Depth-Prior NerF 동향

여름방학 세미나

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

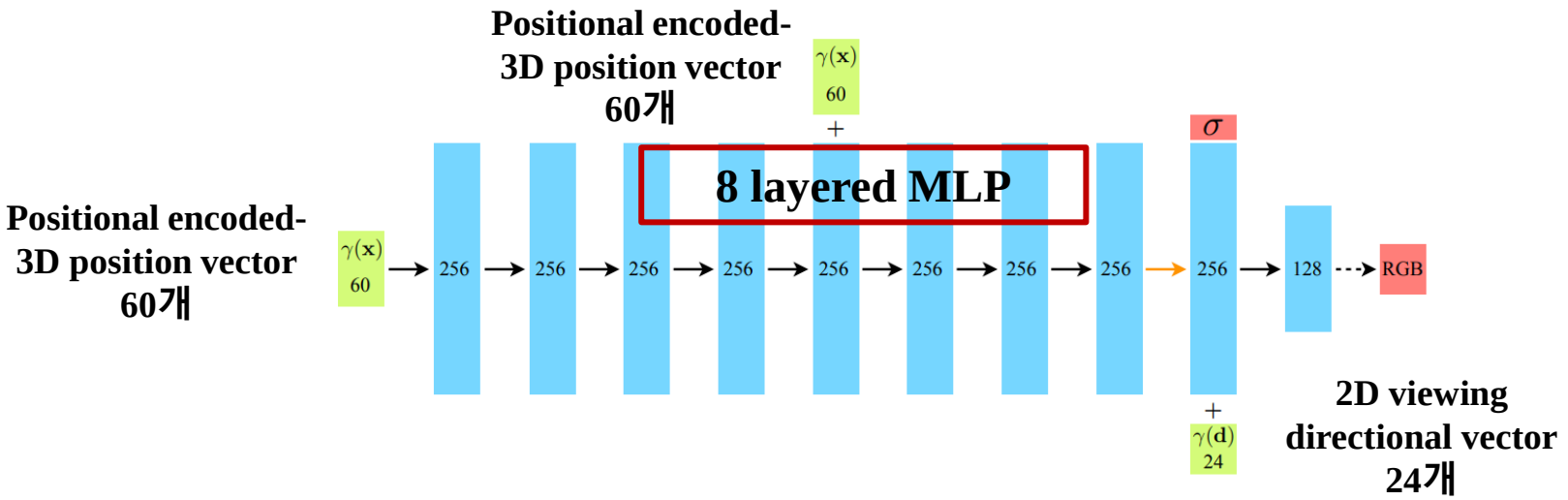
Presented By
Min-jung Shin

목차

- 배경지식
  - NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis
  - NerfingMVS: Guided Optimization of Neural Radiance Fields for Indoor Multi-view Stereo
- 관련 논문 소개
  - 논문1:Dense Depth Priors for Neural Radiance Fields from Sparse Input Views
  - 논문2: DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks

배경 지식 – NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. 2020 ECCV paper



네트워크 및 sampling 특징

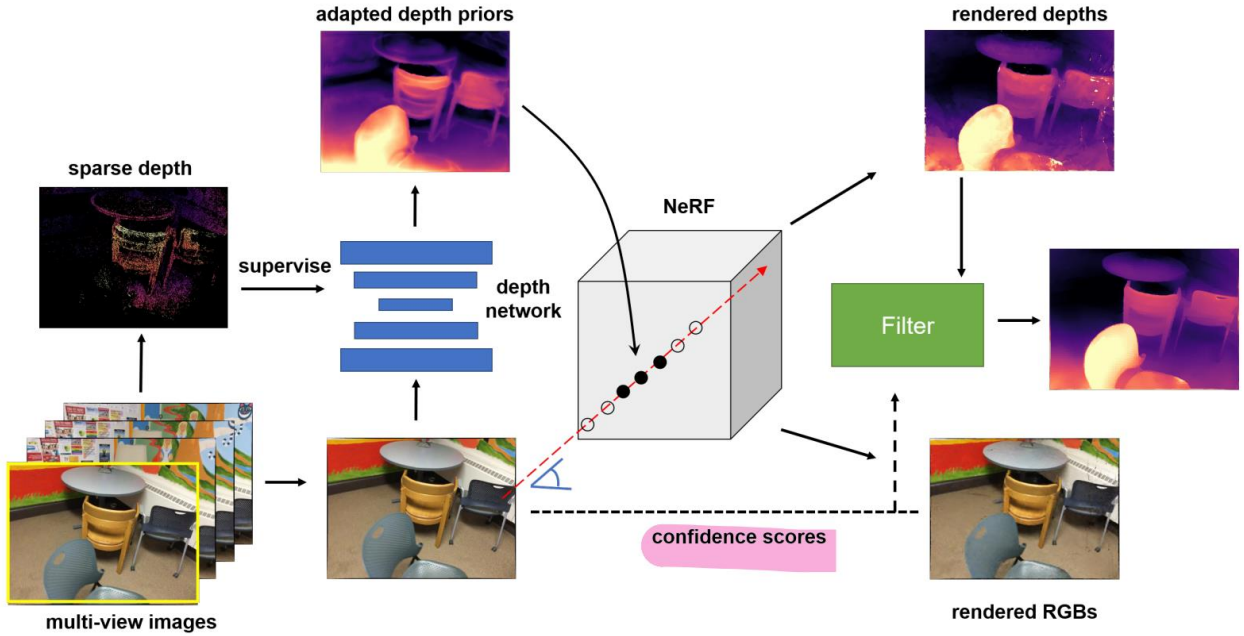
(1) Batch 4096개 각 ray 당 coarse (N_c=64) to Fine (N_F=64+128=192) 으로 총 64+(128+64)=256개 sampling 진행

