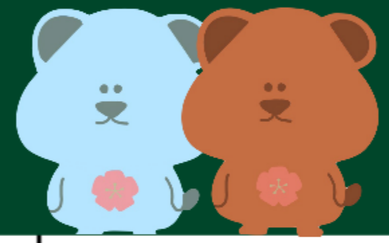
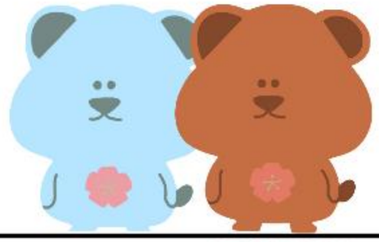


Instant Neural Graphics Primitives with a Multiresolution Hash Encoding

고급 심화 세션 민소연

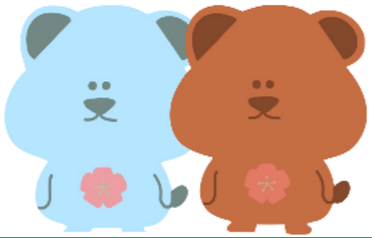
EWHA EURON

Table of Contents



EWHA EURON

Intro



EWHA EURON

1. 논문 선정 이유

IT·과학 한경 Geeks

게임·디자인도 '3D 콘텐츠' 시대...韓 벤처, 초기부터 글로벌 정조준

허란 기자

입력 2023.01.18 17:45 수정 2023.01.19 17:28 지면 A14 가가

3D 시장 노리는 스타트업들

클로버추얼, 3D 의상 디자인 세계 1위
트라이폴리곤, 게임엔진社 유니티 제휴
엔닷라이트, 해외 빅테크들 '러브콜'

"그래픽 기준은 이미 2D→3D로 넘어가"
3D 애니메이션 2030년 63조원 전망
메타버스 플랫폼선 3D 콘텐츠가 '자산'

https://www.hankyung.com/article/2023011820261

오늘의 주요뉴스

'공매도 전격 금지' 초
어지는 韓 자본시장

단독 최수연 네이버
라별 특화된 AI 내놓

단독 "거액 날릴 판"...
과된 고용·산재보험료

"위험 더 커져 비상경
이 두려운 기업들 '피

"한국에선 일당 20만
이삿짐센터 접수한

기술동향

인공지능을 이용한 3D 콘텐츠 기술 동향 및 향후 전망

등록일 2019-08-26 · 조회수 4006 · 분류 기술동향 > 화이트바이오 > 바이오화학·에너지기술

- 자료발간일 | 2019-08-01
- 출처 | 한국전자통신연구원
- 원문링크 | https://ettrends.etri.re.kr/ettrends/178/0905178002/
- 키워드 | #3D #인공지능 #3D 스캔 데이터 #딥러닝
- 첨부파일 | 34-4_15-22.pdf (다운로드 141회) 다운로드 바로보기

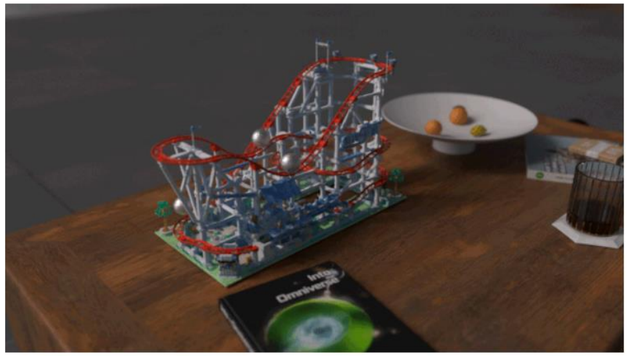
인공지능을 이용한 3D 콘텐츠 기술 동향 및 향후 전망

https://www.bioin.or.kr/board.do?num=289726&cmd=view&bid=tech&cPage=96&cate1=all&cate2=all2&s_key=title&s_str=&sdate=2018-06-01&edate=

메타버스 포커스 인물 오피니언 이벤트 Local

엔비디아, AI로 3D 메시 생성하는 '플렉시큐브' 출시

박찬 기자 입력 2023.08.14 18:50 댓글 0 좋아요 0



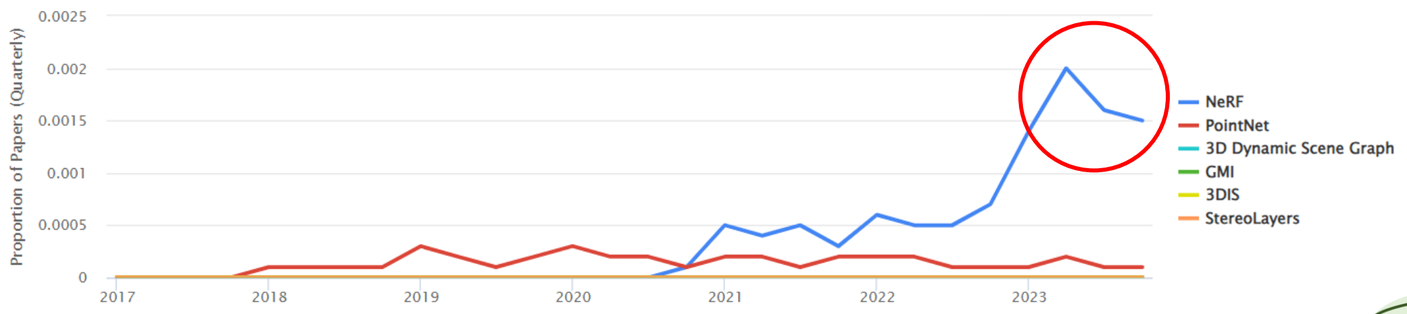
(사진=엔비디아)

엔비디아가 인공지능(AI)을 사용해 다양한 3D 애플리케이션을 위한 표준 삼각형 '메시(mesh)'를 생성하는

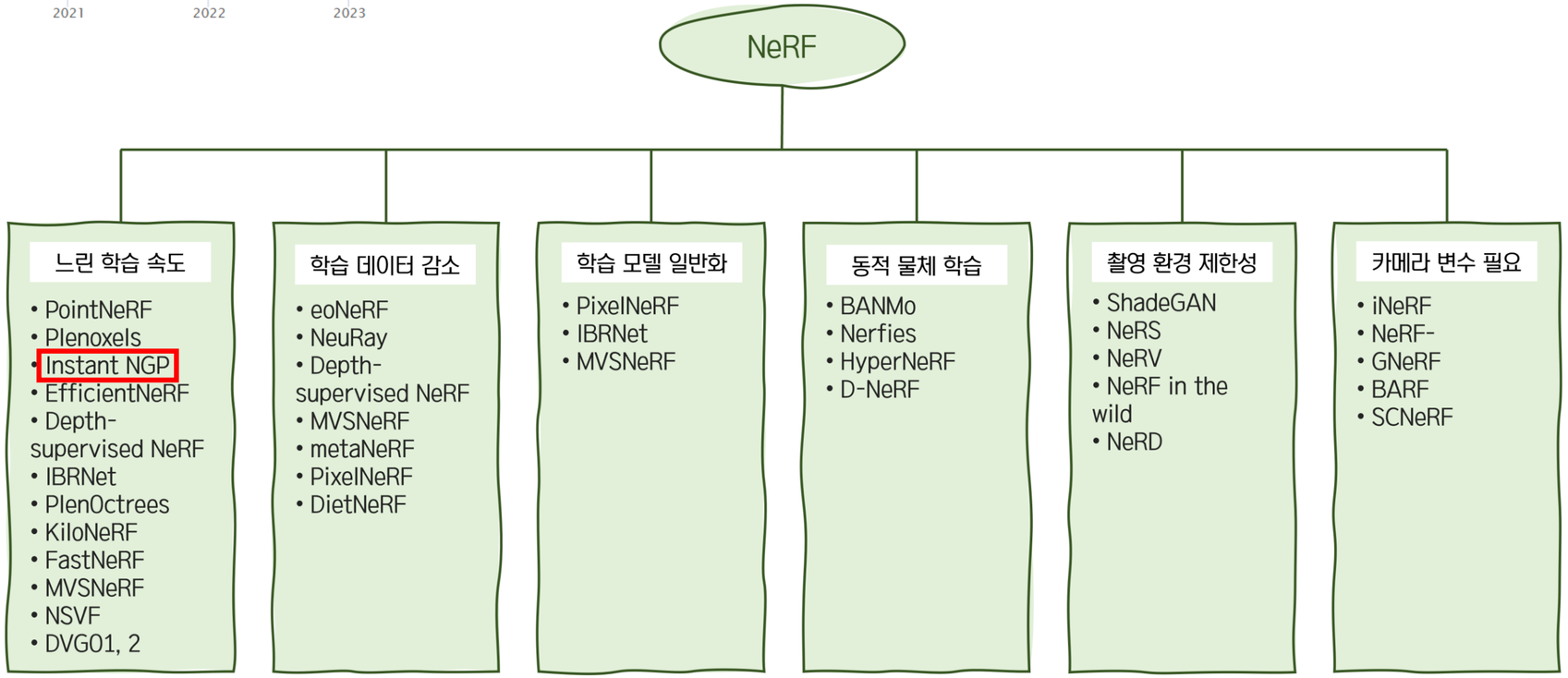
https://www.aitimes.com/news/articleView.html?idxno=152883

1. 전망

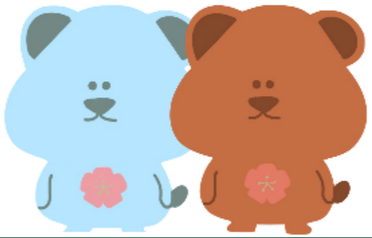
Usage Over Time



This feature is experimental; we are continuously improving our matching algorithm.



