

NAF: Neural Attenuation Fields for Sparse-View CBCT Reconstruction

Ruyi Zha, Yanhao Zhang, and Hongdong Li

MICCAI 2022

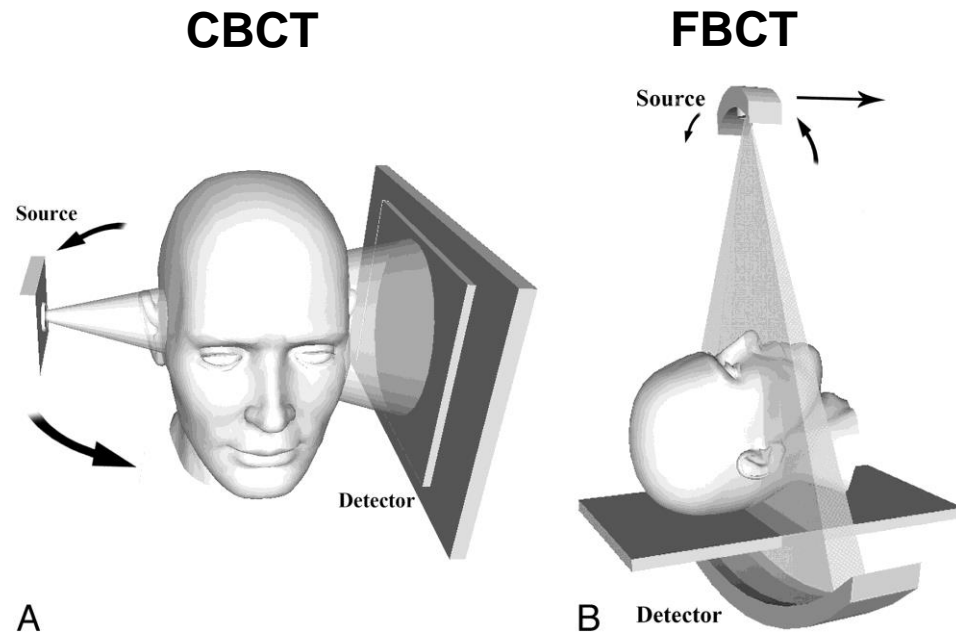
발표자 : 박강길

Contents

- Introduction
 - Background
 - Method
 - Experiment
 - Conclusion
-

Introduction

❖ Sparse-view CBCT for low dose scan



- CBCT: 원뿔형태의 X선을 이용해서 촬영하는 의료영상 기술
- CBCT는 FBCT 보다 높은 공간 해상도, 빠른 스캔 속도를 가짐
- CT 촬영에서 낮은 방사선량으로 촬영하는 low dose CT에 대한 관심 증가
- 피폭을 줄이는 방법
 - Source intensity 줄이기
 - Projection views 줄이기 (sparse-view)
- Sparse-view CBCT 영상으로 3D reconstruction에 대한 연구

Introduction

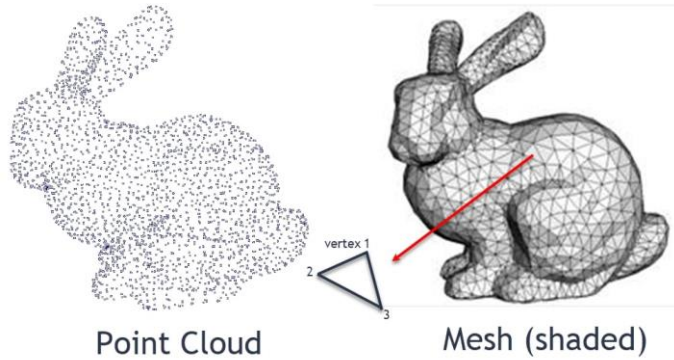
❖ CBCT reconstruction

“CT(Computed Tomography) 영상의 intensity는 해당 위치에서 측정된 X선의 흡수량을 나타냅니다. CT 스캔에서 측정된 X선의 흡수량은 CT attenuation coefficient(CT 감쇄계수)라는 값으로 표현되며, 이 값을 기반으로 CT 영상의 intensity가 결정됩니다.”

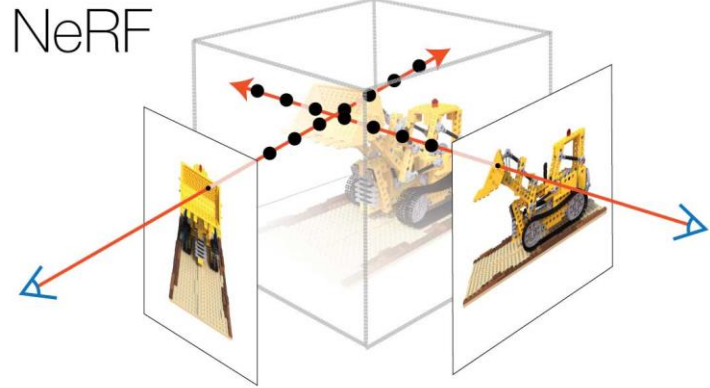
- Sparse-view CBCT reconstruction은 수십개의 projection으로 부터 volumetric attenuation coefficient를 구하는 것을 목표로 함
 - Sparse-view CBCT로 3D reconstruction은 어려움
 - view가 충분하지 않아 artifacts가 심함 (Traditional CBCT 보다 10배 정도 적음)
 - 공간, 계산 복잡도가 FBCT보다 높음 (FBCT : 1D projections → 2D slices → stacking 2D slices(3D), CBCT : 2D projections → 3D model)
 - CBCT reconstruction approaches
 - 1) Analytical method : Radon transform을 풀어서 attenuation coefficient를 구함 ex) FDK algorithm
→ Sparse views에서는 성능이 안 좋음
 - 2) Iterative method : 최적화 문제로 공식화 해서 해결 ex) SART, ASD-POCS
→ Memory와 시간이 너무 많이 필요
 - 3) Learning-based method : Deep-learning, projection 예측, attenuation coefficient 예측 등등
→ Data가 많이 필요함
→ Network가 CT가 어떻게 생겼는지 기억하기 때문에 다른 CT에 대해 적용이 어려움 ex) Abdomen → Aorta
-

Introduction

❖ Neural Attenuation Fields (NAF)



- RGB image 3D reconstruction
 - 1) Point clouds, meshes를 이용해서 reconstruction
 - 2) **Implicit Neural Representation**(INR)을 이용해서 reconstruction 하는 연구 ex) **NeRF**
 - 공간좌표를 color & density(이미지 intensity를 구하는데 필요한 변수)로 mapping 하는 neural network를 이용 (2D → 3D)



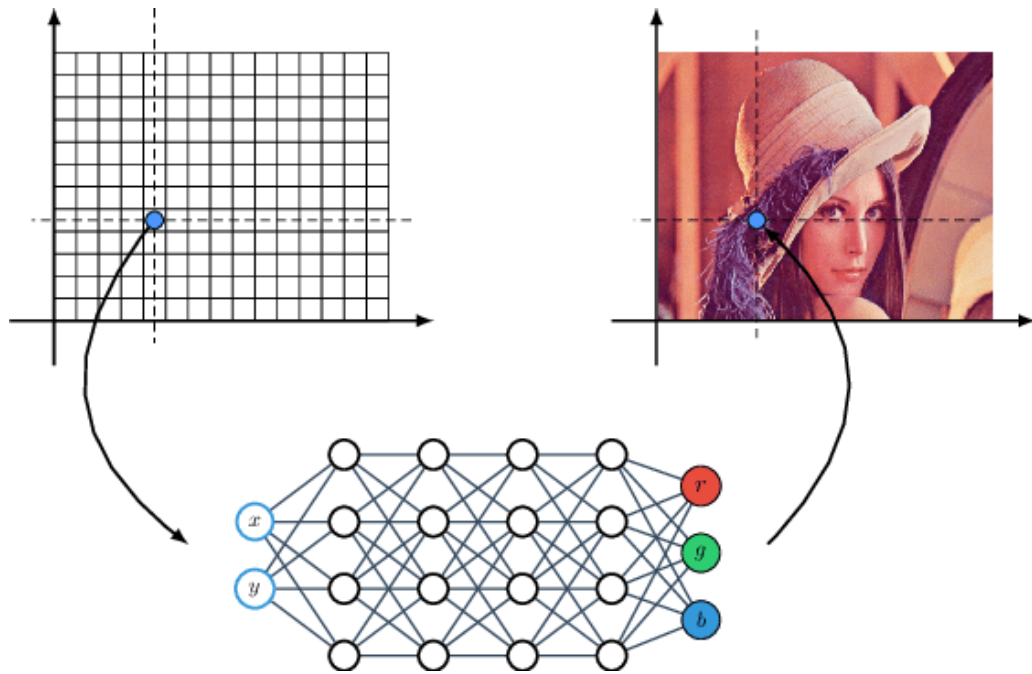
- **Neural Attenuation Fields**
 - 공간좌표(x, y, z)를 attenuation coefficient로 mapping 하는 neural network를 구성
- Fast self-supervised solution for sparse-view CBCT reconstruction
 - No external CT scans
 - Hash encoding 방법을 사용해서 빠르게 reconstruction을 할 수 있음

Implicit Neural Representation(INR), Neural Fields, NeRF ??? → Background

Background

❖ Implicit Neural Representation

"Implicitly defined, continuous, differentiable signal representations parameterized by neural networks" -
Sitzmann, Vincent, et al. "Implicit neural representations with periodic activation functions." *Advances in Neural Information Processing Systems*










- 이미지의 각 픽셀은 특정 RGB 값을 가지고 있고 이 픽셀들이 모여서 이미지를 생성
- 이미지를 함수로 표현 \rightarrow 좌표 (x, y) 를 입력으로 그 위치의 RGB를 출력하는 함수
- 함수 자체가 사실상 이미지 = network parameter들이 이미지와 동일한 정보를 담고 있음

“Implicit Neural Representation은 어떤 정보를 Neural Network로 나타내거나 혹은 저장”

Background

❖ Neural Fields

Neural Fields in Visual Computing and Beyond

Yiheng Xie^{1,2}  Towaki Takikawa^{3,4} Shunsuke Saito⁵  Or Litany⁴  Shiqin Yan¹ Numair Khan¹  Federico Tombari^{6,7}
James Tompkin¹  Vincent Sitzmann^{8†}  Srinath Sridhar^{1†} 

¹Brown University ²Unity Technologies ³University of Toronto ⁴NVIDIA ⁵Meta Reality Labs Research ⁶Google ⁷Technical University of Munich
⁸Massachusetts Institute of Technology [†]*Equal advising*

Field

Definition 1 A *field* is a quantity defined for all spatial and/or temporal coordinates.

