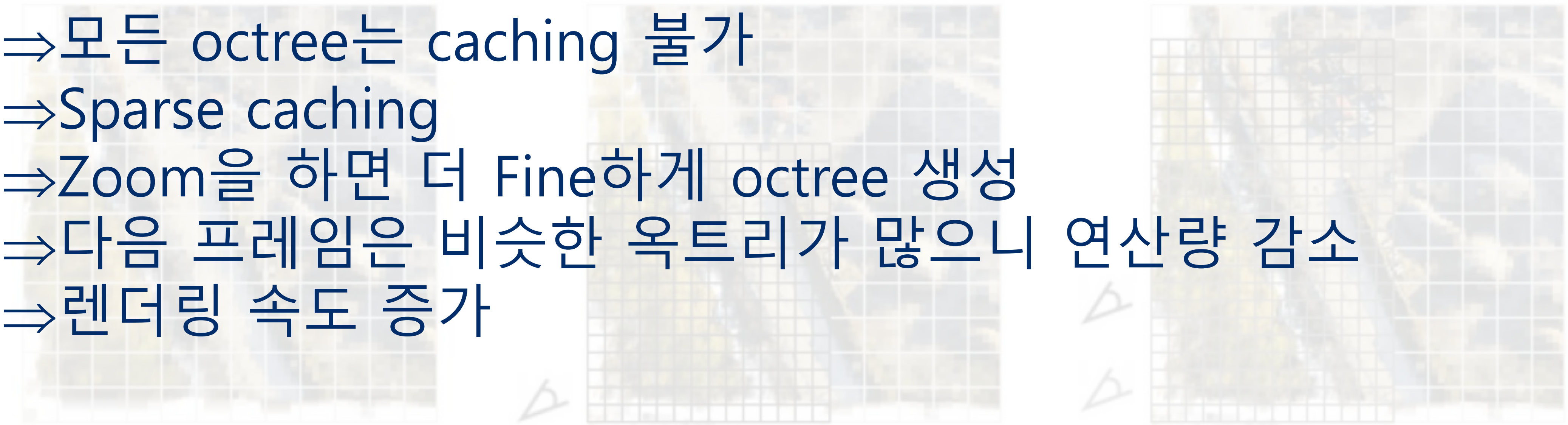


Sparse caching View Next View

Method - Rendering

- Dynamic Octree
- Temporal coherence of interactive flythroughs
- Caching – LRU(Least Recently Used)

Urban Scale environment
⇒모든 octree는 caching 불가
⇒Sparse caching
⇒Zoom을 하면 더 Fine하게 octree 생성
⇒다음 프레임은 비슷한 옥트리가 많으니 연산량 감소
⇒렌더링 속도 증가



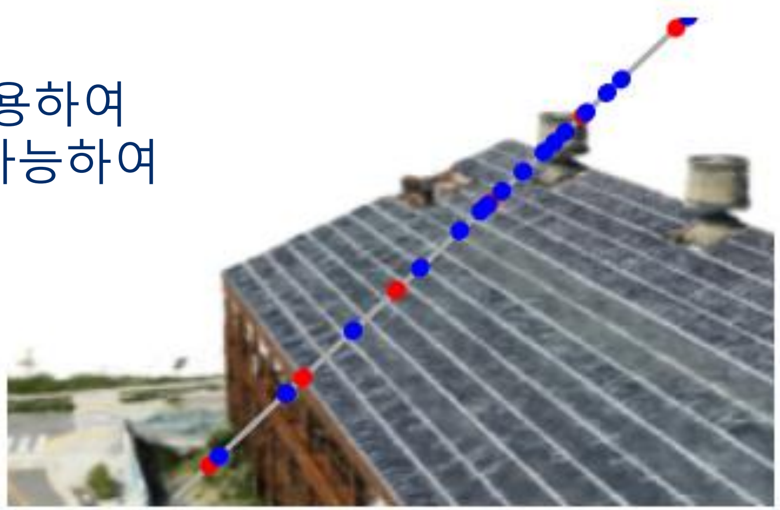
Sparse caching View Next View

Method - Rendering

*Octree: 3D 공간을 8개의 작은 정육면체로 재귀적으로 계속 쪼개서 데이터를 관리하는 트리(Tree) 자료구조 (RGB와 Opacity 값을 보관)