(2) High frequency 표현을 위해 positional encoding 적용

∴많은 샘플링이 필요해서 높은 연산량& 인퍼런스 타임 필요 →최적 샘플링 위치를 아는 방법은?

배경 지식 – NerfingMVS: Guided Optimization of Neural Radiance Fields for Indoor MVS

NerfingMVS: Guided Optimization of Neural Radiance Fields for Indoor MVS. 2021 ICCV paper



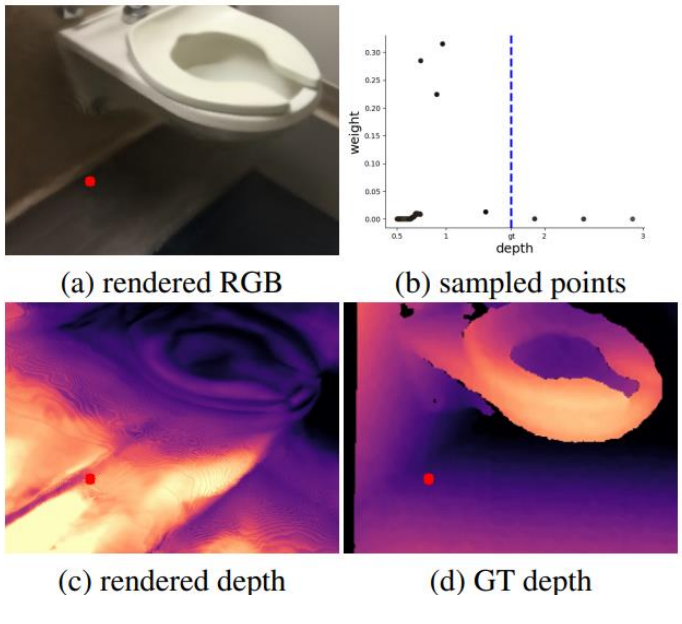
COLMAP sparse depth supervising

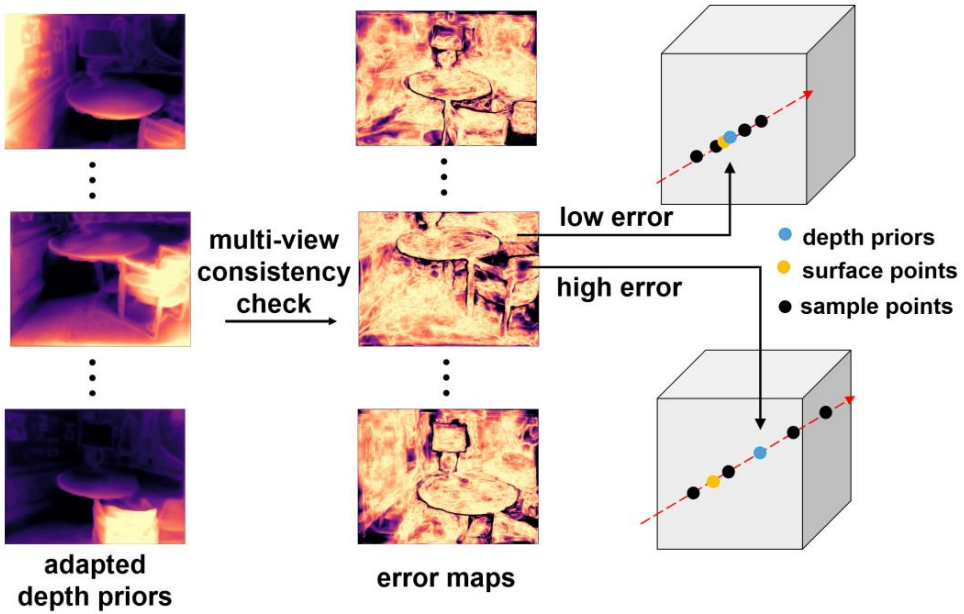
Scale-invariant loss : scale factor $\alpha(D_p^i, D_{Sparse}^i)$ 적용함으로써 scale-invariant finetuning 실시