NeRF (Neural Radiance Field)



EWHA EURON

2. NeRF

ECCV 2020 Oral – Best Paper Honorable Mention

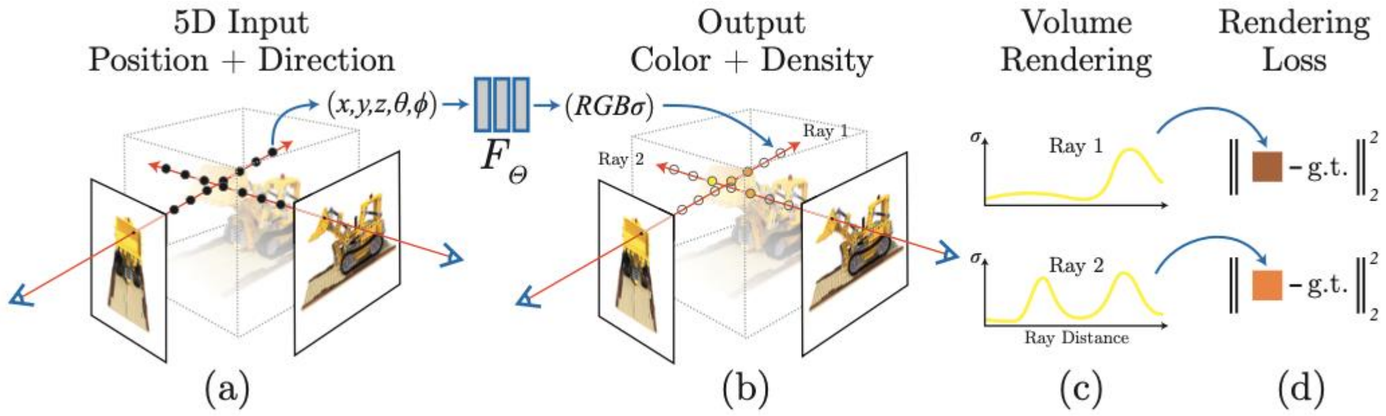
NeRF 논문 저자의 Github link

Keras에서 제공하는 NeRF Tutorial

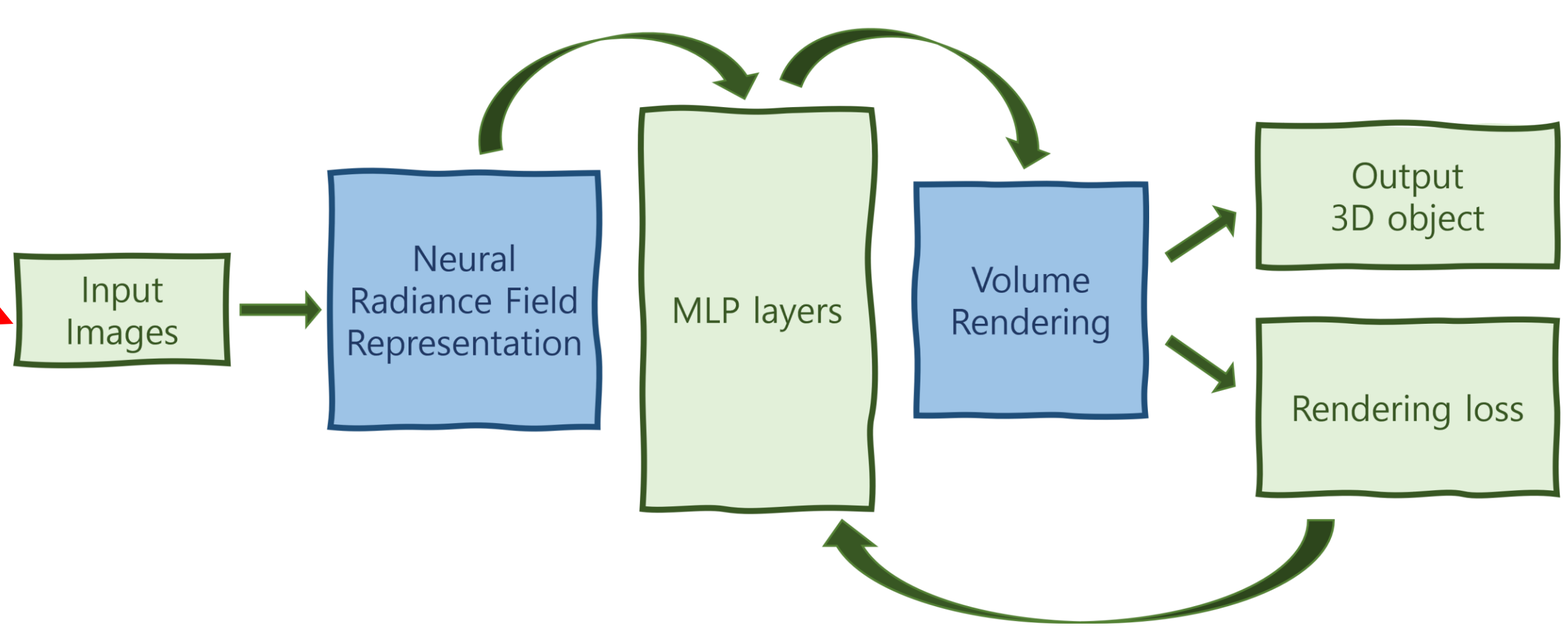
2. NeRF

Input Images를 이용해서 Neural Network로 구축한
Radiance Field를 최적화하여 3D object로 modeling하는 기술