Neural Field

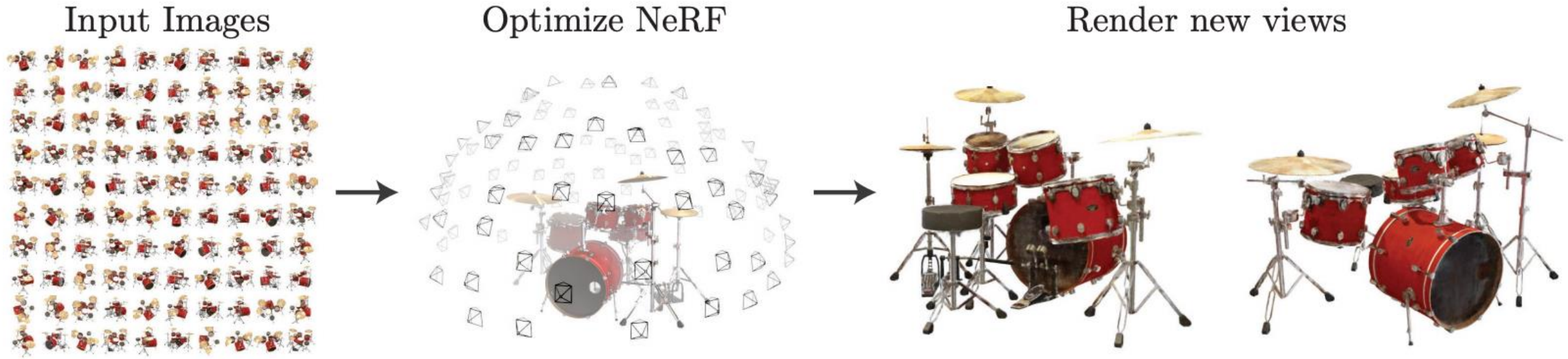
Definition 2 A *neural field* is a field that is parameterized fully or in part by a neural network.

→ “We can represent a field as a function mapping a coordinate x to a quantity, which is typically a scalar or vector.”

✓ NAF (Neural Attenuation Fields) : Coordinate(input)를 Attenuation으로 mapping 하는 함수

Background

❖ Novel view synthesis

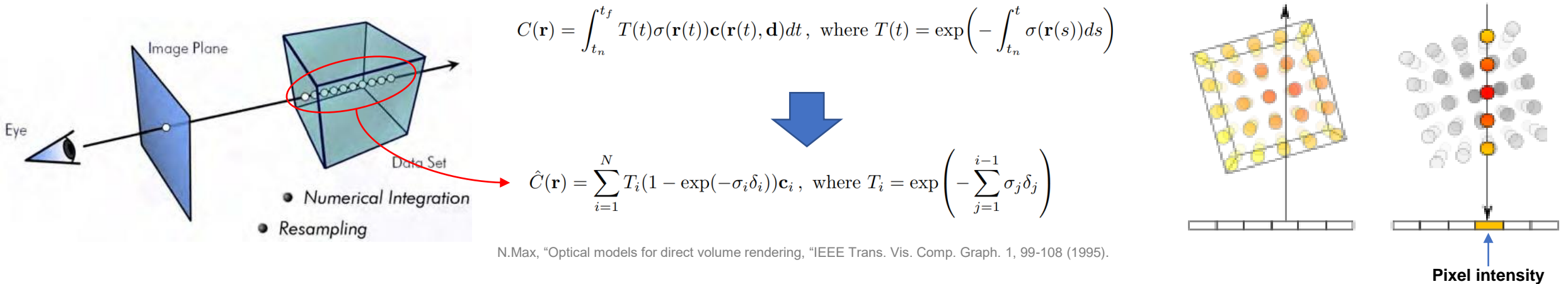


“서로 다른 시점에서 찍은 여러 장의 이미지를 이용해서 새로운 시점의 이미지를 생성”

Background

❖ Volume ray casting

- volume ray casting : 이미지 기반의 volume rendering(3D data를 2D projection으로 보여주는 것) 기술



Eye(카메라) : 카메라로부터 현재 volume을 바라보는 위치와 카메라 렌즈에서 물체로 향하는 ray의 방향이 정해짐
Image plane : 3D volume 데이터가 2D로 projection 되는 plane

1. Image plane의 각 픽셀에서 카메라로부터 각각 하나의 ray가 volume으로 투사됨
2. Ray가 물체의 표면에 닿았을 때 정지시키지 않고 계속 뚫고 나아가도록 함
3. 물체를 뚫고 나간 ray를 통해 물체(point)를 샘플링 함
4. 샘플링한 point의 RGB와 density를 이용해서 rendering 식을 통해 Image plane의 pixel intensity를 계산

Background

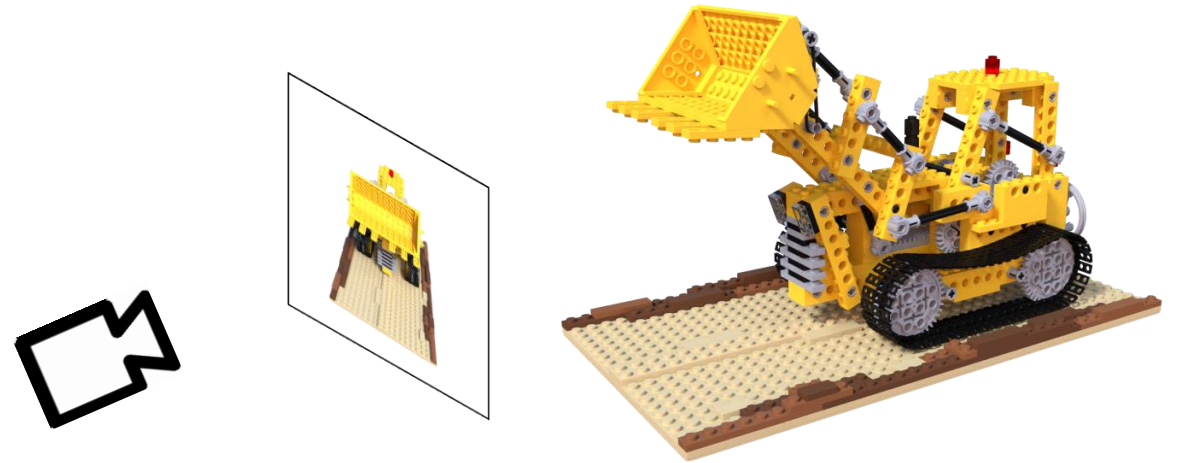
❖ Neural Radiance Fields (NeRF)