$$L(D_p^i, D_{Sparse}^i) = \frac{1}{n}\sum_{j=1}^{n} |\log D_p^i(j) - \log D_{Sparse}^i(j) + \alpha(D_p^i, D_{Sparse}^i)|,$$

$$\alpha(D_p^i, D_{Sparse}^i) = \frac{1}{n}\sum_{j}(\log D_p^i(j) - \log D_{Sparse}^i(j))$$

배경 지식 – NerfingMVS: Guided Optimization of Neural Radiance Fields for Indoor MVS





Geometric-consistency check-based error map(e)

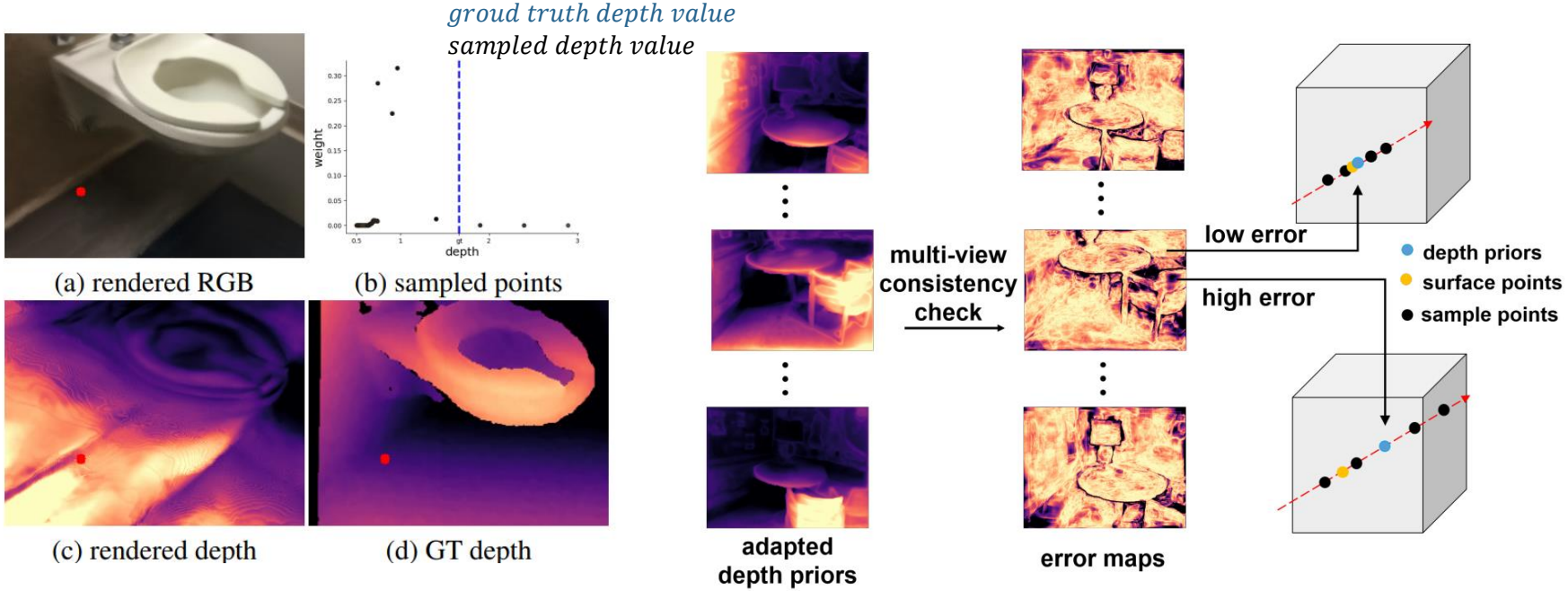
Adaptive depth prior를 이용해 source (j) to target (i) 시점의 reprojection error가 작은 순으로 K개의 projection error 평균을 취하여 error map (e) 생성

$$p^{i\to j}, D^{i\to j} = proj(K, T^{i\to j}, D^i)$$
$$D^{j'} = D^j(p^{i\to j}),$$