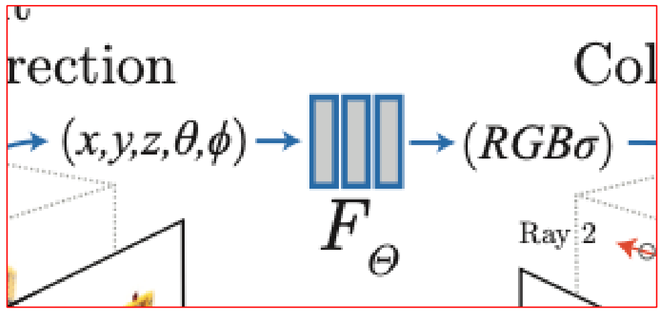
2. NeRF

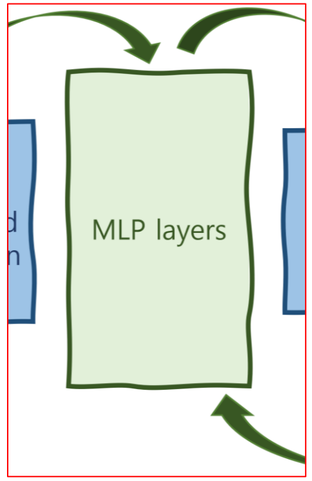


전체 알고리즘 구조



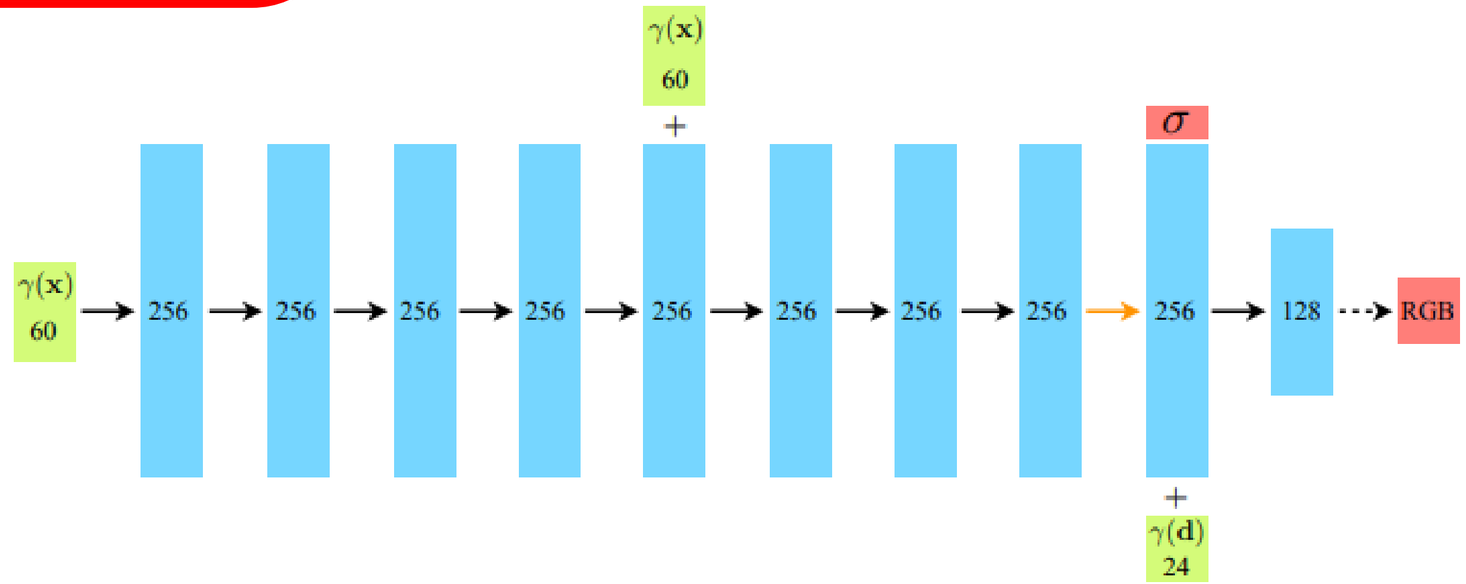
2. NeRF - MLP 구조



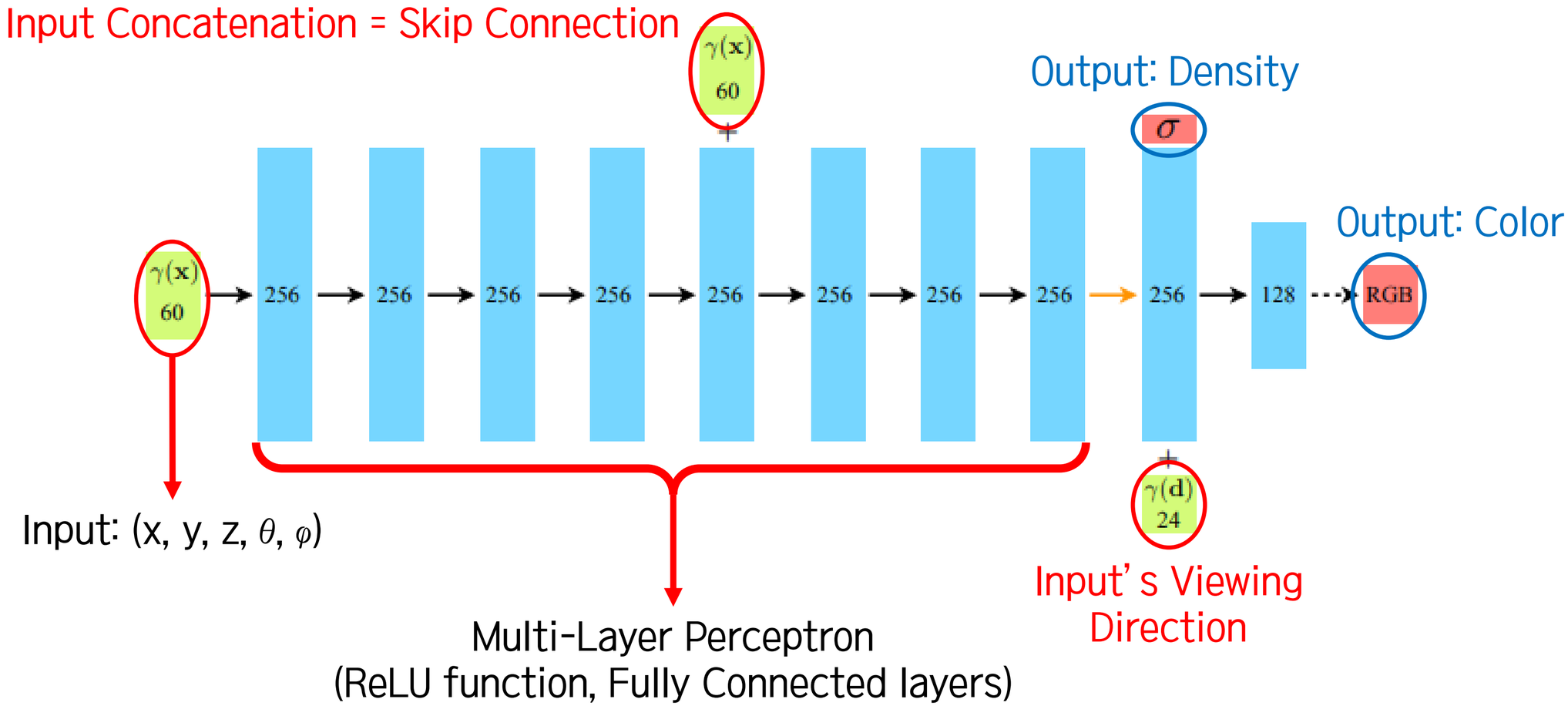


$$F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$$

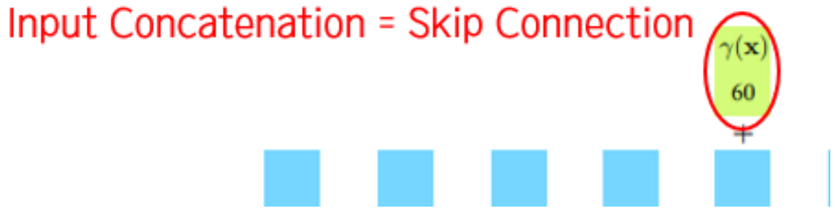
Input: voxel, viewing direction → Output: color, density



2. NeRF - MLP 구조


Input Concatenation = Skip Connection
γ(x)
60
Output: Density
σ
Output: Color
γ(x)
60
256
256
256
256
256
256
256
256
256
128
RGB
Input: (x, y, z, θ, φ)
γ(d)
24
Input's Viewing Direction
Multi-Layer Perceptron
(ReLU function, Fully Connected layers)

2. NeRF - MLP 구조



너무 layer가 깊은 네트워크에서 gradient vanishing 문제가 생기는 것을 방지하기 위해 n개 층을 건너뛰고 input을 다시 넣어주는 기법

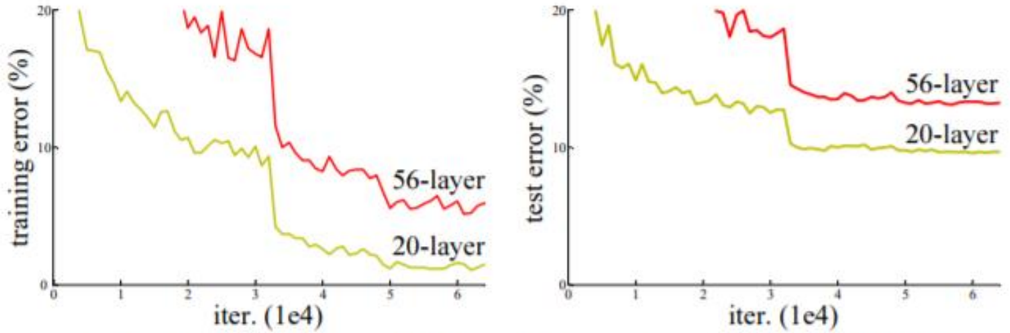


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

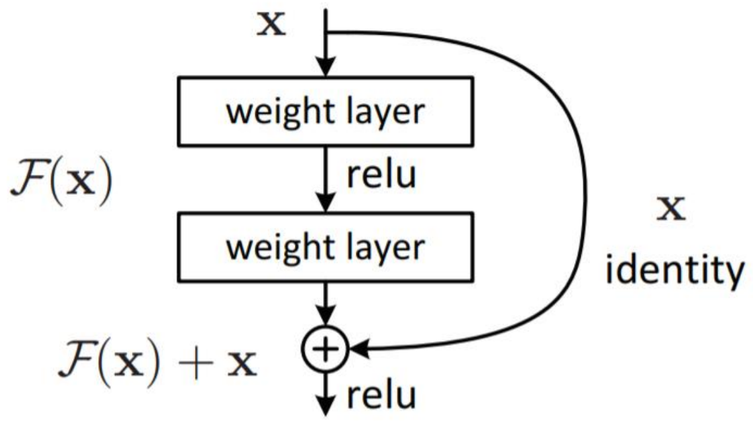
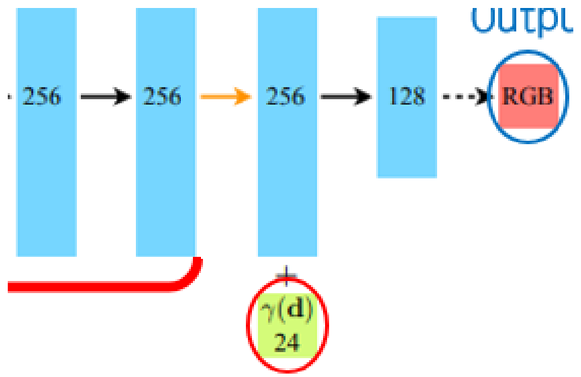
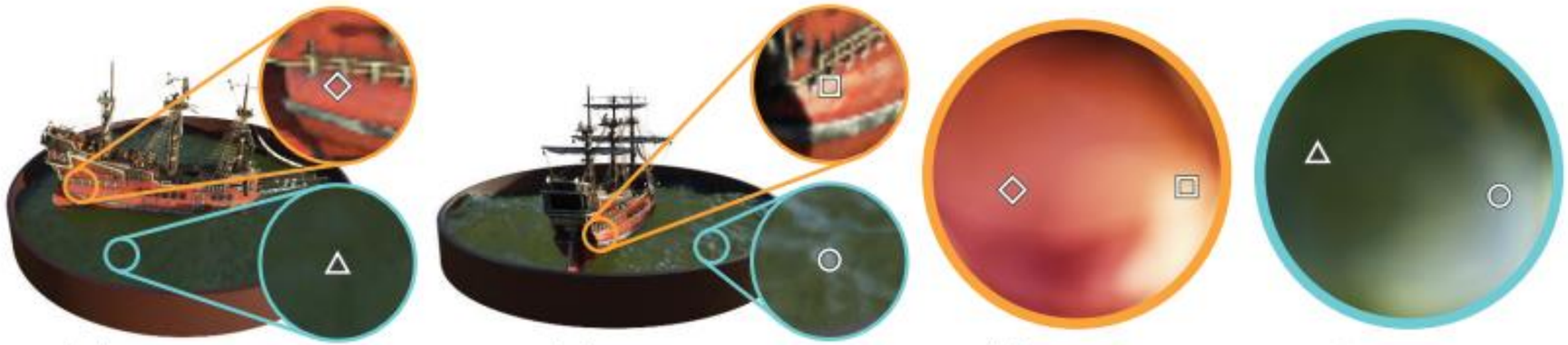


Figure 2. Residual learning: a building block.