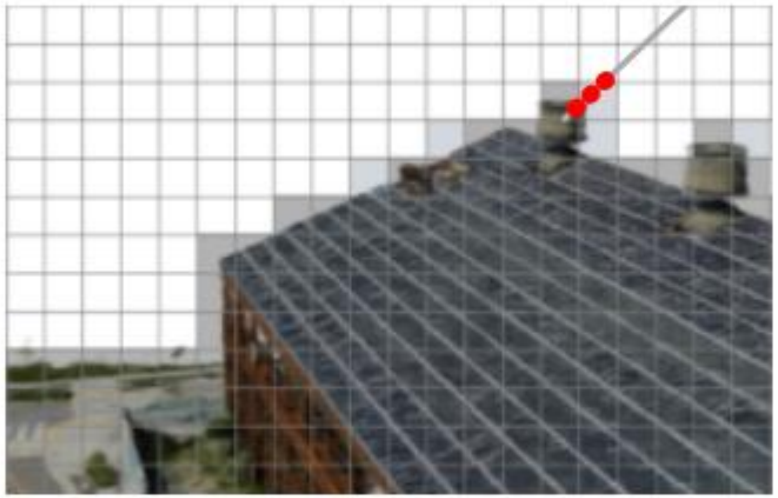
- Temporal coherence of interactive flythroughs
- Caching – LRU(Least Recently Used)
- Guided sampling

Octree의 opacity 값을 활용하여 octree가 비었는지 확인가능하여 샘플링을 줄일 수 있다.