Data

여러 시점에서 찍은 사진



각각 사진들을 찍을 때 사용한 카메라 정보

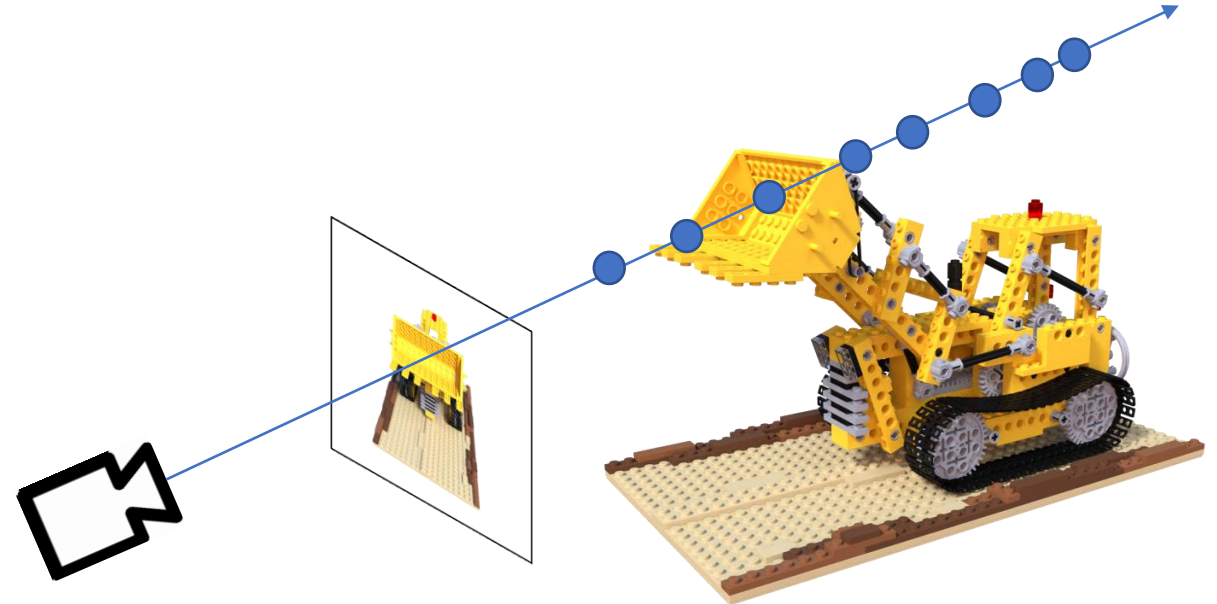
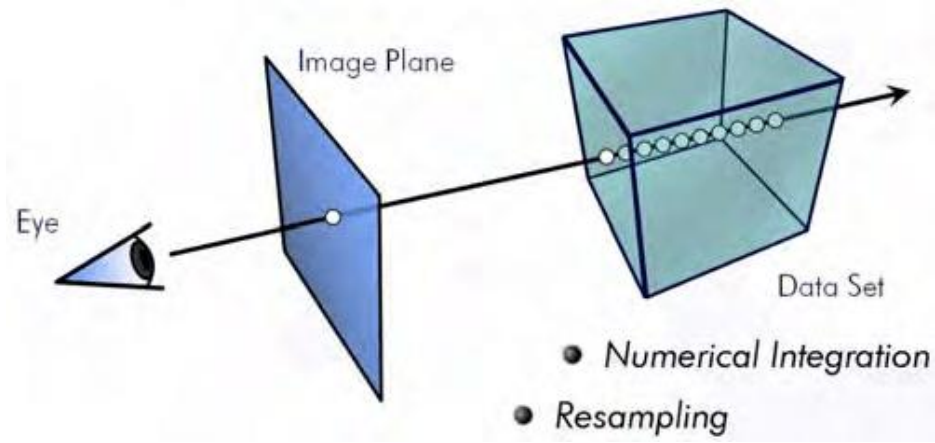


사진을 찍을 당시 위치, 렌즈가 물체를 향하는 방향

Background

❖ Neural Radiance Fields (NeRF)

Volume ray casting



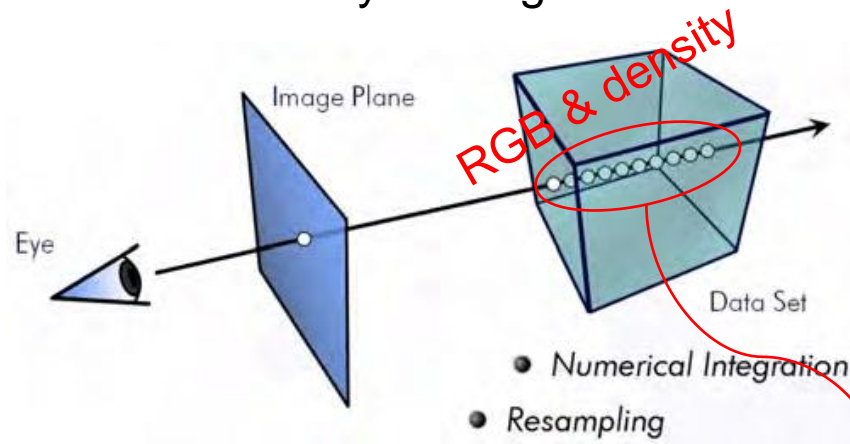
카메라 위치 정보 = Ray의 시작점

- 이미지의 한 픽셀을 지나는 ray를 그릴 수 있음
- ray에서 point들을 sampling 할 수 있음

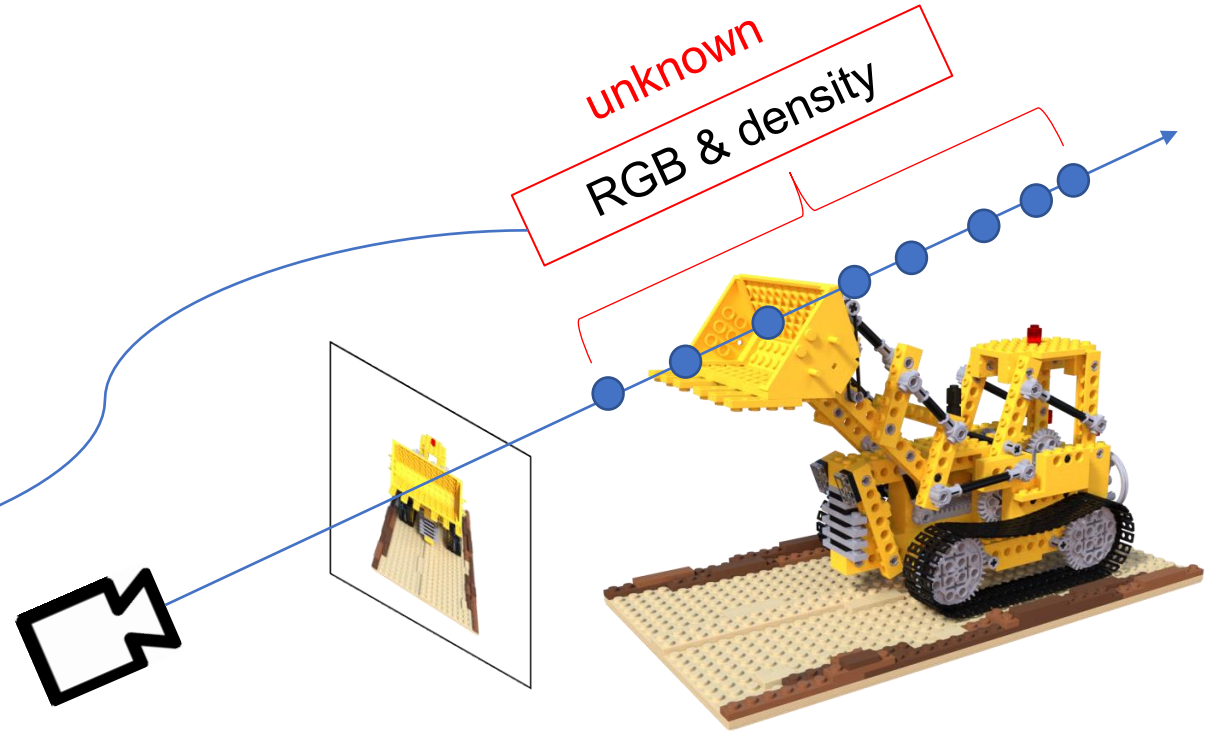
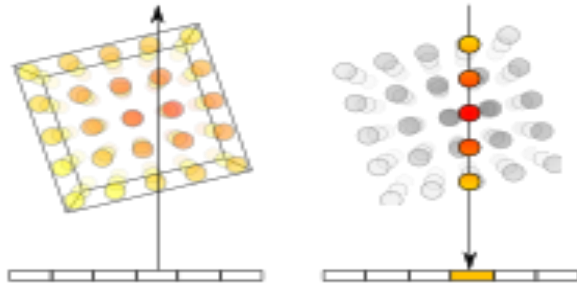
Background

❖ Neural Radiance Fields (NeRF)