2. NeRF - MLP 구조

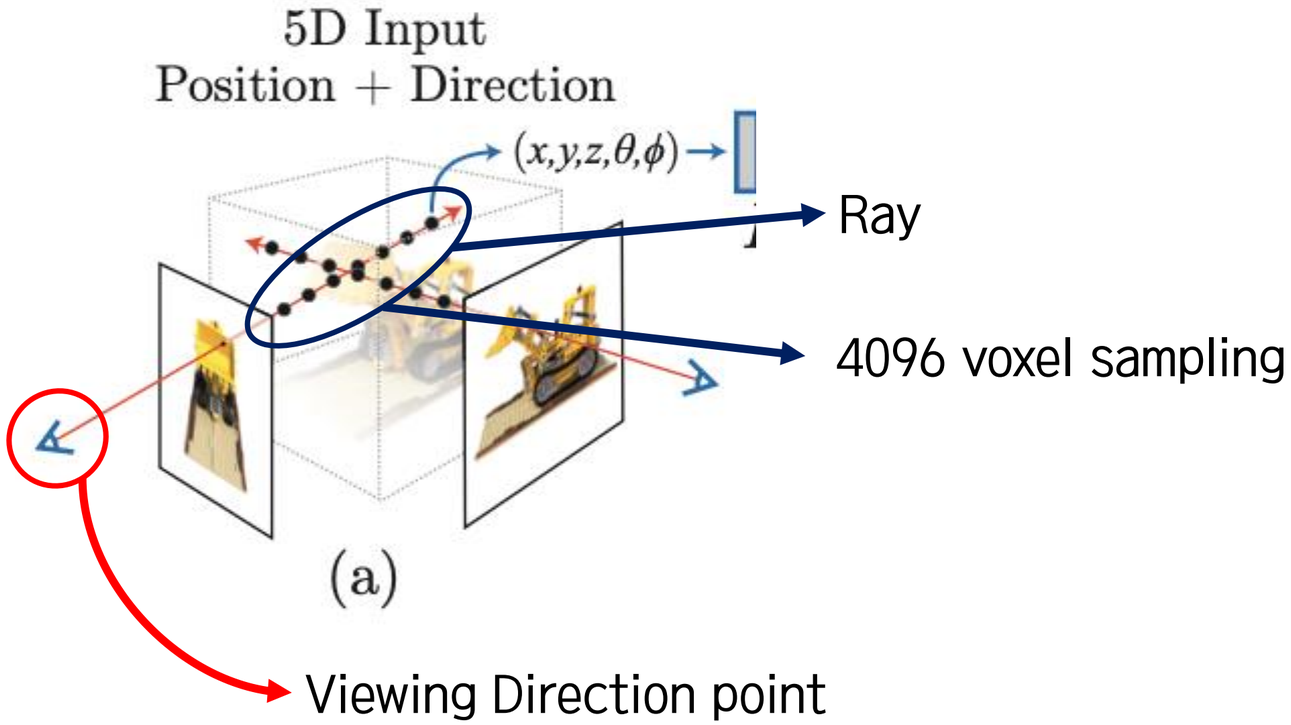


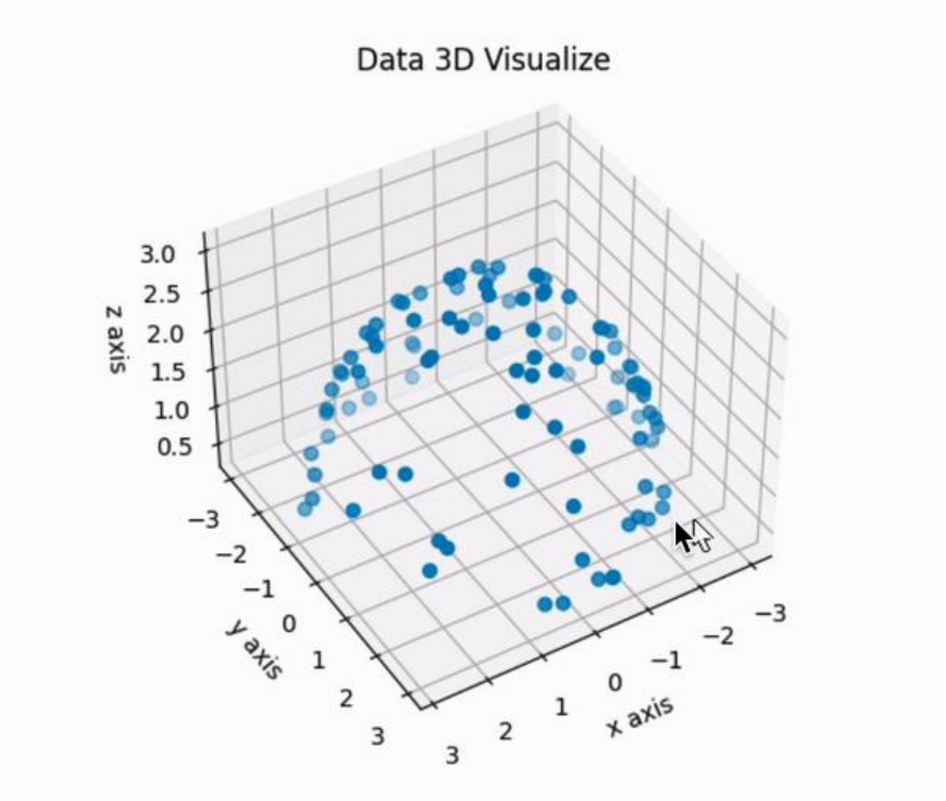
같은 (x, y, z) voxel에 있어도 다른 viewing direction에서 보면
색상과 밀도 값이 다를 수 있기 때문에 RGB 추적 과정에 이를 추가해 줌.



(a) View 1 (b) View 2 (c) Radiance Distributions

2. NeRF – Volume Rendering

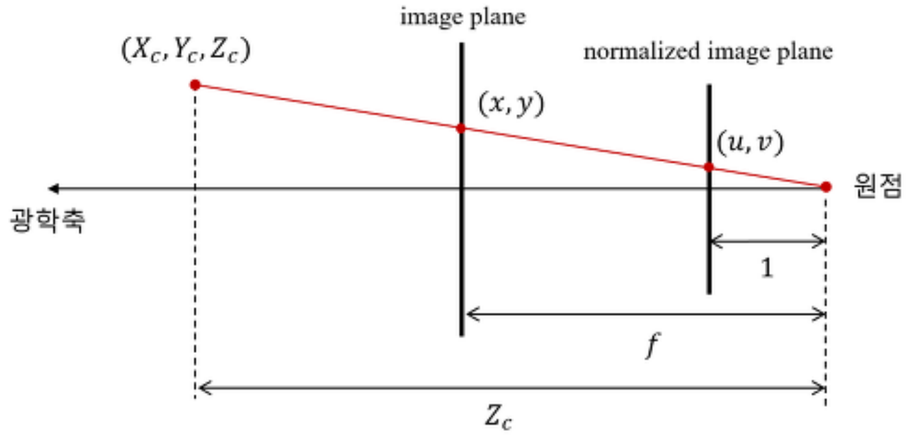




Colmap library를 사용해서 Voxel과 Camera View point로 Input Images에서
MLP 구조의 input으로 넣기 위한 정보를 만든다.

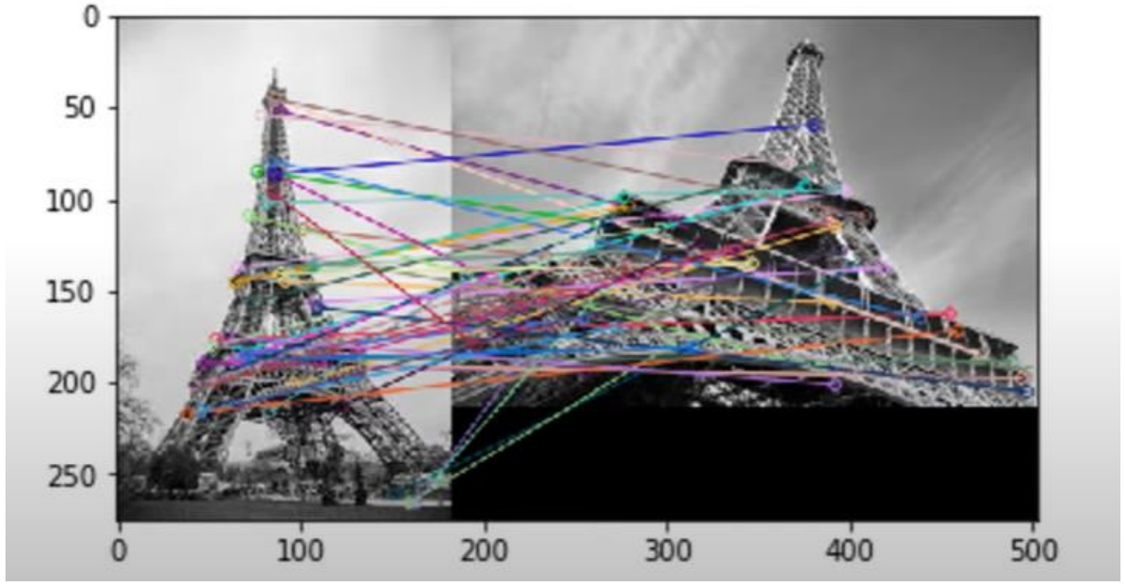
2. NeRF – Volume Rendering

* Colmap이란?



$$s\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & \text{skew\_c}f_x & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= A[R \mid t] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$



https://velog.io/@seosan/Colmap%EC%9D%B4%EB%9E%80

https://github.com/colmap/colmap

2. NeRF – Volume Rendering

$$F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$$

Input: voxel, viewing direction

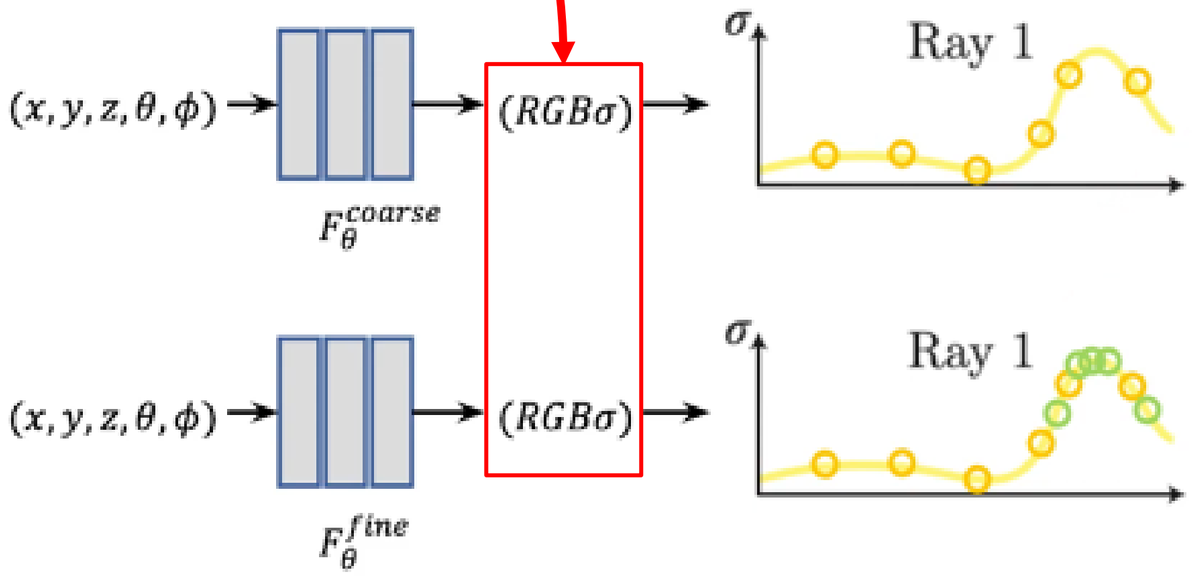
Output: color, density

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt\,, \text{ where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$$

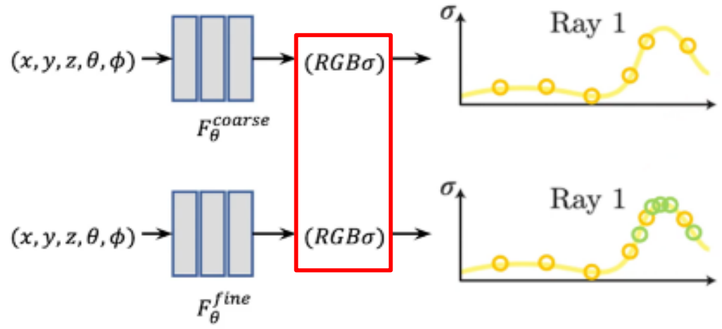
$$\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$$

$$t_i \sim \mathcal{U}\left[t_n + \frac{i-1}{N}(t_f - t_n),\ t_n + \frac{i}{N}(t_f - t_n)\right]$$

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i\,, \text{ where } T_i = \exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right)$$



2. NeRF – Volume Rendering



물체가 있을법한 범위 (t-near ~ t-far)

Weight 역할: 투과도가 높으면 weigh는 낮다 =
물체가 있을 확률이 낮다

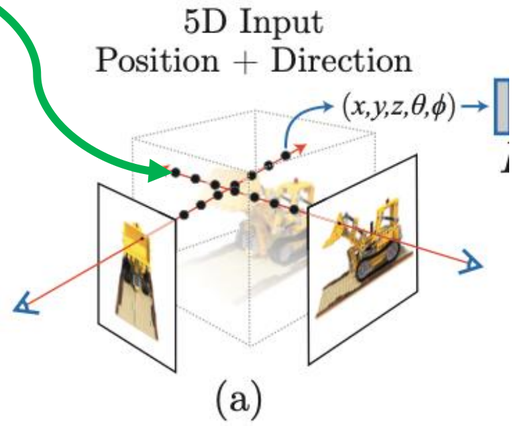
$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt\,, \quad \text{where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$$