Depth prior from N input view $\to \{D^i\}_{i=1}^N$

Error map from each view$\to e_i$

배경 지식 – NerfingMVS: Guided Optimization of Neural Radiance Fields for Indoor MVS



*textureles – region*에서 *deviation*이 큼

NeRF-based uniform sampling

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

$$t_n = D(1 - clamp(e, \alpha_l, \alpha_h))$$
$$t_f = D(1 + clamp(e, \alpha_l, \alpha_h))$$

α_l: Lower bound of range
α_h: higher bound of range

$$C(\mathbf{r}) = \sum_{i=1}^{M} T_i(1 - \exp(-\sigma_i\delta_i))c_i$$

$$D(\mathbf{r}) = \sum_{i=1}^{M} T_i(1 - \exp(-\sigma_i\delta_i))t_i$$

$$T_i = \exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right)$$

배경 지식 – NerfingMVS: Guided Optimization of Neural Radiance Fields for Indoor MVS

NeRF	depth priors	filter	Abs Rel	Sq Rel	$\delta < 1.25$
✓			0.302	0.210	0.518
	✓		0.067	0.010	0.960
✓		✓	0.287	0.167	0.546
	✓	✓	0.065	0.009	0.966
✓	✓		0.053	0.006	0.979
✓	✓	✓	**0.051**	**0.005**	**0.987**

Method	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
COLMAP [43, 44]	0.4619	0.6308	1.0125	1.7345	0.4811	0.5139	0.5333
ACMP [58]	0.1945	0.1710	0.4551	0.3056	0.7309	0.8810	0.9419
DELTAS [45]	0.1001	0.0319	0.2070	0.1284	0.8618	0.9920	0.9991
Atlas [34]	0.0776	0.0631	0.2441	0.2693	0.9289	0.9536	0.9594
DeepV2D [50]	0.0818	0.0226	0.1714	0.1095	0.9414	0.9908	0.9979
NeRF [33]	0.3929	1.4849	1.0901	0.5210	0.4886	0.7318	0.8285
Mannequin [23]	0.1554	0.0636	0.2969	0.1806	0.7859	0.9735	0.9953
CVD [29]	0.0995	0.0304	0.1945	0.1269	0.9008	0.9879	0.9971
Ours w/o filter	0.0635	0.0145	0.1455	0.0936	0.9541	0.9910	0.9989
Ours	**0.0614**	**0.0126**	**0.1345**	**0.0861**	**0.9601**	**0.9955**	**0.9996**

배경 지식 – NerfingMVS: Guided Optimization of Neural Radiance Fields for Indoor MVS

K	α_l	α_h	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
2	0.05	0.15	0.055	0.006	0.083	0.075	0.977	0.998	**1.000**
8	0.05	0.15	0.054	0.006	0.084	0.074	0.979	**0.999**	**1.000**
4	0.01	0.3	0.054	0.007	0.087	0.080	0.971	0.997	**1.000**
4	0.05	0.3	0.055	0.007	0.087	0.079	0.976	0.998	**1.000**
4	0.01	0.15	0.053	0.006	0.083	0.075	0.980	0.998	**1.000**
4	0.05	0.15	**0.051**	**0.005**	**0.076**	**0.069**	**0.987**	0.998	**1.000**

한계 및 고찰 사항

1) Batch 1024개의 ray당 64개 sample 학습 → 기존 NeRF 에 비해 3배 빨라짐
2) Coarse to fine 네트워크 사용하지 않고, MLP 1번만 NeRF와 동일한 구조로 사용
3) Scene마다 적절 K, α_l, α_h 값을 조절 하는 것이 중요
4) 일부 geometric structure를 잘 반영한 scene 에 대해서만 inference를 실시해 PSNR이 높을 가능성이 있음 ex) 8개 scene 평균 PSNR : 31.55

∴ 여전히 속도 측면에서 한계가 존재하고, 전반적으로 geometric structure 반영이 약함

2022년 Depth Prior-NeRF 동향

- **Dense Depth Priors for Neural Radiance Fields from Sparse Input View. CVPR 2022**
- **DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks. Eurographics Symposium on Rendering (EGSR) 2022**

Dense Depth Priors for NeRF estimates depth using a depth completion network run on the SfM point cloud in order to constrain NeRF optimization, yielding higher image quality on scenes with sparse input images. (pdf)

Depth-supervised NeRF also uses a depth completion network on structure-from-motion point clouds to impose a depth-supervised loss for faster training time on fewer views of a given scene. (pdf)