Volume ray casting



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

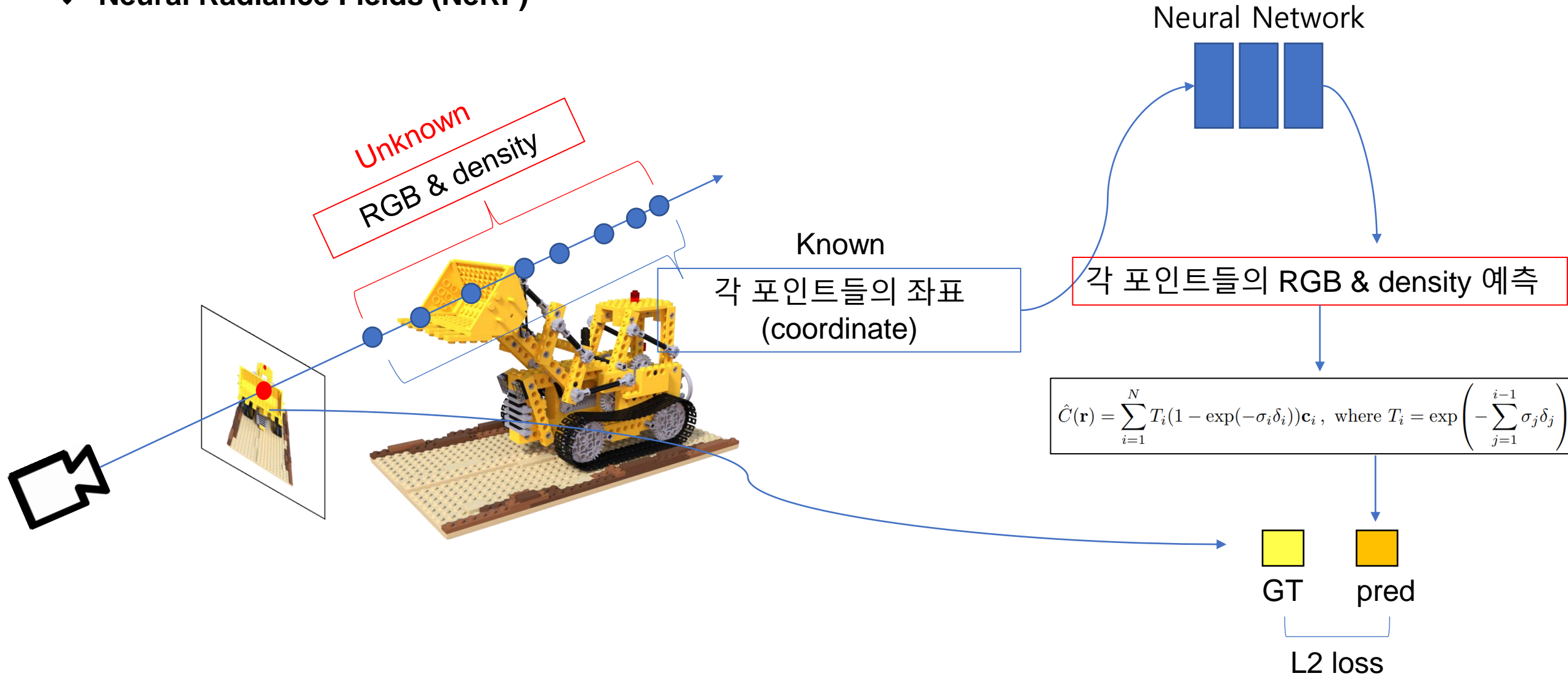


카메라 위치 = Ray의 시작점

- 포인트들의 RGB & density로 pixel의 intensity를 계산

Background

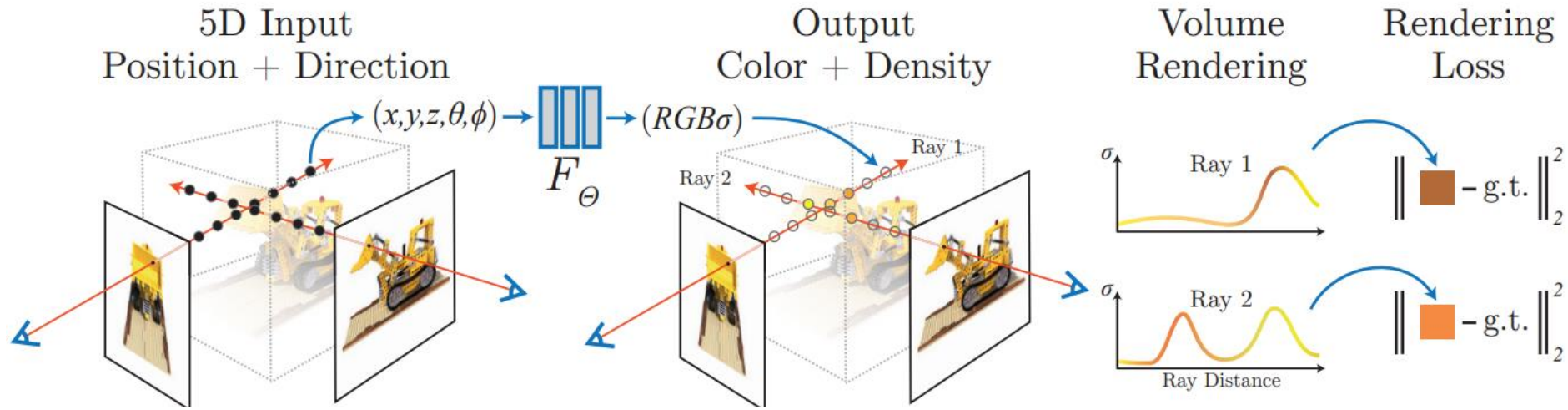
❖ Neural Radiance Fields (NeRF)



"NeRF: Neural Radiance Fields(Coordinate 를 넣어서 RGB & density를 mapping 하는 함수)"

Background

❖ Neural Radiance Fields (NeRF)



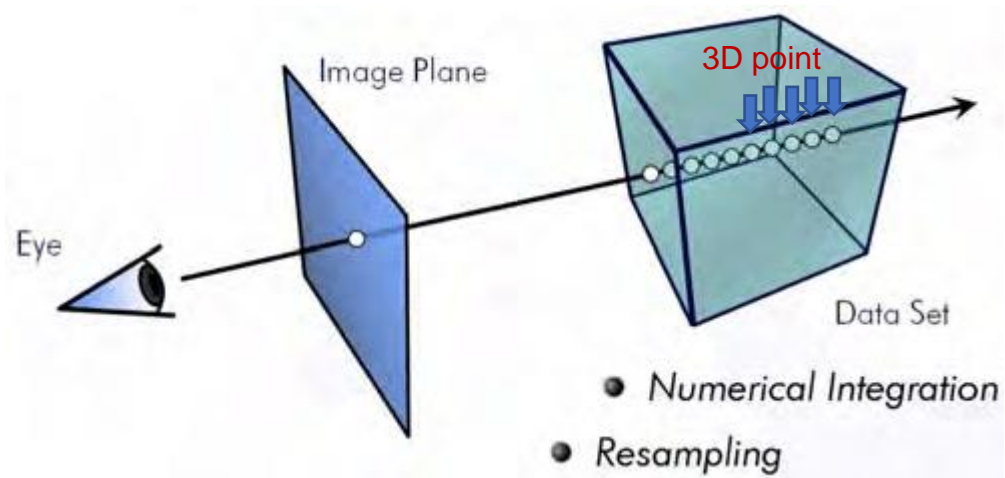
1. Ray를 지나는 공간상의 3D point들(x, y, z)로 모델 input 으로 구성
2. 3D 포인트 (x, y, z) 좌표와 ray의 방향(θ, ϕ) 정보를 모델의 input으로 주어 RGB와 density를 output으로 출력
3. RGB와 density를 volume rendering 식에 입력으로 넣어 이미지 픽셀의 intensity를 계산
4. 계산한 intensity와 GT pixel과 L2 loss 계산

Background

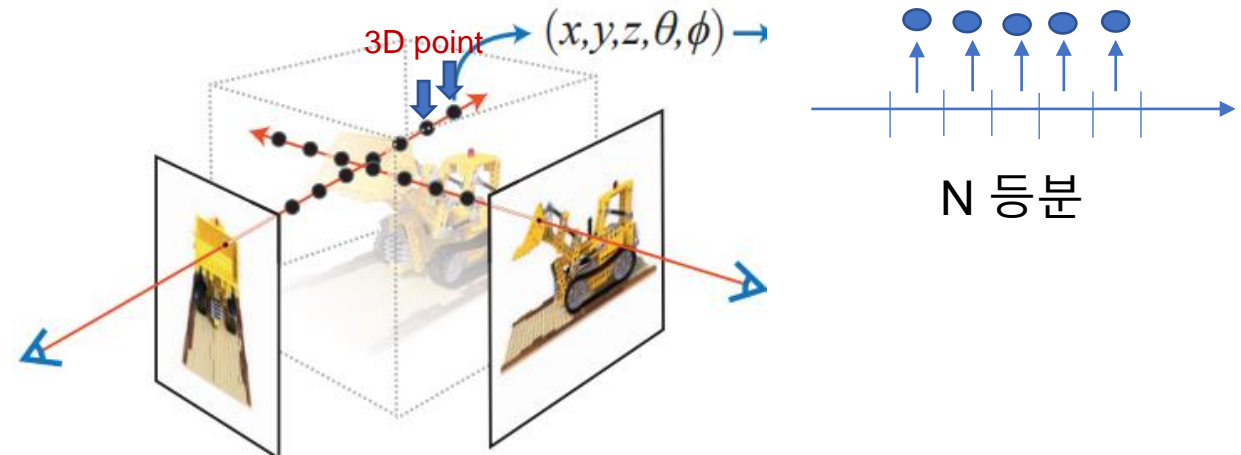
❖ Ray sampling

이미지 픽셀 값을 결정하는 가중치 큰 point를 sampling 하는 것은 중요

Volume Ray Casting



Stratified sampling



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

t_n : 카메라가 렌더링하는 객체와 가장 가까운 포인트

t_f : 카메라가 렌더링 하는 객체와 가장 먼 포인트

Background

❖ Positional encoding

Sampling한 point 들..



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

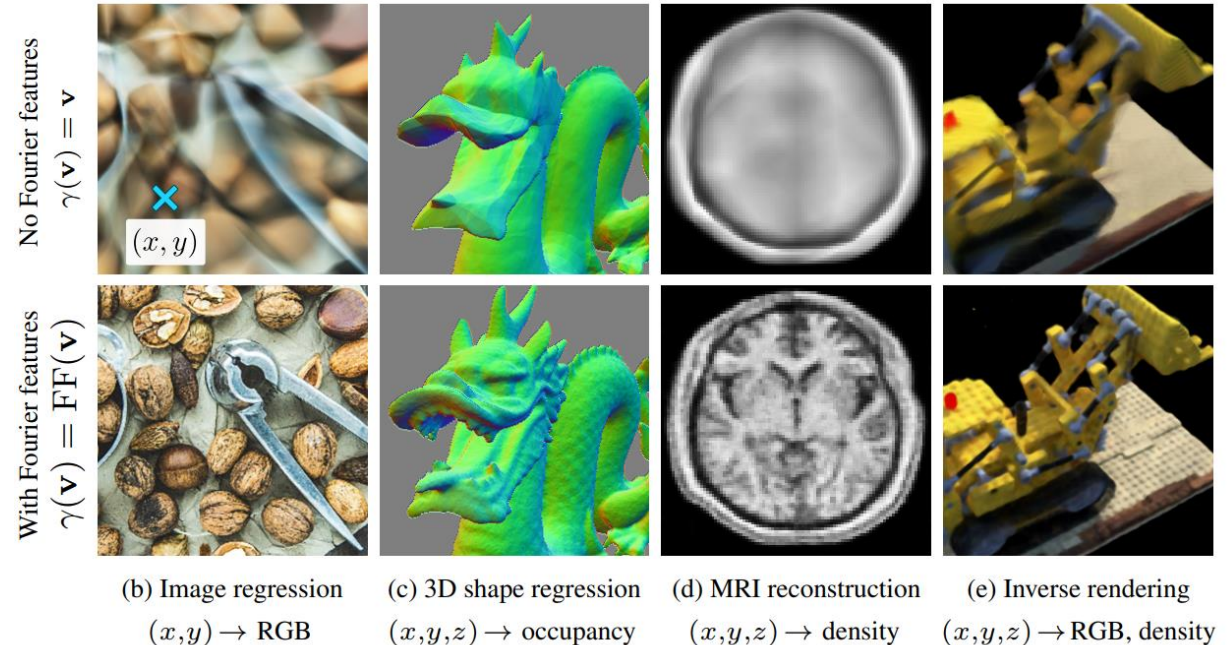
Spectral bias : MLP가 low-frequency details만 학습하려는 특성

Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains

Matthew Tancik^{1*} Pratul P. Srinivasan^{1,2*} Ben Mildenhall^{1*} Sara Fridovich-Keil¹

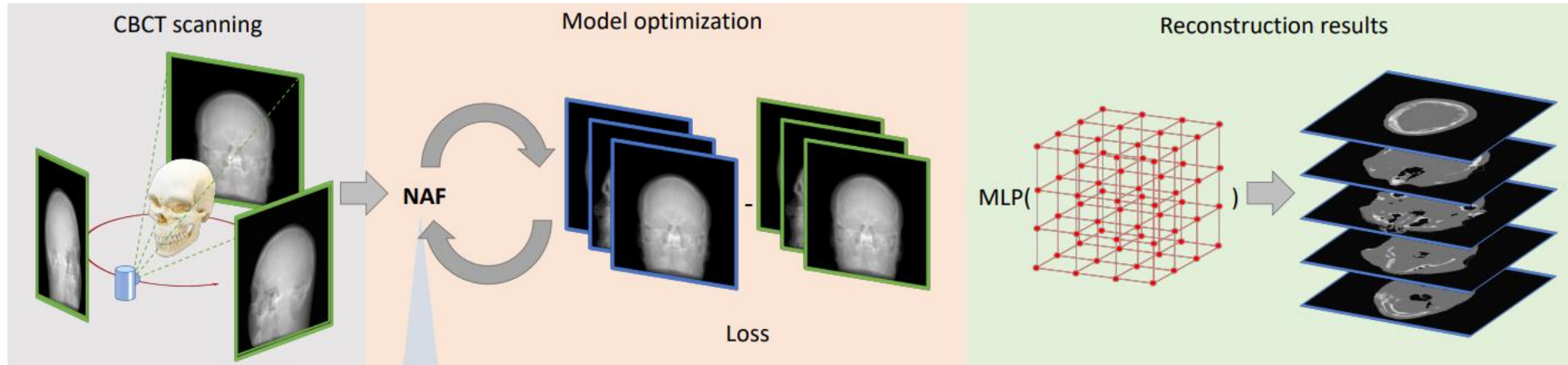
Nithin Raghavan¹ Utkarsh Singhal¹ Ravi Ramamoorthi³ Jonathan T. Barron² Ren Ng¹

¹University of California, Berkeley ²Google Research ³University of California, San Diego



Method

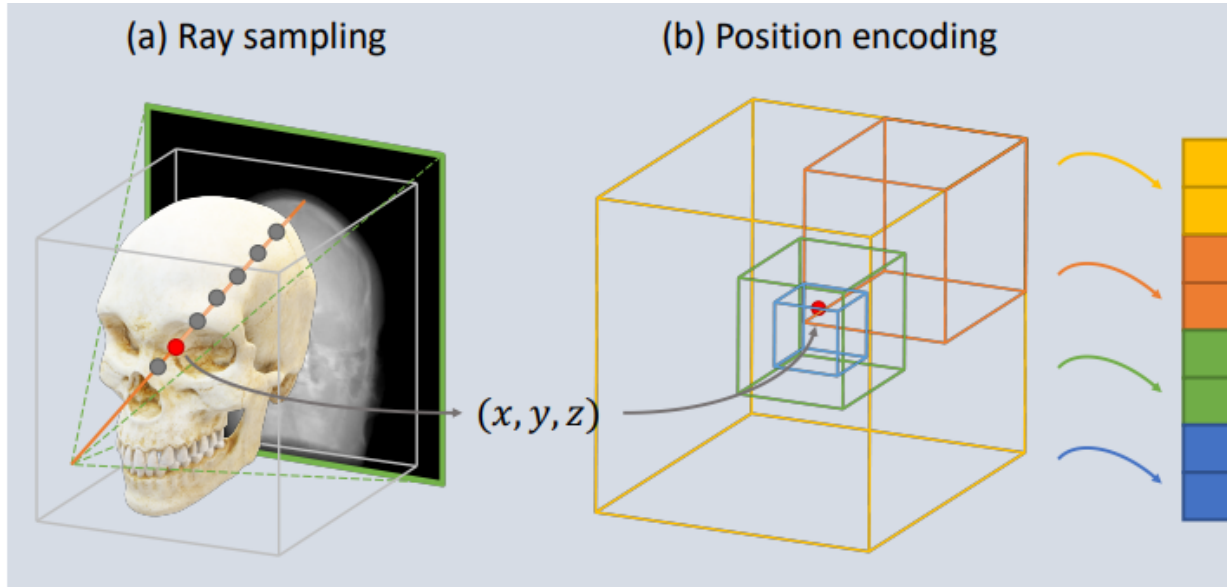
❖ Neural Attenuation Fields (NAF)



1. X-ray를 지나는 3D point들(x, y, z)로 모델 input으로 구성
2. 3D 포인트 (x, y, z) 집합을 모델의 input으로 주어 attenuation coefficient를 output으로 출력
3. Attenuation coefficient를 CT intensity를 계산하는 함수에 입력으로 넣어 intensity를 계산
4. 계산한 intensity와 GT pixel과 L2 loss 계산

Method

❖ Ray sampling & Position encoding



Ray sampling

- NeRF와 동일한 stratified sampling method 사용
- N은 얻고자 하는 CT size(slice) 보다 크게 설정
→ X-ray가 통과하는 모든 grid cell 안에 point가 들어가도록 하기 위함

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

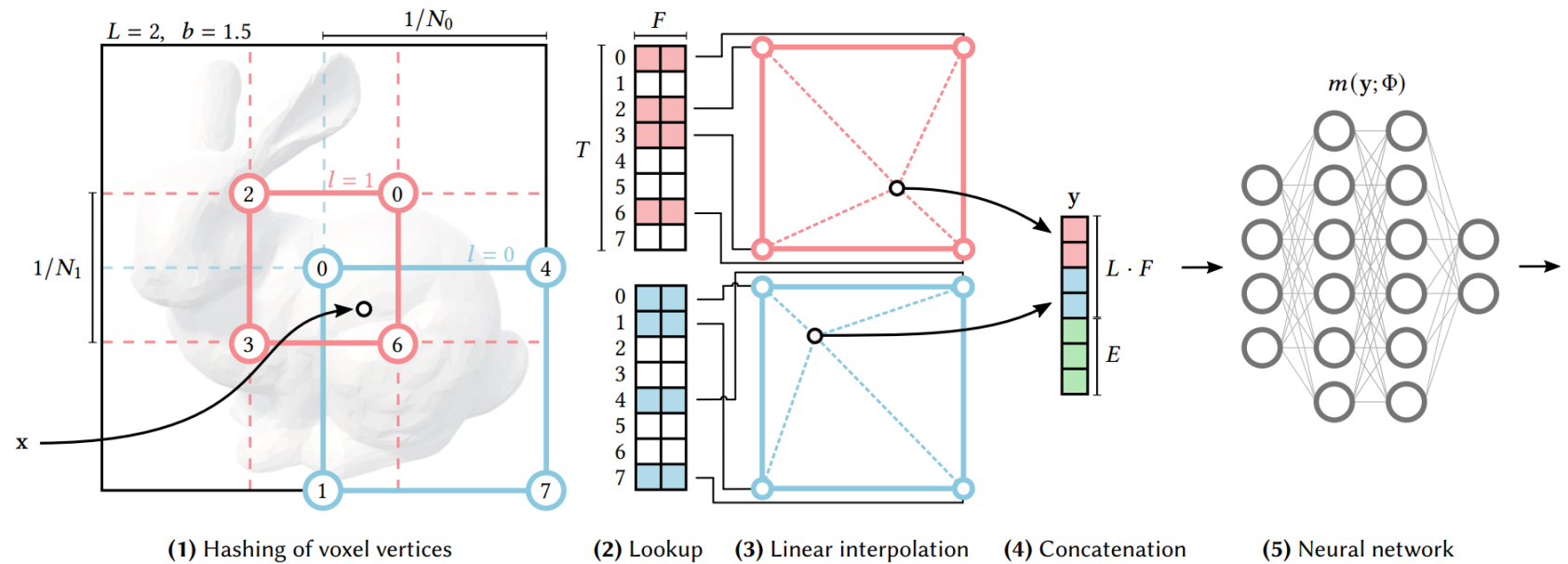
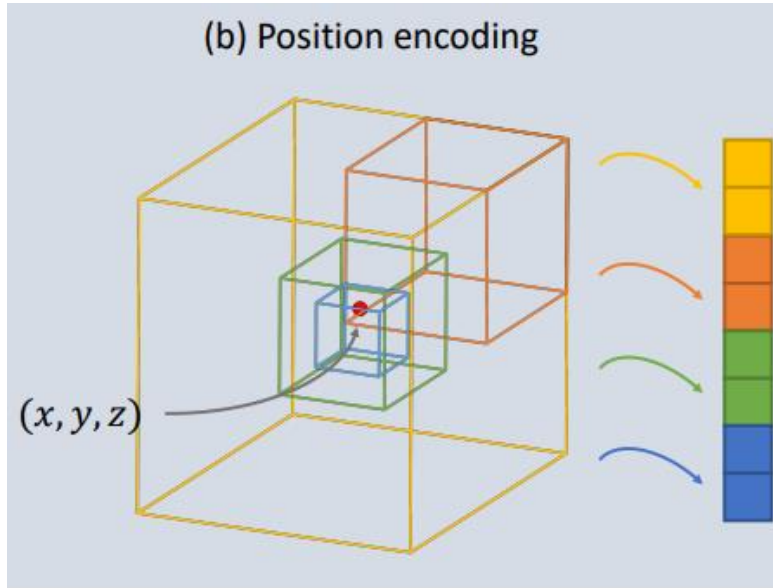
Position encoding