Density RGB

Transmittance (투과도)

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

Ray Position



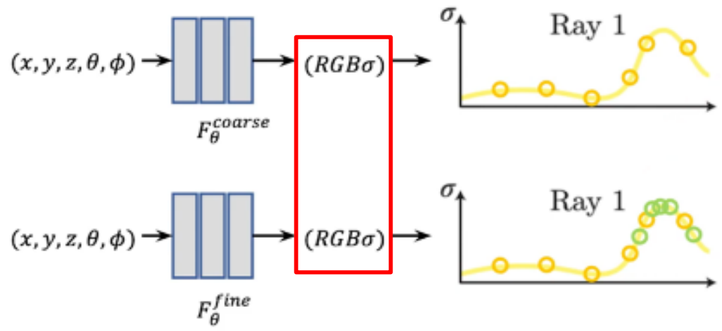
$$t_i \sim \mathcal{U}\left[t_n + \frac{i-1}{N}(t_f - t_n),\ t_n + \frac{i}{N}(t_f - t_n)\right]$$

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i\,, \quad \text{where } T_i = \exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right)$$

2. NeRF – Volume Rendering

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt\,, \text{ where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$$

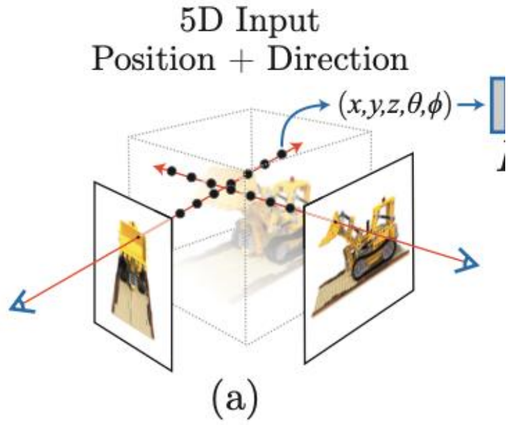
$$\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$$



$$t_i \sim \mathcal{U}\left[t_n + \frac{i-1}{N}(t_f - t_n),\ t_n + \frac{i}{N}(t_f - t_n)\right]$$

Random Sampling

* Ray 위에서 random한 point를 뽑아 정해진 discrete한 point를 뽑을 때보다 continuous한 model을 만들고자 함



$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i\,, \text{ where } T_i = \exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right)$$

Modeling (Coarse Model)

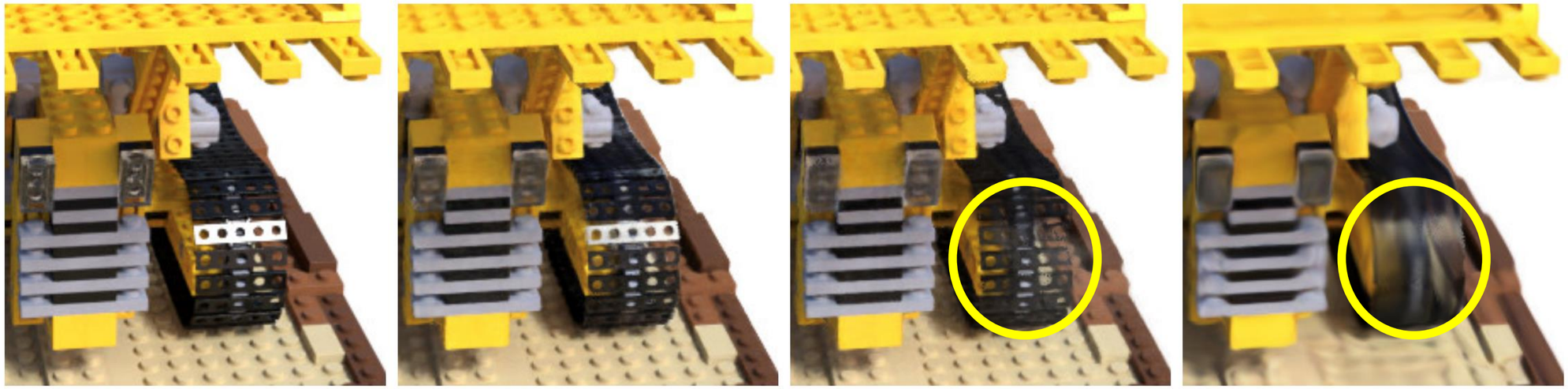
2. NeRF – Volume Rendering

* Positional Encoding (Data augmentation의 역할)

High Frequency 수준의 performance를 위해서 각각의
point (x, y, z) 정보를 늘려주는 역할

$$\gamma(p) = \left(\sin\left(2^0\pi p\right), \cos\left(2^0\pi p\right), \cdots, \sin\left(2^{L-1}\pi p\right), \cos\left(2^{L-1}\pi p\right)\right)$$

* p: (x, y, z)나 d 중 하나
Ex) r(x) -> L=10 -> (x, y, z)에 대해서 60개의 데이터 추가 생성
r(d) -> L=4 -> d에 대해서 24개 데이터 추가 생성



Ground Truth | Complete Model | No View Dependence | No Positional Encoding

2. NeRF – Volume Rendering

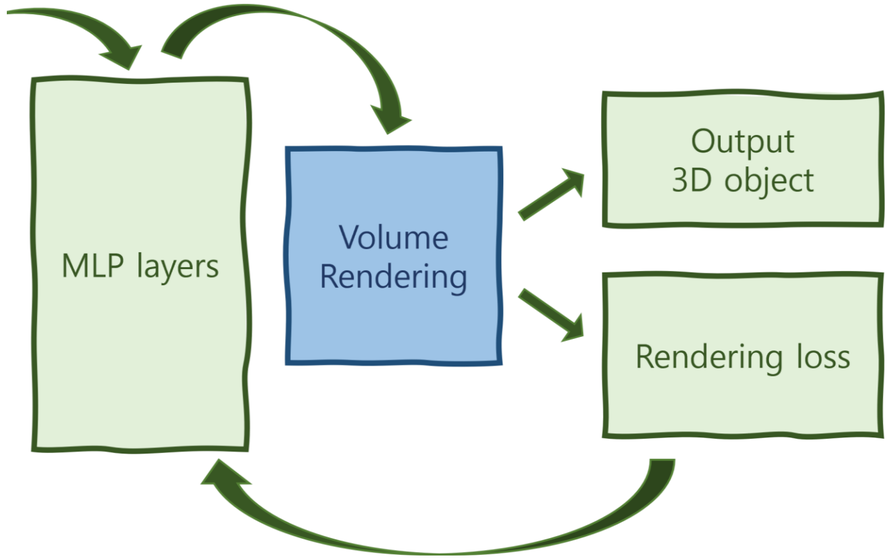
* Hierarchical Volume Sampling

$$\hat{C}_c(\mathbf{r}) = \sum_{i=1}^{N_c} w_i c_i \,, \qquad w_i = T_i(1 - \exp(-\sigma_i \delta_i))$$

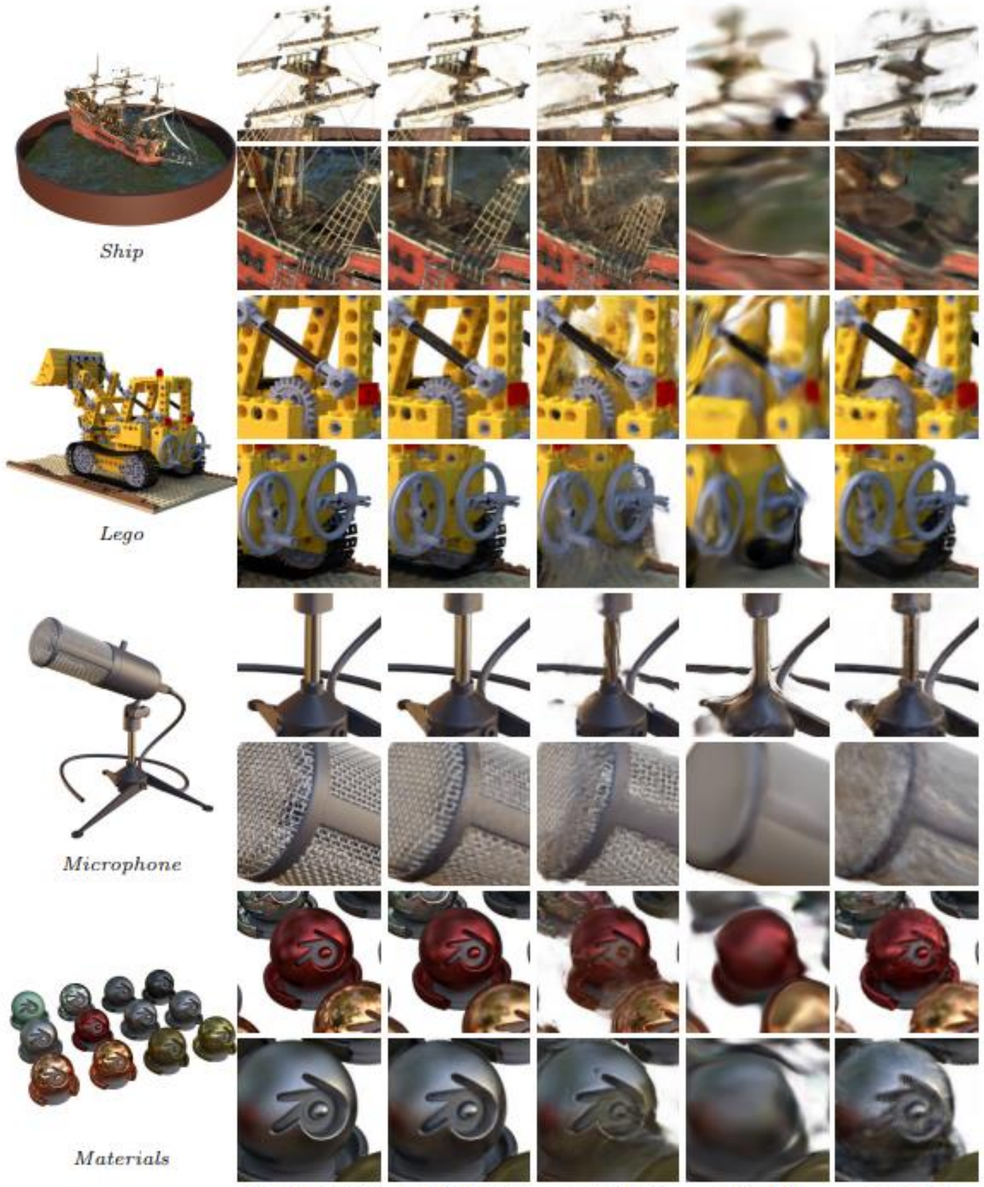
* Density가 높은 값들(실제로 물체가 있을 확률이 높은 point) 위주로 다시 sampling하여
Volume Rendering을 수행 -> 의미 있는 point에 대해서 다시 Rendering하여 Fine model 생성