InfoNeRF penalizes the NeRF overfitting ray densities on scenes with limited input views through ray entropy regularization, resulting in higher quality depth maps when rendering novel views. (pdf)

RapNeRF focuses on view-consistency to enable view extrapolation, using two new techniques: random ray casting and a ray atlas. (pdf)

RegNeRF enables good reconstructions from a view images by renders patches in *unseen* views and minimizing an appearance and depth smoothness prior there. (pdf)

GeoNeRF uses feature-pyramid networks and homography warping to construct cascaded cost volumes on input views that infer local geometry and appearance on novel views, using a transformer-based approach. (pdf)

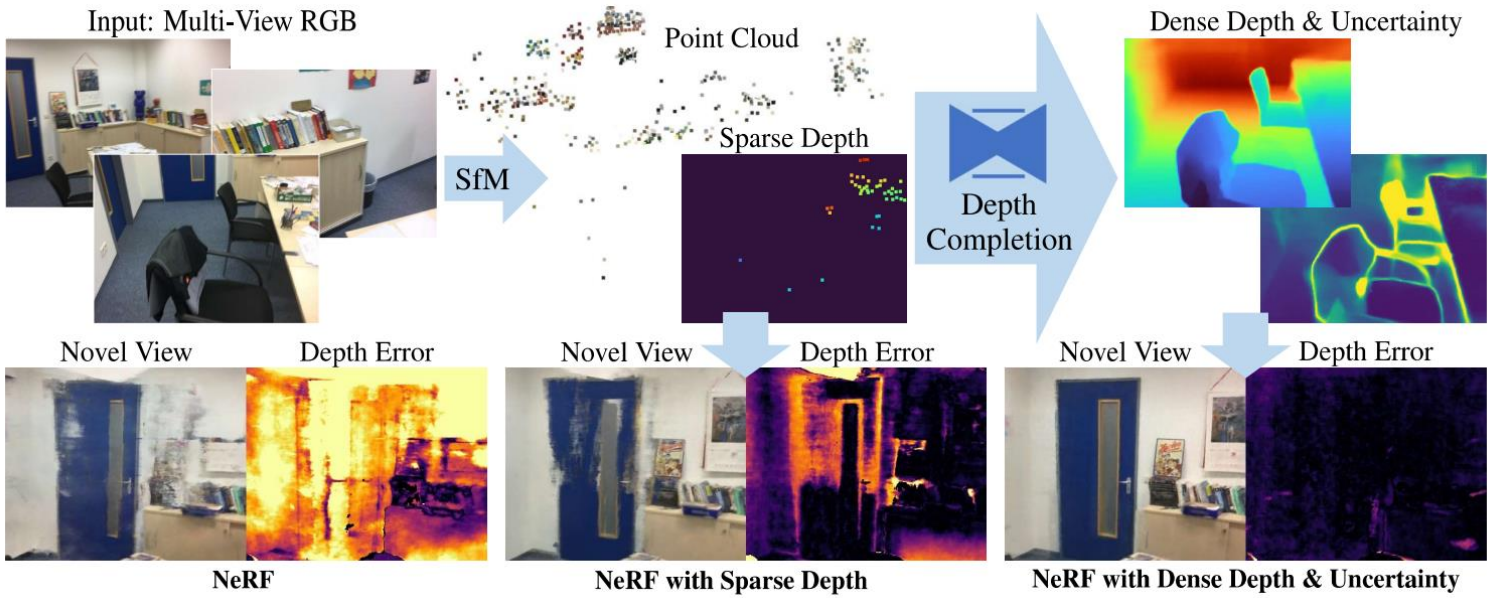
Light Field Neural Rendering uses a lightfield parameterization for target pixel and its epipolar segments in nearby reference views, to produce high-quality renderings using a novel transformer architecture. (pdf) **Best Paper Finalist**

NAN builds upon IBRNet and NeRF to implement burst-denoising, now the standard way of coping with low-light imaging conditions. (pdf)

NeRFusion first reconstructs local feature volumes for each view, using neighboring views, and then uses recurrent processing to construct a global neural volume. (pdf)

- 관련 링크: https://dellaert.github.io/NeRF22/?fbclid=IwAR3aIWYQXbegQpQajLAKqRbmy3mGQZcdgJnHAl6b6dWwawZydgHStvptg&fs=e&s=cl