Standard Hierachical Sampling

NeRF



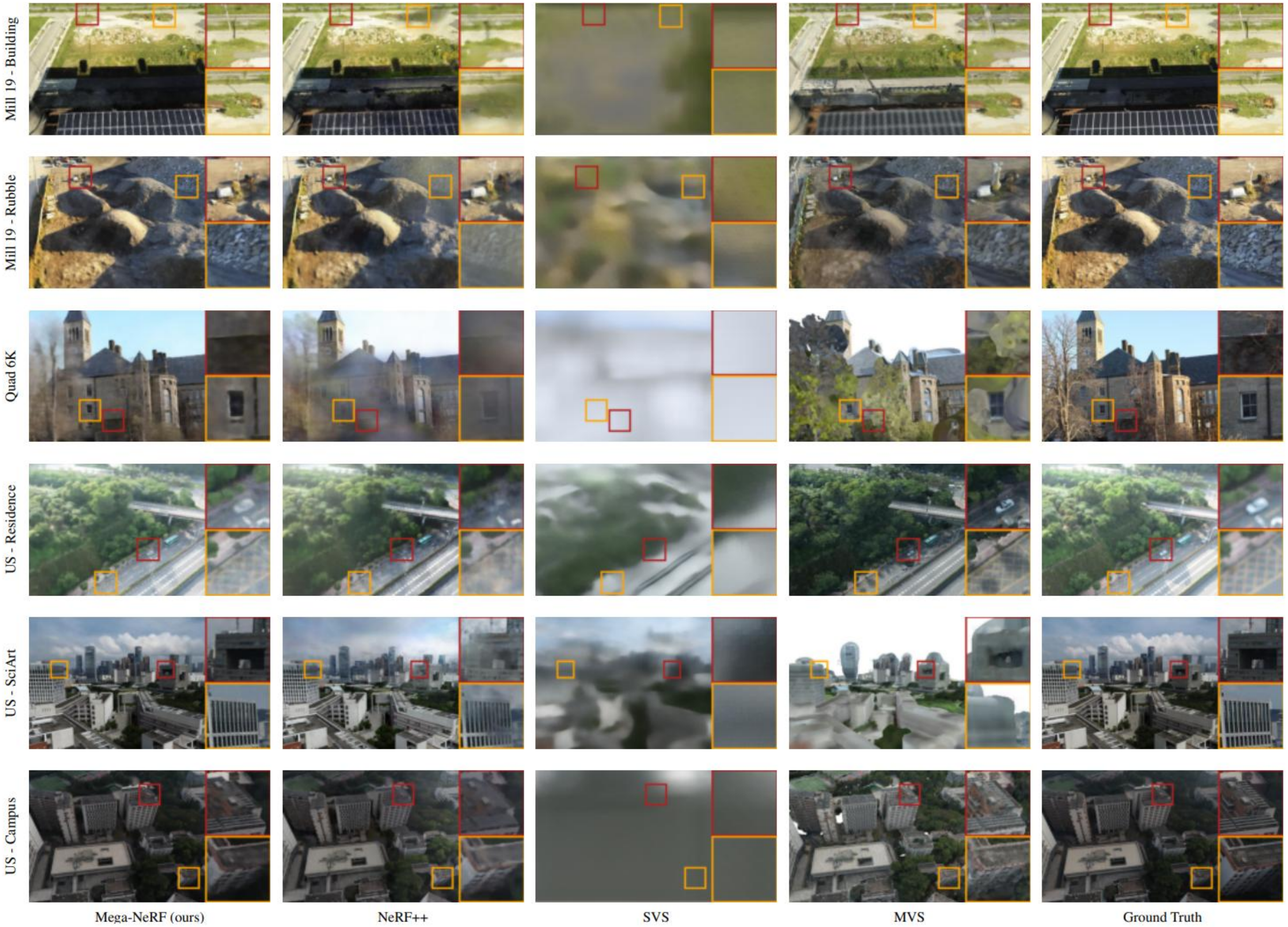
Guided Sampling

Mega-NeRF

Experiment

Experiment

Training



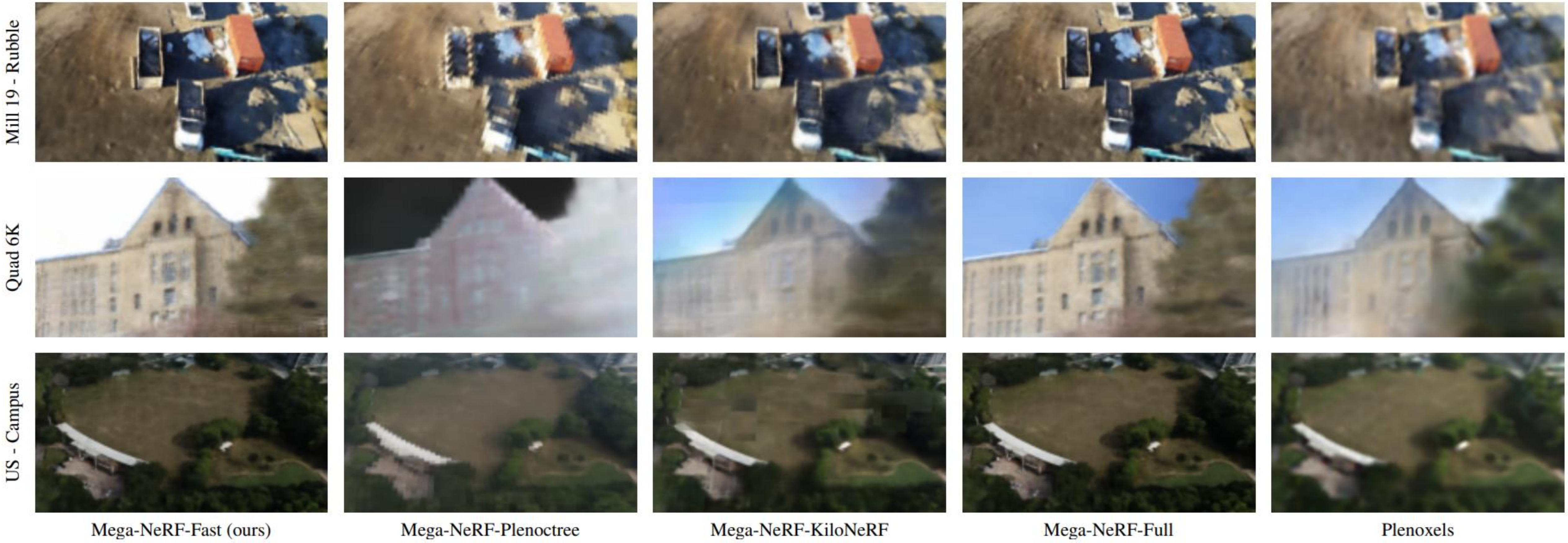
Experiment

Training

	Mill 19 - Building				Mill 19 - Rubble				Quad 6k			
	↑PSNR	↑SSIM	↓LPIPS	↓Time (h)	↑PSNR	↑SSIM	↓LPIPS	↓Time(h)	↑PSNR	↑SSIM	↓LPIPS	↓Time(h)
NeRF	19.54	0.525	0.512	59:51	21.14	0.522	0.546	60:21	16.75	0.559	0.616	62:48
NeRF++	19.48	0.520	0.514	89:02	20.90	0.519	0.548	90:42	16.73	0.560	0.611	90:34
SVS	12.59	0.299	0.778	38:17	13.97	0.323	0.788	37:33	11.45	0.504	0.637	29:48
DeepView	13.28	0.295	0.751	31:20	14.47	0.310	0.734	32:11	11.34	0.471	0.708	19:51
MVS	16.45	0.451	0.545	32:29	18.59	0.478	0.532	31:42	11.81	0.425	**0.594**	**18:55**
Mega-NeRF	**20.93**	**0.547**	**0.504**	**29:49**	**24.06**	**0.553**	**0.516**	**30:48**	**18.13**	**0.568**	0.602	39:43
	UrbanScene3D - Residence				UrbanScene3D - Sci-Art				UrbanScene3D - Campus			
	↑PSNR	↑SSIM	↓LPIPS	↓Time (h)	↑PSNR	↑SSIM	↓LPIPS	↓Time(h)	↑PSNR	↑SSIM	↓LPIPS	↓Time(h)
NeRF	19.01	0.593	0.488	62:40	20.70	0.727	0.418	60:15	21.83	0.521	0.630	61:56
NeRF++	18.99	0.586	0.493	90:48	20.83	0.755	0.393	95:00	21.81	0.520	0.630	93:50
SVS	16.55	0.388	0.704	77:15	15.05	0.493	0.716	59:58	13.45	0.356	0.773	105:01
DeepView	13.07	0.313	0.767	30:30	12.22	0.454	0.831	31:29	13.77	0.351	0.764	33:08
MVS	17.18	0.532	**0.429**	69:07	14.38	0.499	0.672	73:24	16.51	0.382	**0.581**	96:01
Mega-NeRF	**22.08**	**0.628**	0.489	**27:20**	**25.60**	**0.770**	**0.390**	**27:39**	**23.42**	**0.537**	0.618	**29:03**

Experiment

Rendering



Experiment

Rendering

best <u>second-best</u>	Mill 19					Quad 6k					UrbanScene3D				