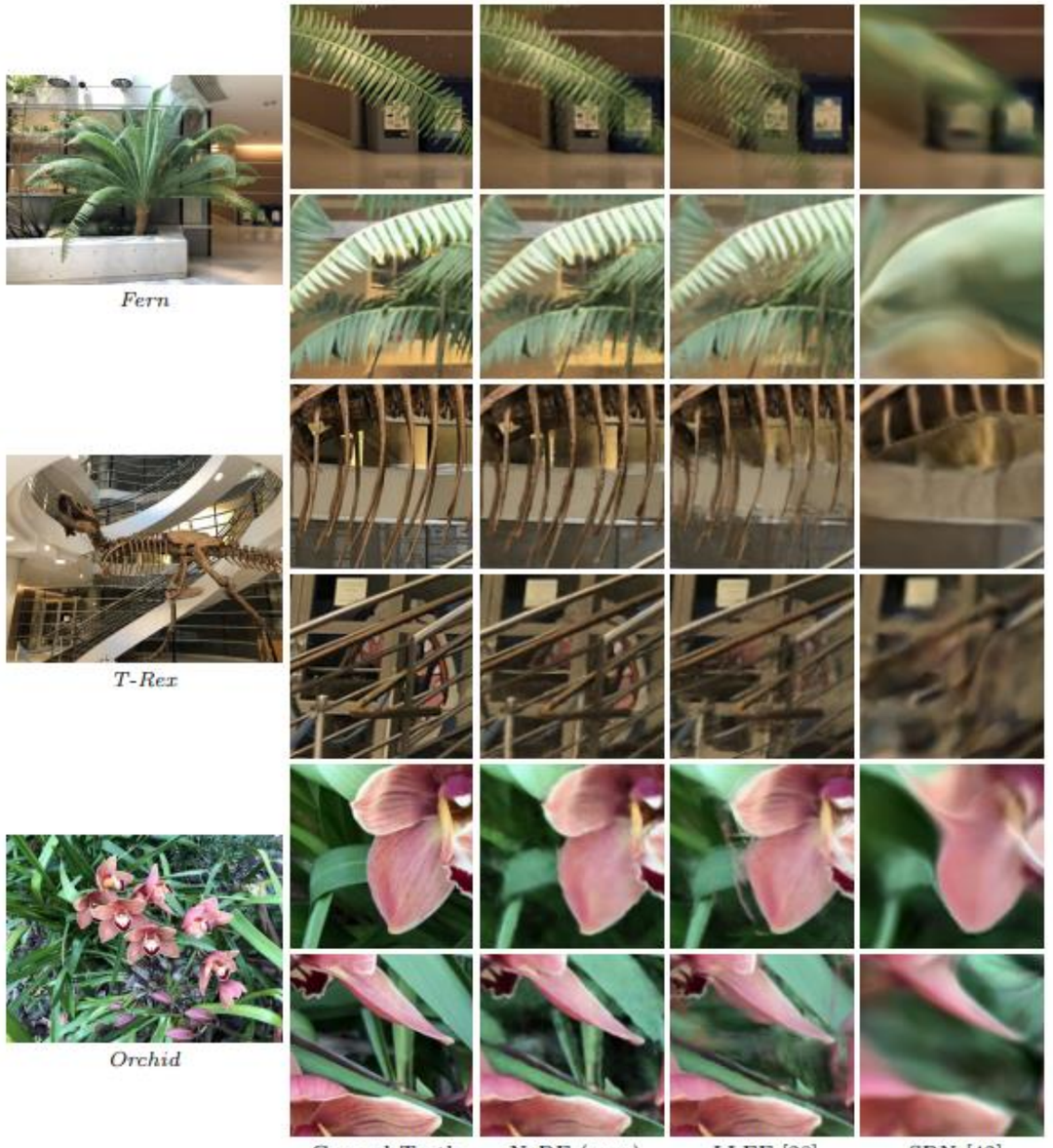
* Loss Function (MSE loss)

$$\mathcal{L} = \sum_{\mathbf{r}\in\mathcal{R}} \left[\underbrace{\left\| \hat{C}_c(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2}_{\text{Coarse Model}} + \underbrace{\left\| \hat{C}_f(\mathbf{r}) - C(\mathbf{r}) \right\|_2^2}_{\text{Fine Model}} \right]$$



2. NeRF – Results





2. NeRF – Results

* Datasets

Method	Diffuse Synthetic 360° [41]			Realistic Synthetic 360°			Real Forward-Facing [28]		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
SRN [42]	33.20	0.963	0.073	22.26	0.846	0.170	22.84	0.668	0.378
NV [24]	29.62	0.929	0.099	26.05	0.893	0.160	-	-	-
LLFF [28]	34.38	0.985	0.048	24.88	0.911	0.114	24.13	0.798	**0.212**
Ours	**40.15**	**0.991**	**0.023**	**31.01**	**0.947**	**0.081**	**26.50**	**0.811**	0.250

2. NeRF – Results

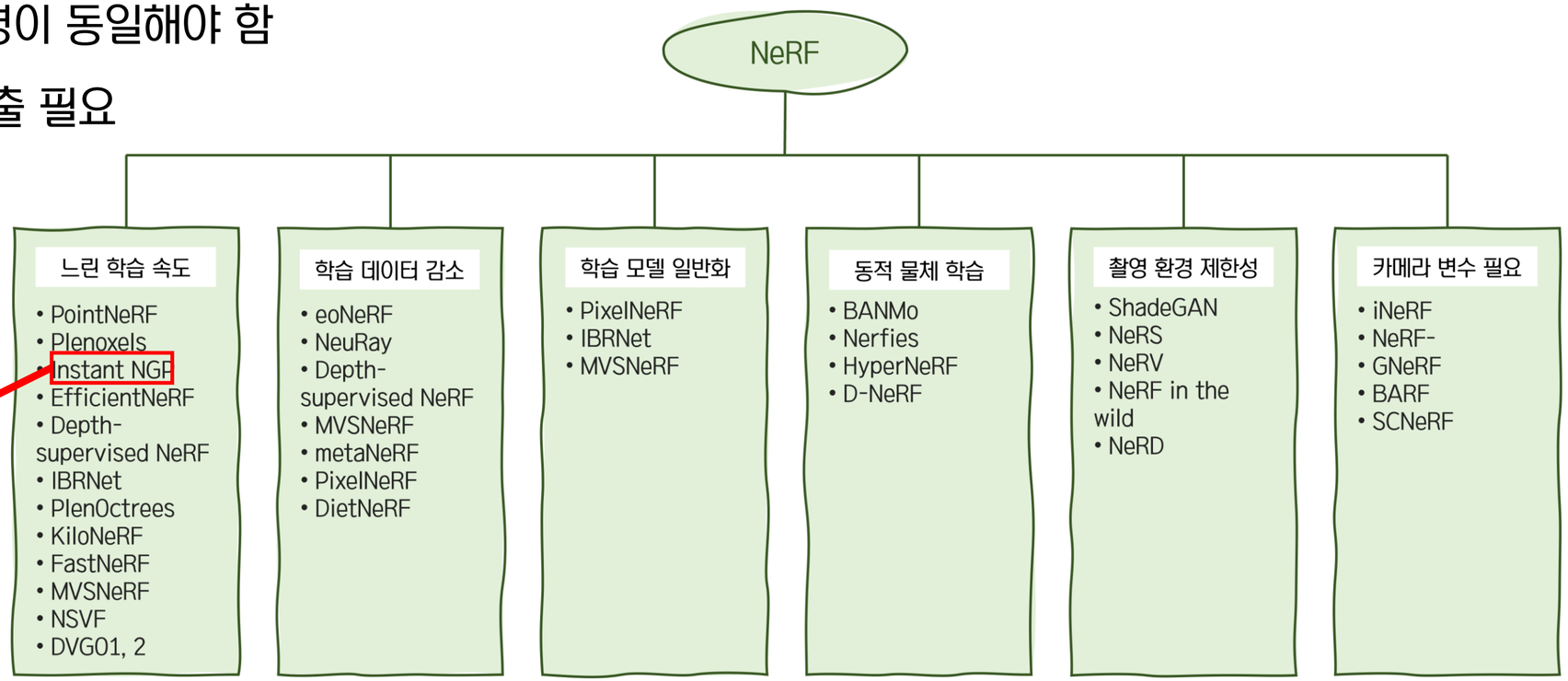
* Ablation Studies

	Input	#Im.	L	(N_c , N_f)	PSNR↑	SSIM↑	LPIPS↓
1) No PE, VD, H	xyz	100	-	(256, -)	26.67	0.906	0.136
2) No Pos. Encoding	$xyz\theta\phi$	100	-	(64, 128)	28.77	0.924	0.108
3) No View Dependence	xyz	100	10	(64, 128)	27.66	0.925	0.117
4) No Hierarchical	$xyz\theta\phi$	100	10	(256, -)	30.06	0.938	0.109
5) Far Fewer Images	$xyz\theta\phi$	25	10	(64, 128)	27.78	0.925	0.107
6) Fewer Images	$xyz\theta\phi$	50	10	(64, 128)	29.79	0.940	0.096
7) Fewer Frequencies	$xyz\theta\phi$	100	5	(64, 128)	30.59	0.944	0.088
8) More Frequencies	$xyz\theta\phi$	100	15	(64, 128)	30.81	0.946	0.096
9) Complete Model	$xyz\theta\phi$	100	10	(64, 128)	**31.01**	**0.947**	**0.081**

2. NeRF – Discussions

1. 느린 학습 속도
2. 너무 많이 필요한 Input Images
3. 한 알고리즘 모델에 하나의 물체만 학습 가능
4. 움직이는 물체 학습 불가
5. Input Images의 촬영 환경이 동일해야 함
6. Input으로 카메라 변수 추출 필요