Dense Depth Priors for Neural Radiance Fields from Sparse Input Views

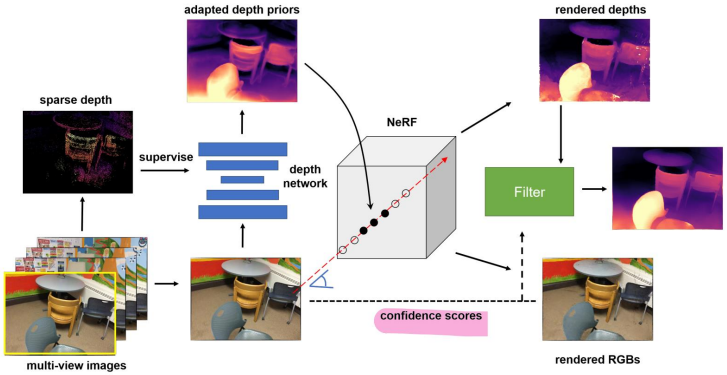


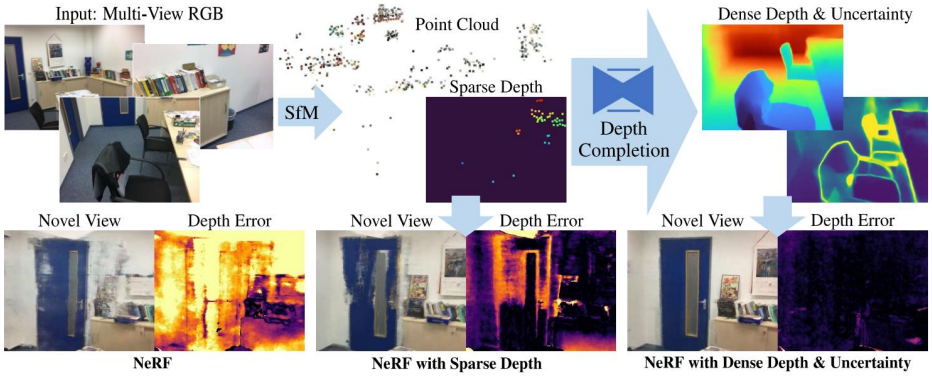
SfM 기반 depth prior의 한계 및 개선

1. Sparse depth는 noisy outlier가 있어 dense depth map 추출 시 시점당 정확도가 다름
   - 픽셀별 uncertainty를 아는 것이 중요 → ResNet을 이용하여 dense depth(Z_i^{dense})및 pixel-wise standard deviation (S_i) 추출
2. Point cloud의 density가 시점별로 다름
   - Convolutional Spatial Propagation Network (CSPN)이용 → iteration을 높일수록 sparse density 정보를 잘 처리 → dense depth의 디테일이 더 살아있음