Frequency encoder

- High frequency detail을 잘 배울 수 있음
- wider & deeper network가 필요
→ 학습시간이 오래 걸림
→ Not acceptable for fast CT reconstruction
- 사람의 몸은 근육과 뼈로 이루어져 있음
- 동일한 매질끼리 거의 비슷한 attenuation coefficient를 가짐
→ Edge 근처가 아니면 굳이 high-frequency feature가 필요 없음
- 대부분의 organs은 simple shapes
- Low-frequency feature로 충분히 쉽게 배움
→ NAF에서는 **hash-encoding**을 사용

Method

❖ Hash encoding



Müller, T., Evans, A., Schied, C., & Keller, A. (2022). Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*

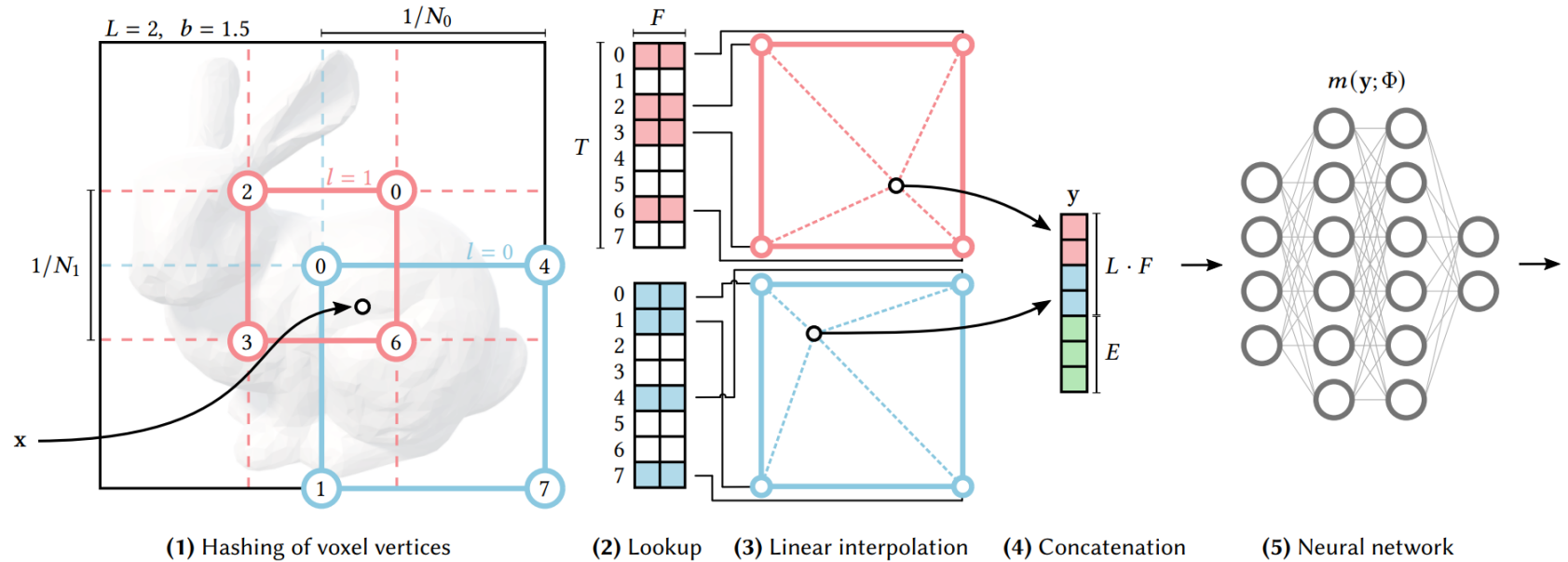
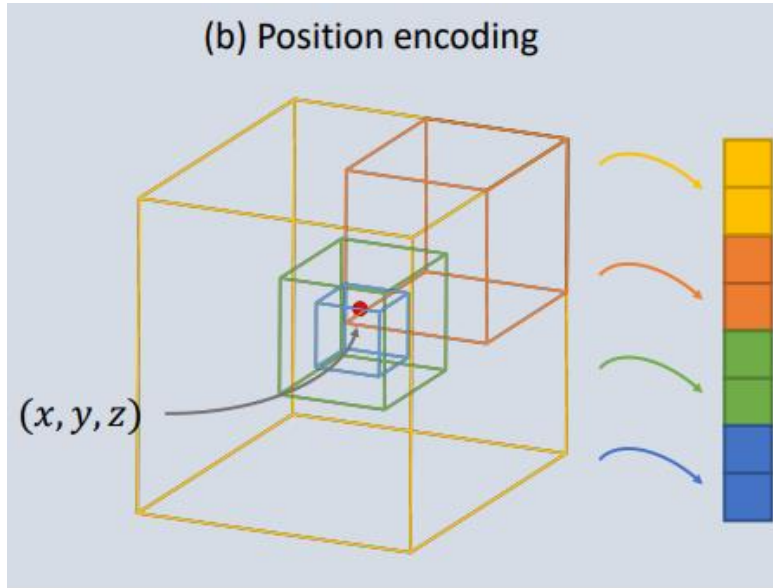
Instant-NGP

- Frequency-encoding 방법은 input dimension이 많이 커지기 때문에 계산 속도가 느림
 - 각 (INR을 활용하는) tasks 마다 좋은 결과를 얻기 위해서는 encoding 방식이 달라야 함.
- Trainable 한 hash-table을 이용한 encoding 방식으로 task agnostic 한 encoding 방식과 학습 속도 개선

hash-table : key, value로 데이터를 저장하는 자료구조 중 하나로 빠르게 데이터를 검색할 수 있는 자료구조

Method

❖ Hash encoding

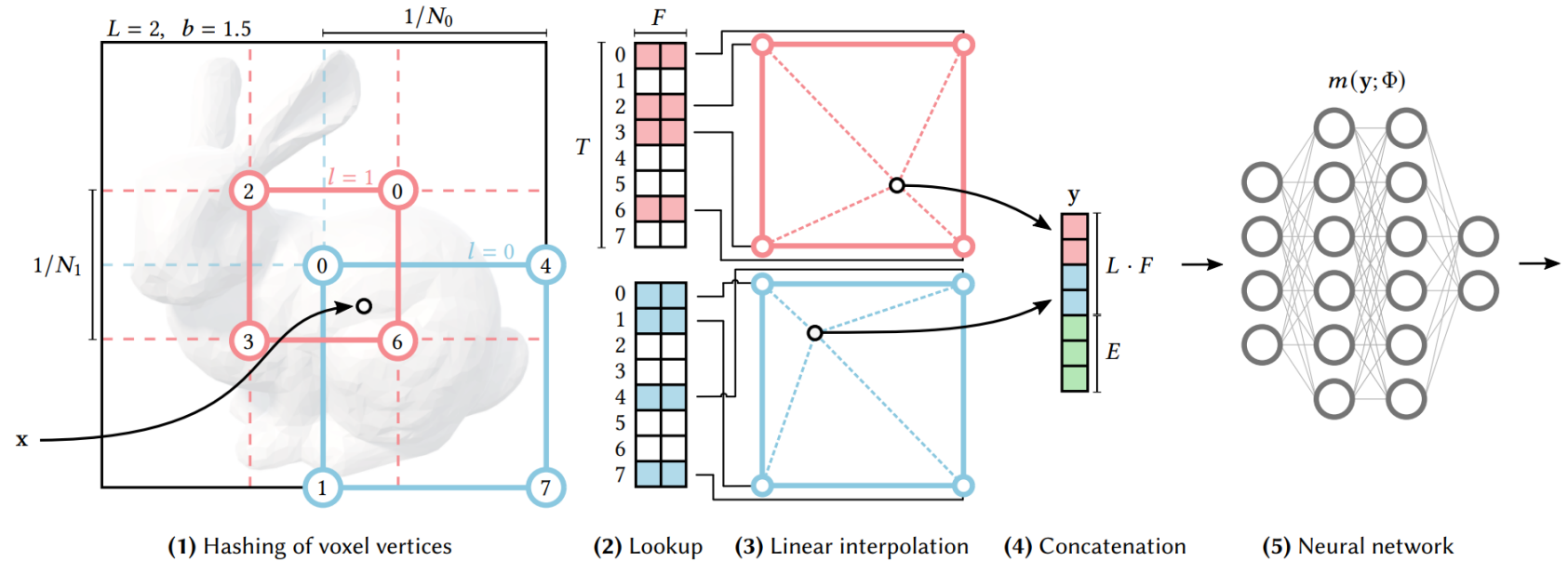
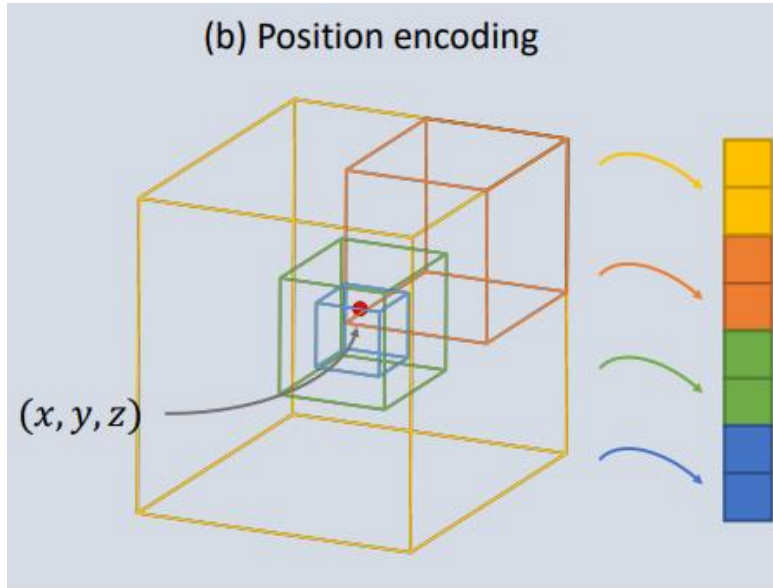