현재 가장 빠른 SOTA model



Instant Neural Graphics Primitives

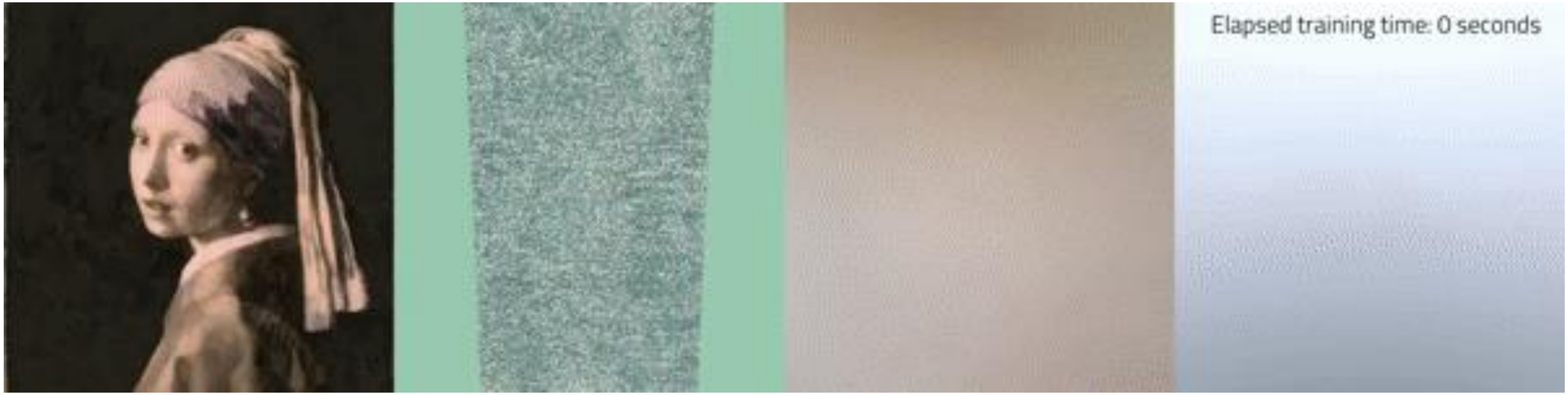
EWHA EURON

3. Instant Neural Graphics Primitives

SIGGRAPH 2022 Best Paper

현재 가장 빠른 SOTA model

Instant NGP github 링크

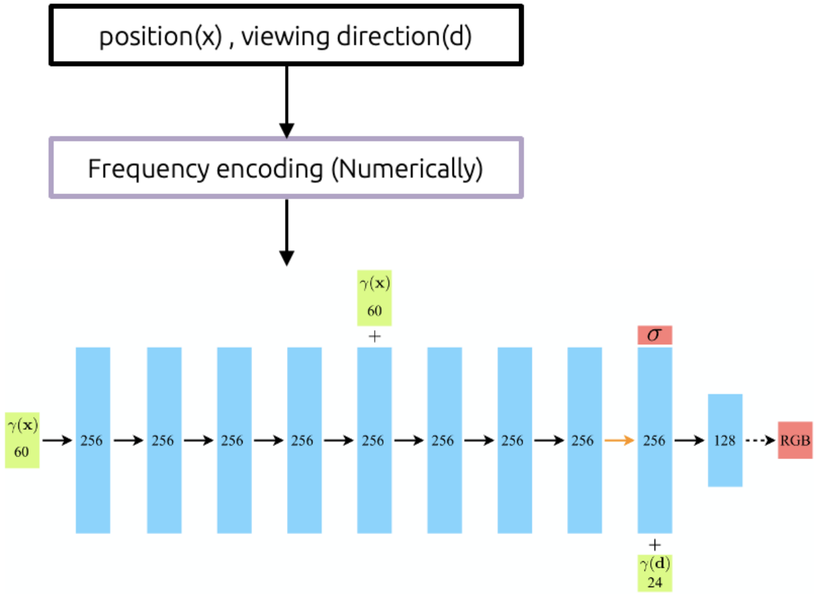


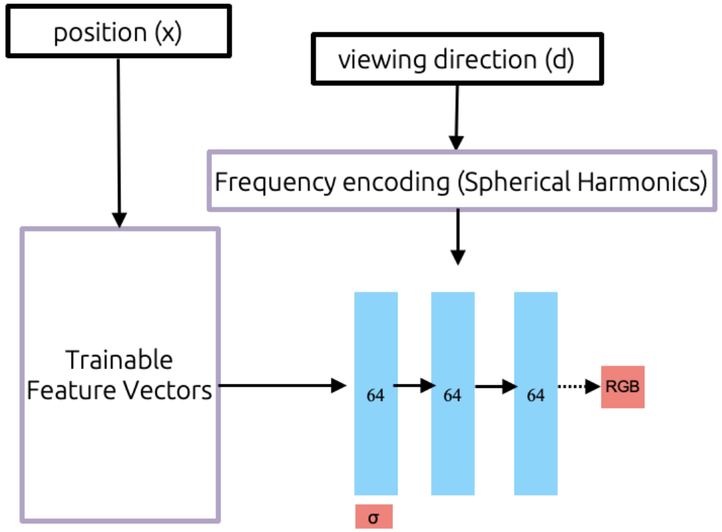
3. Instant NGP vs NeRF

*** Positional Encoding (Data augmentation의 역할)**

$$\gamma(p) = \left(\sin\left(2^0\pi p\right), \cos\left(2^0\pi p\right), \cdots, \sin\left(2^{L-1}\pi p\right), \cos\left(2^{L-1}\pi p\right)\right)$$

-> 늘어난 data 수만큼 MLPs inference가 늘어남 = NeRF의 학습 속도가 느린 핵심 원인





3. Instant NGP – Main Methods

* Multi-Resolution Hash Encoding

$$b := \exp\left(\frac{\ln N_{\max} - \ln N_{\min}}{L-1}\right)$$

1. Multi-Level Decomposition

$$N_l := \lfloor N_{\min} \cdot b^{l-1} \rfloor.$$

전체 scene 을 multi-level 로 나누어 저장하여 각 level 별로 scene geometry 의 다른 부분에 집중할 수 있도록 한다.

2. Hash Function

해상도가 높은 Voxel 일수록 저장해야하는 point 의 수가 size 의 세제곱에 비례하여 늘어나기 때문에, 모든 점에 대한 1:1 저장을 하지 않고 hash function 을 도입하여 필요한 메모리를 줄인다.

$$\mathbf{x}_l := \mathbf{x}_l \cdot N_l \quad h(x) = \left(\bigoplus_{i=1}^{d} x_i \pi_i\right) \mod T$$