$$[Z_i^{\text{dense}}, S_i] = D_{\theta_0}(I_i, Z_i^{\text{sparse}})$$

Dense Depth Priors for Neural Radiance Fields from Sparse Input Views



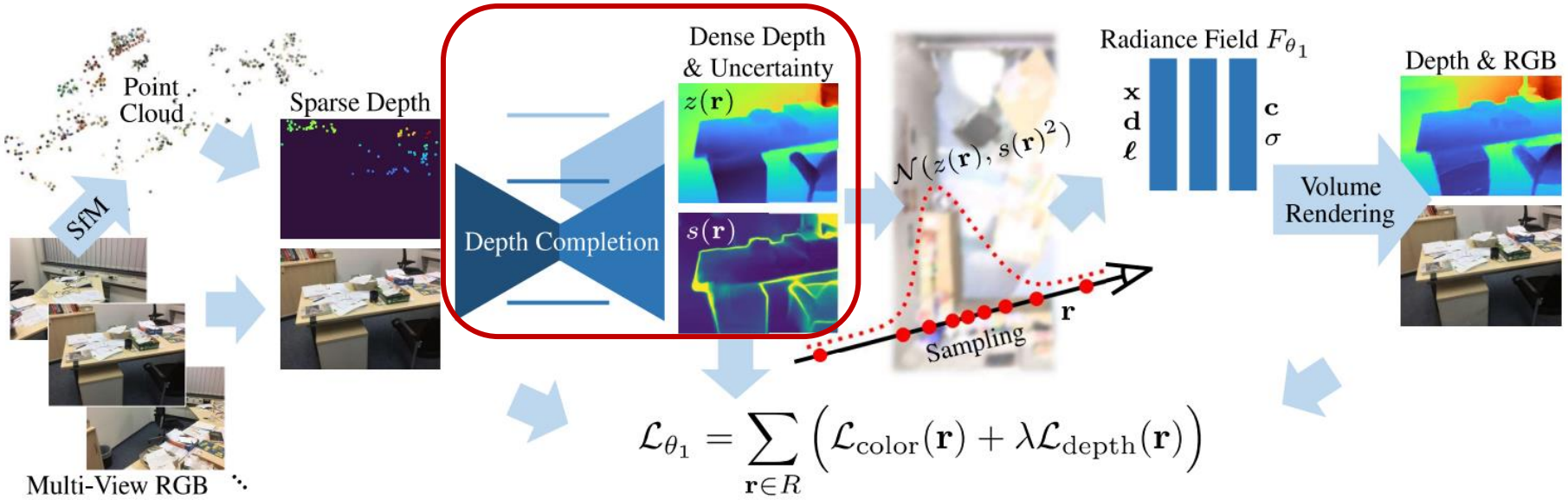


NerfingMVS와 학습 및 테스트 방식 차이

1. NerfingMVS는 training 및 test 모두 COLMAP 에서 추출한 sparse depth를 사용하여 test 시에도 sampling 범위를 finetuning하며 진행
   - PSNR 자체는 scene에 따라 높을 수 있음

2. 반면, 해당 모델은 training에만 sparse depth를 사용하고, test 에는 COLMAP SfM sampling을 진행하지 않음
   - PSNR 자체는 scene에 따라 낮을 수 있음

Dense Depth Priors for Neural Radiance Fields from Sparse Input Views



Depth completion network 학습

입력 sparse depth map이 sensor depth(GT depth) scale 범위에 맞추어서 학습할 필요 존재

→ SfM 에서 추출된 특징점과 동일 위치에서 GT depth sampling → Gaussian noise 부여

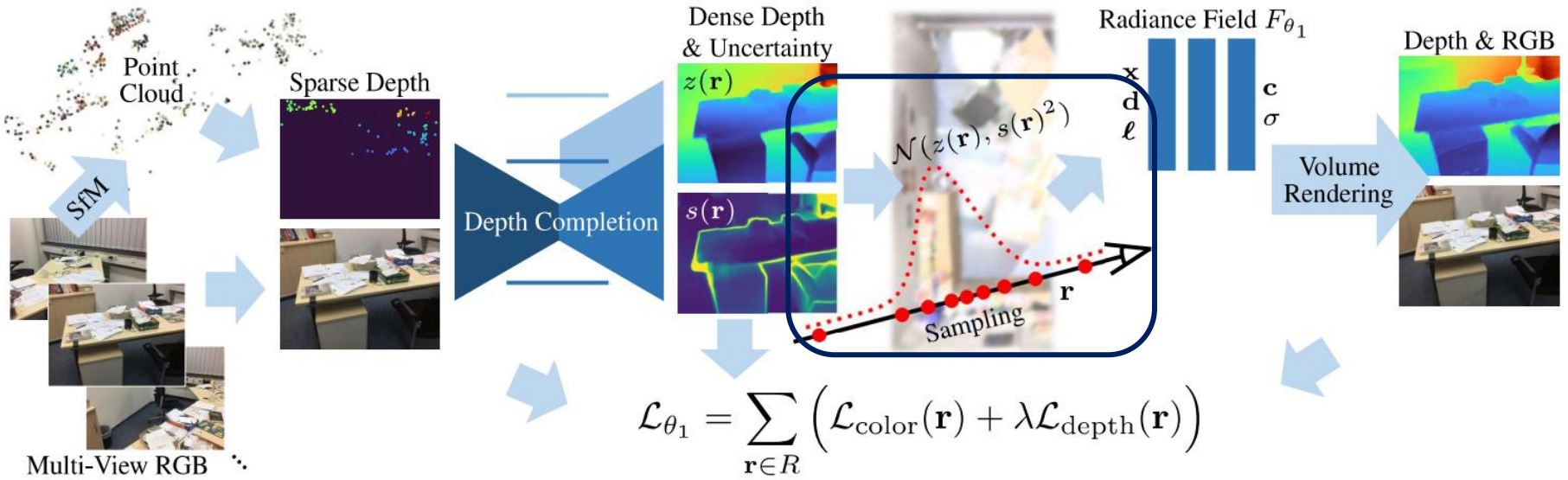
Negative-log likelihood of gaussian loss 학습

$$\mathcal{L}_{\theta_0} = \frac{1}{n} \sum_{j=1}^{n} \left(\log(s_j^2) + \frac{(z_j - z_{\text{sensor},j})^2}{s_j^2} \right)$$

$$\mathcal{N}(0, s_{\text{noise}}(z)^2)$$

Z_j: predicted depth of pixel j
S_j: standard deviation of pixel j
Z_{sensor}: sensor depth at j
n: number of valid pixels