Müller, T., Evans, A., Schied, C., & Keller, A. (2022). Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (ToG)

1. Multiresolution : 전체 이미지에서 다양한 N_i 사이즈로 여러 크기의 사각형으로 만듦 (encoding 세팅)
2. 사각형의 꼭지점 위치를 index(key)로 꼭지점에 위치한 value를 hash-table에 저장 (encoding 세팅)
3. Input(point)가 들어왔을 때 input을 포함하는 여러 사이즈의 사각형들을 검색하여 사각형들의 꼭지점 value들로 linear interpolation 하여 Input의 value를 계산함
4. 각 사각형에서 구한 value들을 concat하여 model에 전달함
5. 학습을 진행하면서 hash-table이 계속 업데이트 됨

Method

❖ Hash encoding

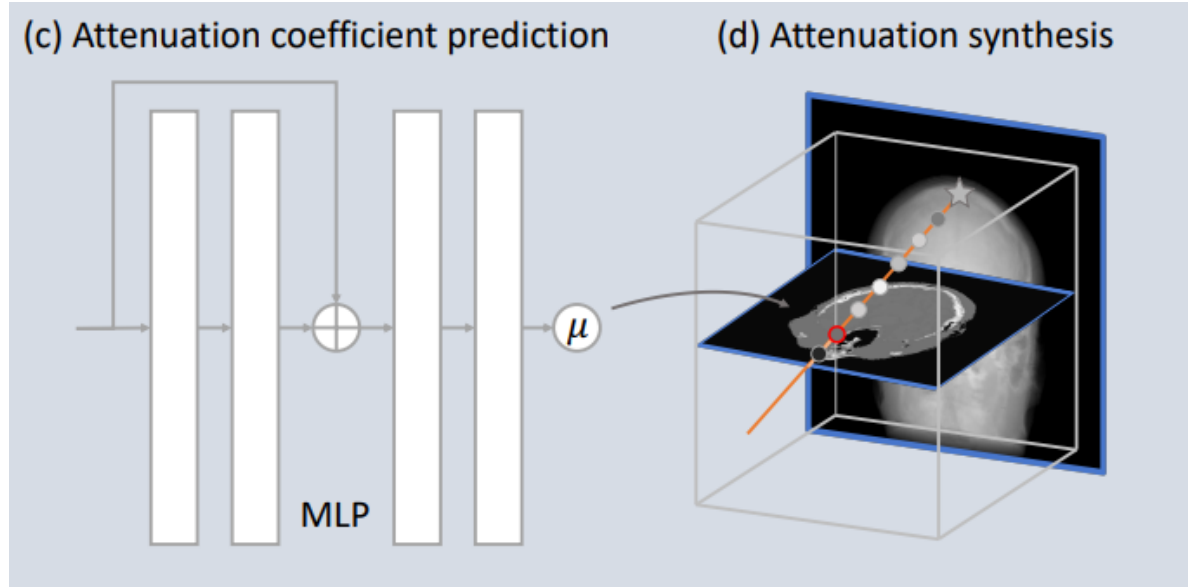


Müller, T., Evans, A., Schied, C., & Keller, A. (2022). Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*

- 인코딩 된 정보에 multiresolution 정보가 포함되어 있으면서도 frequency encoding 보다 dimension이 작음
 - trainable(hash-table) encoder라 학습하면서 결과에 도움이 되는 정보에 집중할 수 있음.
- network를 작게 만들 수 있음, 학습 속도가 빠름

Method

❖ Model architecture



Model

- 4 fully-connected layers 32-channel wide, ReLU activation
- 10x smaller (IntraTomo: Self-supervised Learning-based Tomography via Sinogram Synthesis and Prediction)
- Output : attenuation coefficients (μ)

Model optimization

$$I = I_0 \exp\left(-\sum_{i=1}^N \mu_i \delta_i\right),$$

Attenuation coefficients

$$\mathcal{L}(\Theta, \Phi) = \sum_{\mathbf{r} \in \mathbf{B}} \|I_r(\mathbf{r}) - I_s(\mathbf{r})\|^2$$

Model

hash-encoder

Experiment

❖ Experimental settings

Data

- Human organ : Chest, Jaw, Foot, Abdomen (Human organ CTs, LIDC-IDRI)
→ 3D volume data만 제공하기 때문에 TIGRE라는 toolbox를 사용하여 180도 범위에서 2D 이미지로 50장 projection 시킴
- Phantom : Aorta (silicon aortic phantom with GE C-arm)
→ -103도~93도 범위에서 2D 이미지 582 장을 촬영, 내장 알고리즘으로 GT (512x512x510 3D volume) 생성

Table 1: Details of CT datasets used in the experiments.

Dataset name	CT dimension	Scanning method	Scanning range	Number of projections	Detector resolution
Chest [4]	$128 \times 128 \times 128$	TIGRE [5]	$0^\circ \sim 180^\circ$	50	256×256
Jaw [12]	$256 \times 256 \times 256$	TIGRE [5]	$0^\circ \sim 180^\circ$	50	512×512
Foot [12]	$256 \times 256 \times 256$	TIGRE [5]	$0^\circ \sim 180^\circ$	50	512×512
Abdomen [12]	$512 \times 512 \times 463$	TIGRE [5]	$0^\circ \sim 180^\circ$	50	1024×1024
Aorta	$512 \times 512 \times 510$	GE C-arm	$-103^\circ \sim 93^\circ$	50 (582)	500×500

Experiment

❖ Result

Baseline

- Analytical method (FDK)
- Iterative reconstruction method (SART)
→ Robust한 method라고 함
- Iterative reconstruction method (ASD-POCS)
→ Total-variation regularizer 사용
(고주파 성분(예를 들면 잡음)을 제거하고 부드러운 결과물을 만들어내는 기법)
- Deep learning method (IntraTomo3D)
→ Frequency encoding 사용

Table 2: PSNR/SSIM measurements of five methods on five datasets.

	Chest	Jaw	Foot	Abdomen	Aorta
FDK [7]	22.89/.78	28.59/.78	23.92/.58	22.39/.59	12.11/.21
SART [2]	32.12/.95	32.67/.93	30.13/.93	31.38/.92	27.31/.77
ASD-POCS [20]	29.78/.92	32.78/.93	28.67/.89	30.34/.91	27.30/.76
IntraTomo3D [28]	31.94/.95	31.95/.91	31.43/.91	30.43/.90	29.38/.82
NAF (Ours)	33.05/.96	34.14/.94	31.63/.94	34.45/.95	30.34/.88