	↑PSNR	↑SSIM	↓LPIPS	Preprocess Time (h)	Render Time (s)	↑PSNR	↑SSIM	↓LPIPS	Preprocess Time (h)	Render Time (s)	↑PSNR	↑SSIM	↓LPIPS	Preprocess Time (h)	Render Time (s)
Mega-NeRF-Plenoctree	16.27	0.430	0.621	<u>1:26</u>	**0.031**	13.88	0.589	0.427	<u>1:33</u>	**0.010**	16.41	0.498	0.530	**1:07**	**0.025**
Mega-NeRF-KiloNeRF	21.85	0.521	0.512	30:03	0.784	20.61	0.652	0.356	27:33	1.021	21.11	0.542	0.453	34:00	0.824
Mega-NeRF-Full	**22.96**	**0.588**	**0.452**	-	101	**21.52**	**0.676**	<u>0.355</u>	-	174	**24.92**	**0.710**	**0.393**	-	122
Plenoxels	19.32	0.476	0.592	-	0.482	18.61	0.645	0.411	-	<u>0.194</u>	20.06	0.608	0.503	-	0.531
Mega-NeRF-Initial	17.41	0.447	0.570	**1:08**	<u>0.235</u>	14.30	0.585	0.386	**1:31**	0.214	17.22	0.527	0.506	<u>1:10</u>	<u>0.221</u>
Mega-NeRF-Dynamic	<u>22.34</u>	<u>0.573</u>	<u>0.464</u>	**1:08**	3.96	<u>20.84</u>	<u>0.658</u>	**0.342**	**1:31**	2.91	<u>23.99</u>	<u>0.691</u>	<u>0.408</u>	<u>1:10</u>	3.219

Experiment

Ablation Study

	Mill 19			Quad 6k			UrbanScene3D		
	↑PSNR	↑SSIM	↓LPIPS	↑PSNR	↑SSIM	↓LPIPS	↑PSNR	↑SSIM	↓LPIPS
Mega-NeRF-no-embed	20.42	0.500	0.561	16.16	0.544	0.643	19.45	0.587	0.545
Mega-NeRF-embed-only	21.48	0.494	0.566	17.91	0.559	0.638	22.79	0.611	0.537
Mega-NeRF-no-bounds	22.14	0.534	0.522	18.02	0.565	0.616	23.42	0.636	0.511
Mega-NeRF-dense	21.63	0.504	0.551	17.94	0.562	0.627	22.44	0.605	0.558
Mega-NeRF-joint	21.10	0.490	0.574	17.43	0.560	0.616	21.45	0.595	0.567
Mega-NeRF	**22.34**	**0.540**	**0.518**	**18.08**	**0.566**	**0.602**	**23.60**	**0.641**	**0.504**

Contribution

Contribution

- Presents a novel rendering method that exploits temporal coherence.
- Presents a new large-scale dataset.
- Divide the model into multiple submodules and train in a fully parallelizable manner.