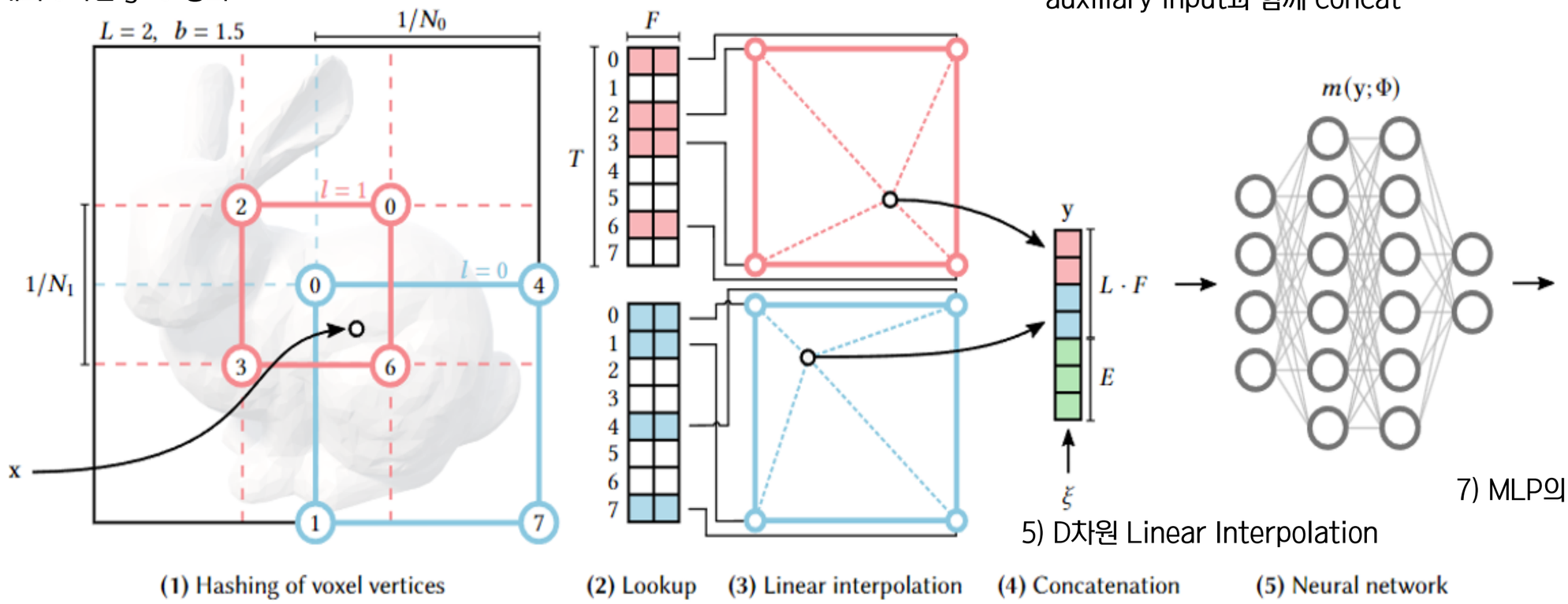
EWHA EURON

3. Instant NGP – Main Methods

1) L개의 d차원 grid 정의

3) Input Coordinates가 voxel로 mapping

6) Interpolate된 각 Level의 vector들을 auxiliary input과 함께 concat



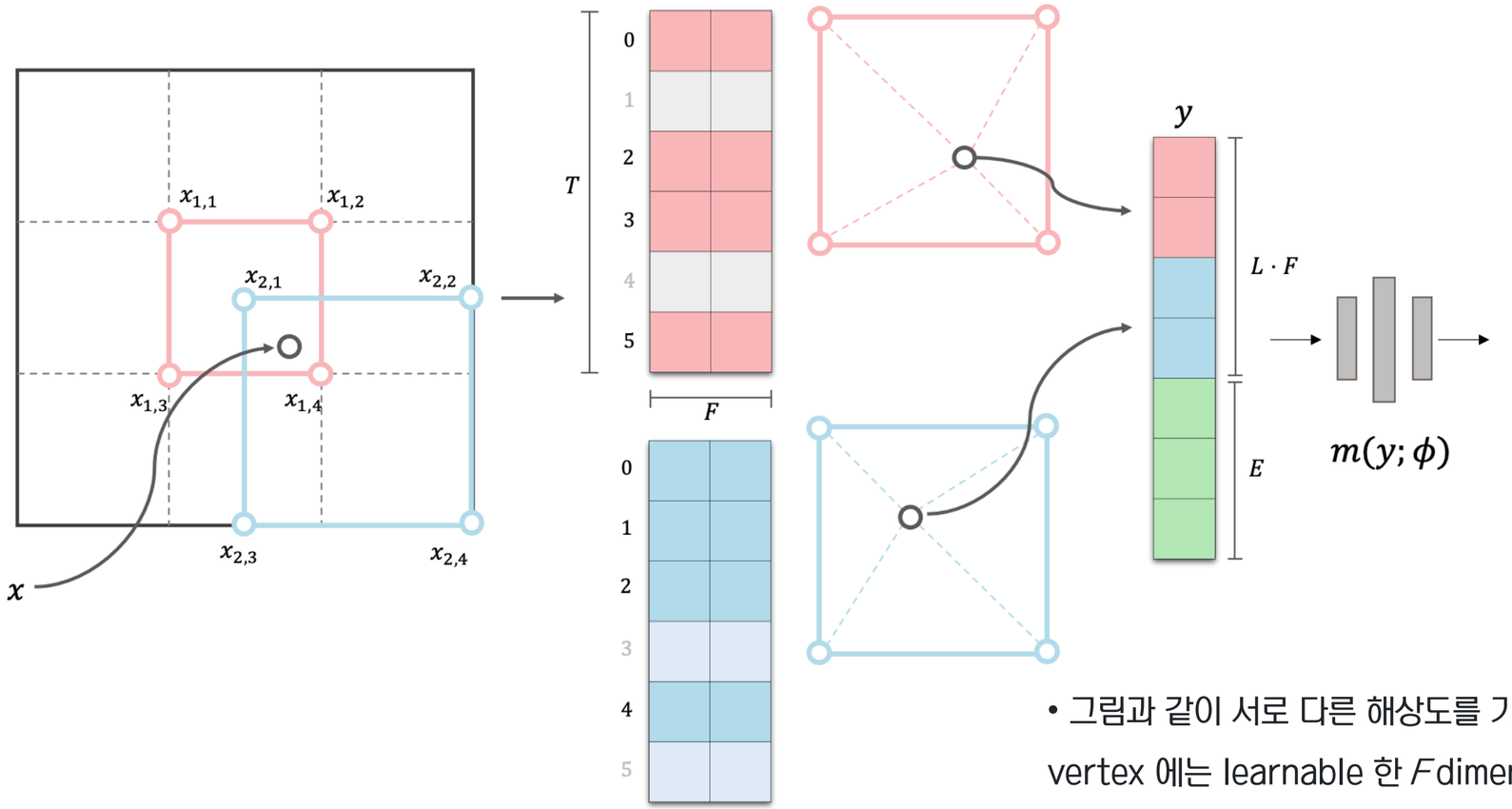
5) D차원 Linear Interpolation

7) MLP의 input으로 입력

2) 각각의 Level을 F차원의 T개 feature vector에 저장

4) Mapping된 Voxel의 꼭짓점(vertices)들이 feature vector로 mapping

3. Instant NGP – Multiresolution Hash Encoding



• 그림과 같이 서로 다른 해상도를 가진 level 에 대해서 (Red, Blue), 각 voxel 의 vertex 에는 learnable 한 F dimension 의 feature vector 를 table 에 저장한다. 이 때 table 과 vertex 간의 mapping 은 vertex 좌표에 대한 hashing 으로 정의된다.

• 공간 위의 어떤 한 점에 대해서, 이 점의 encoding 은 점이 속한 hypercube 의 모든 corner vertex feature 간의 linear interpolation 으로 결정되고,

• 이 값이 view-direction encoding 과 합쳐져서 decoding network $m(\mathbf{y};\phi)$ 에 input 으로 들어가게 된다.

3. Instant NGP – Multiresolution Hash Encoding

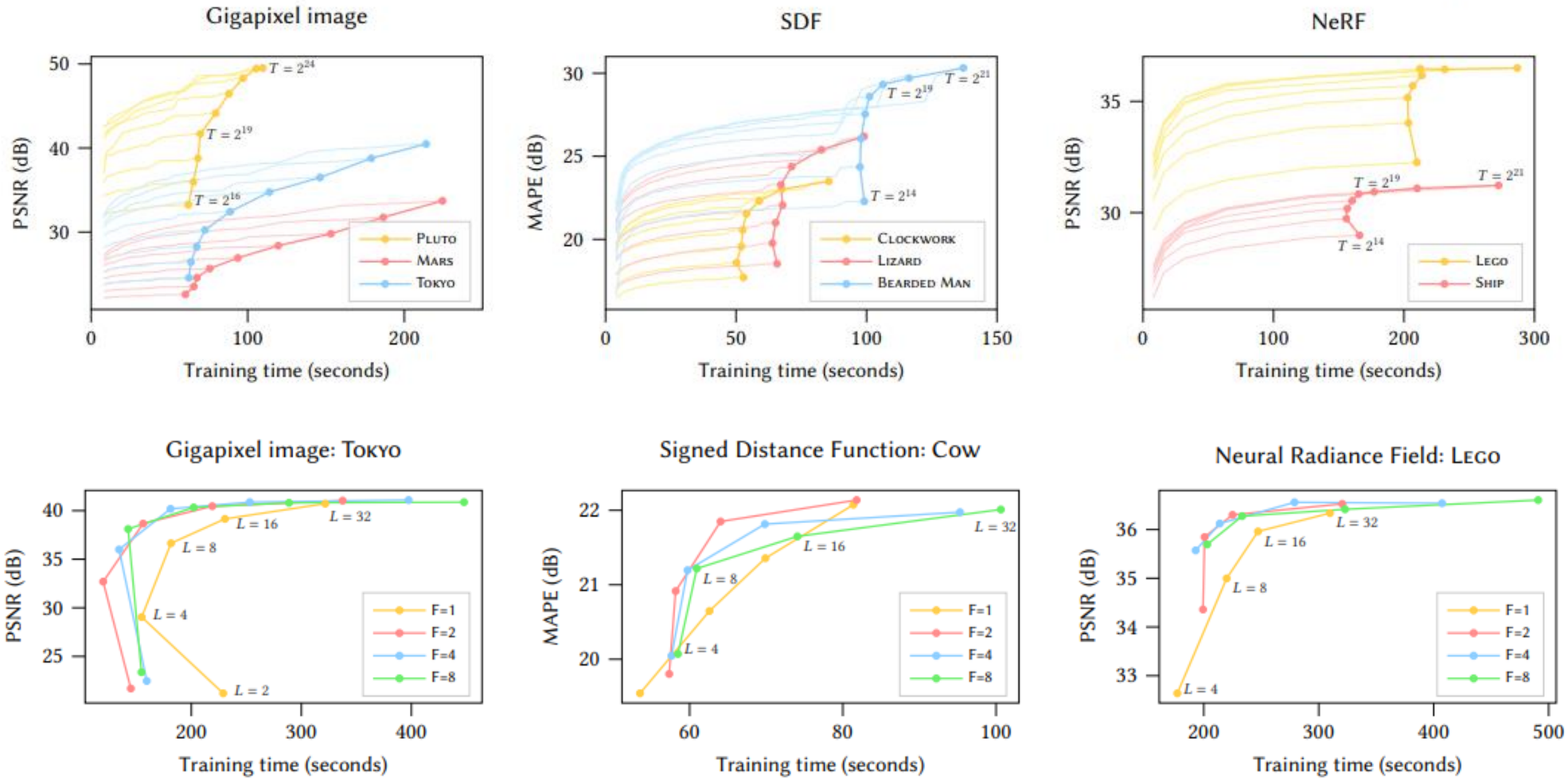
$$b := \exp\left(\frac{\ln N_{\max} - \ln N_{\min}}{L-1}\right)$$

$$N_l := \lfloor N_{\min} \cdot b^{l-1} \rfloor.$$

Parameter	Symbol	Value
Number of levels	L	16
Max. entries per level (hash table size)	T	2^{14} to 2^{24}
Number of feature dimensions per entry	F	2
Coarsest resolution	$N_{\min}$	16
Finest resolution	$N_{\max}$	512 to 524288

3. Instant NGP – Results

* PSNR (Peak Signal-to-noise ratio)
영상 화질 손실양을 평가하기 위해 사용되는 지표



3. Instant NGP – Results

	Mic	Ficus	Chair	Hotdog	Materials	Drums	Ship	Lego	avg.
Ours: Hash (1 s)	26.09	21.30	21.55	21.63	22.07	17.76	20.38	18.83	21.202
Ours: Hash (5 s)	32.60	30.35	30.77	33.42	26.60	23.84	26.38	30.13	29.261
Ours: Hash (15 s)	34.76	32.26	32.95	35.56	28.25	25.23	28.56	33.68	31.407
Ours: Hash (1 min)	35.92	33.05	34.34	36.78	29.33	25.82	30.20	35.63	32.635
Ours: Hash (5 min)	36.22	33.51	35.00	37.40	29.78	26.02	31.10	36.39	33.176
mip-NeRF (~hours)	36.51	33.29	35.14	37.48	30.71	25.48	30.41	35.70	33.090
NSVF (~hours)	34.27	31.23	33.19	37.14	32.68	25.18	27.93	32.29	31.739
NeRF (~hours)	32.91	30.13	33.00	36.18	29.62	25.01	28.65	32.54	31.005
Ours: Frequency (5 min)	31.89	28.74	31.02	34.86	28.93	24.18	28.06	32.77	30.056
Ours: Frequency (1 min)	26.62	24.72	28.51	32.61	26.36	21.33	24.32	28.88	26.669

3. Instant NGP – Discussion

* spatial coordinate → feature 의 mapping이 랜덤이다. (hash function)

1. Hash Collision

microstructure artifacts의 발생

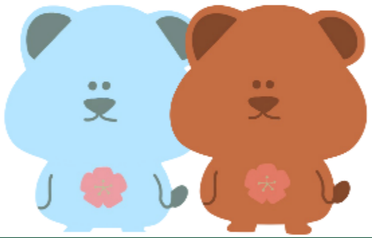
2. No spatial locality of features

feature를 coordinate space와 관련된 data structure에 보관 불가능 (다른 모델은 가능)

3. Requires Bigger Code book

* mappin을 잘 하면 code book이 작아도 될 것으로 추정됨

Discussion



EWHA EURON

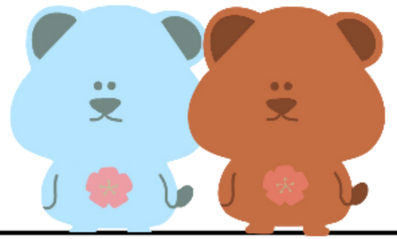
4. 특이했던 점들

1. Computer Graphic 시장에서의 AI 기술 전망
2. 알고리즘 속도 향상 기술의 전략
3. 복잡할 필요가 없는 AI 구조

4. Discussion

1. C++… 공부해야 할까요…?
2. Hash functio이 아이디어로 사용된 것에 대한 의견
3. 실상 CUDA와 C++을 사용했기 때문에 생긴 속도 차이도 있지 않을까?

THANK YOU



EWHA EURON