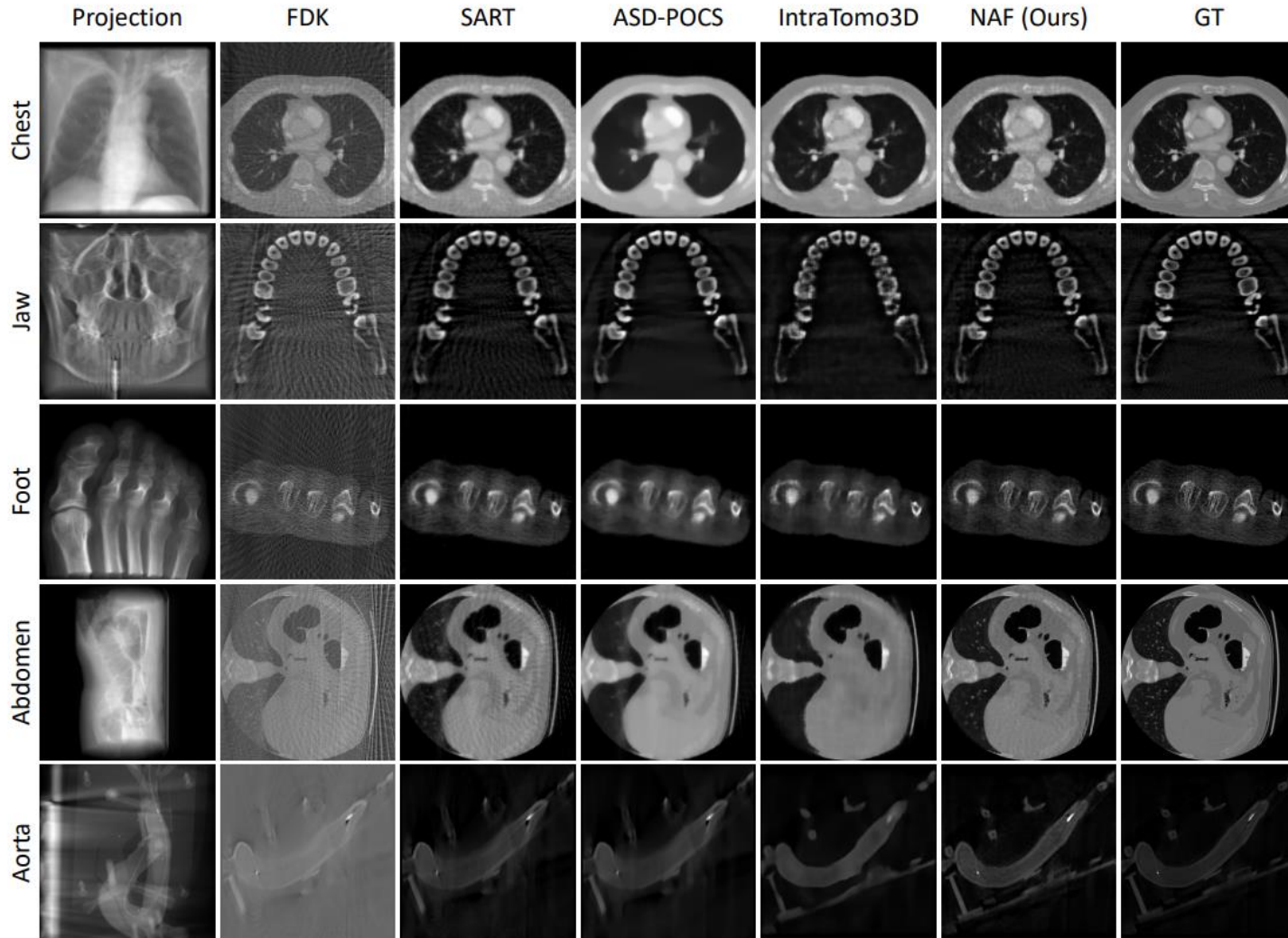
Evaluation Metrics

PSNR : 일반적으로 영상을 압축했을 때 화질이 얼마나 손실되었는지 평가하는 목적으로 사용 (높을 수록 좋음)

SSIM : 영상의 구조 정보를 고려하여 얼마나 구조 정보를 변화시키지 않았는지를 계산 (높을 수록 좋음)

Experiment

❖ Result



FDK : artifact 가 심함

SART : noise가 줄지만 detail을 잃음

ASD-POCS : total-variation regularization 영향으로 high-frequency detail(edge)이 사라짐

IntraTomo3D : frequency encoder가 edge 부분을 집중적으로 배우도록 할 수 없어서 매질사이의 edge부분이 약간 blur

NAF : hash-encoding 덕에 edge detail도 잘 배우고 artifact도 적다고 함

Experiment

❖ Result

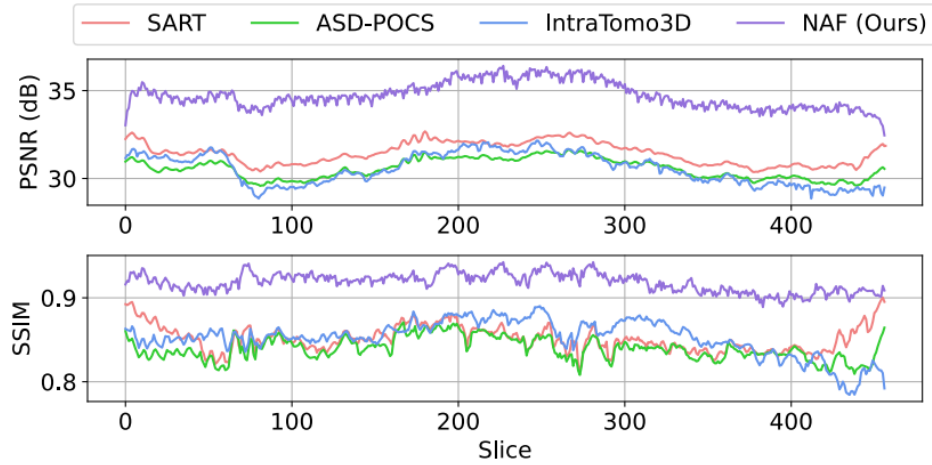


Fig. 3: Slice-wise performance of iterative and learning-based methods on the abdomen dataset.

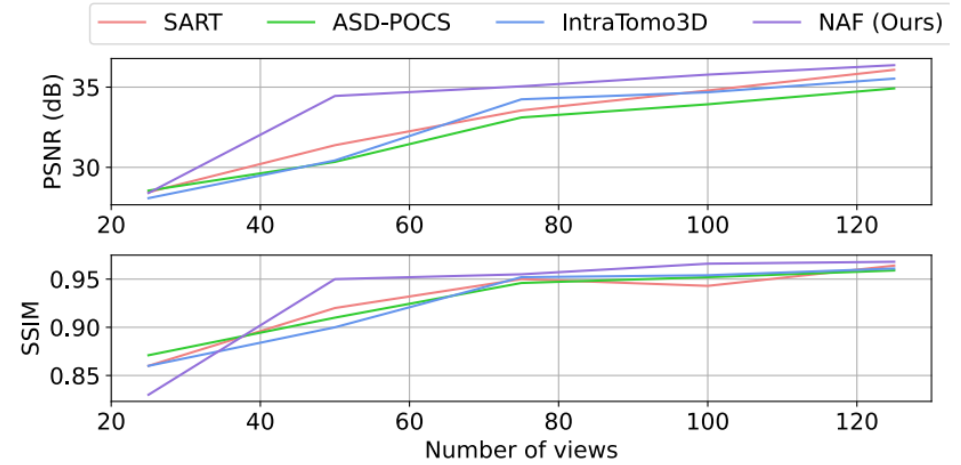


Fig. 4: Performance under different number of views on the abdomen dataset.

Experiment

❖ Result

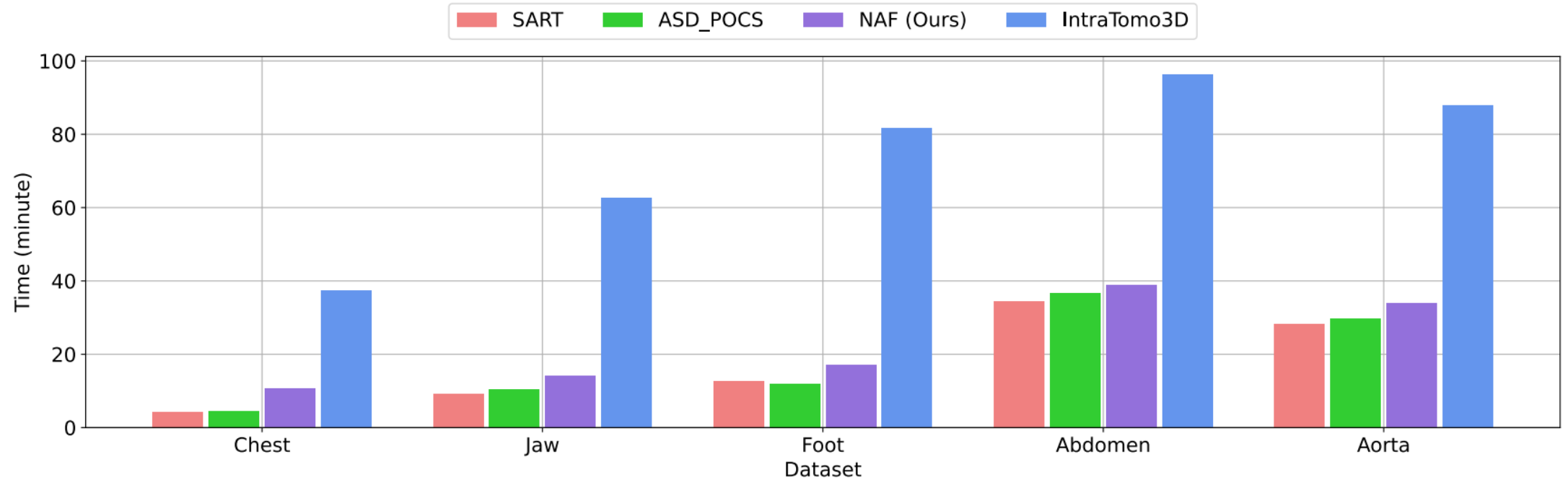


Fig. 5: Running time that iterative and learn-based methods take to converge to stable results.

Conclusion

- Sparse-view CBCT reconstruction을 위한 Fast self-supervised learning-based solution 제안
 - Tomographic reconstruction task에서는 frequency-encoding이 적절하지 않다는 것을 보임
 - Human organ, phantom dataset 실험에서 기존 reconstruction method 보다 확실히 좋은 성능을 보였다고 함
-