Dense Depth Priors for Neural Radiance Fields from Sparse Input Views



Sampling 기법 소개 (1)

Depth prior값을 통해 생성된 rendering weight를 반영해서 새로운 $\hat{z}(r)$, $\hat{s}(r)$ 생성 후 sampling

$$\hat{z}(\mathbf{r}) = \sum_{k=1}^{K} w_k t_k \,, \quad \hat{s}(\mathbf{r})^2 = \sum_{k=1}^{K} w_k (t_k - \hat{z}(\mathbf{r}))^2$$

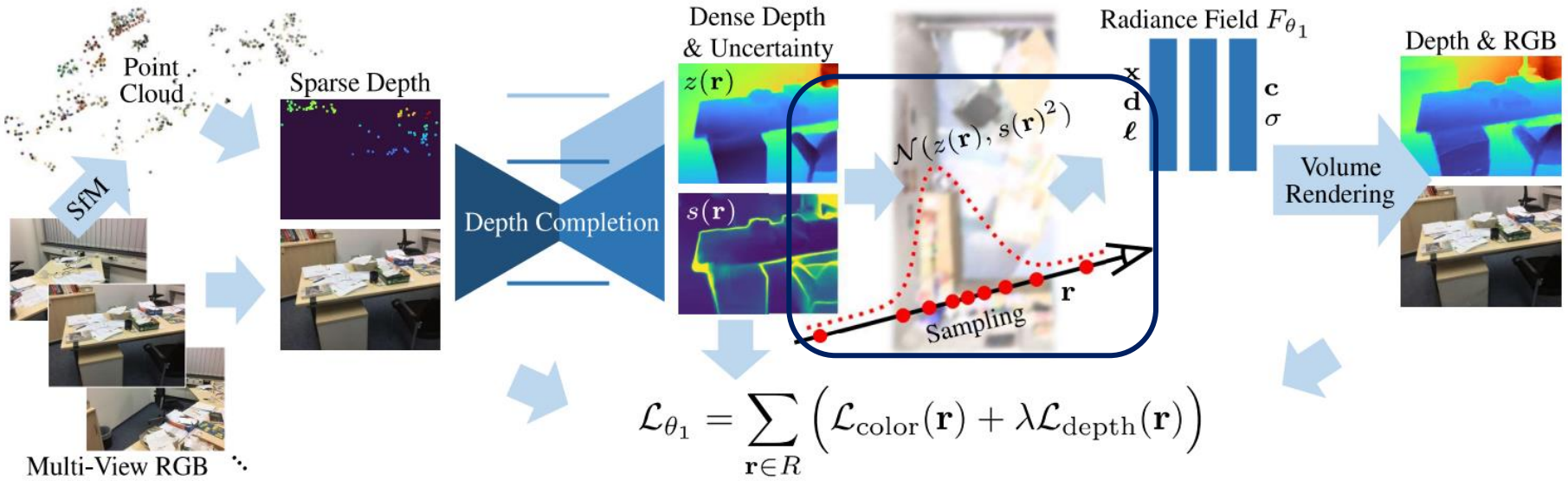
$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{k=1}^{K} w_k \mathbf{c}_k \,,$$

where $w_k = T_k \left(1 - \exp(-\sigma_k \delta_k)\right)$

$$T_k = \exp\left(- \sum_{k'=1}^{k} \sigma_{k'} \delta_{k'} \right)$$

$$\delta_k = t_{k+1} - t_k \,.$$

Dense Depth Priors for Neural Radiance Fields from Sparse Input Views



Sampling 기법 소개 (2)

1) Training: sample들의 절반은 uniform sampling , 나머지 절반은 gaussian sampling
→ depth prior 기반의 gaussian sampling : $\mathcal{N}(z(\mathbf{r}), s(\mathbf{r})^2)$

2) Testing: predicted depth 기반의 gaussian sampling : $\mathcal{N}(\hat{z}(\mathbf{r}), \hat{s}(\mathbf{r})^2)$

Dense Depth Priors for Neural Radiance Fields from Sparse Input Views

Total Network 학습 방법

$$\mathcal{L}_{\theta_1} = \sum_{\mathbf{r}\in R}\Big(\mathcal{L}_{\text{color}}(\mathbf{r}) + \lambda\mathcal{L}_{\text{depth}}(\mathbf{r})\Big)$$

MLE of Gaussian negative log-liklihood이란?

$$\begin{aligned}\mathcal{LL} &= \sum_{n=1}^{N}\Big(-\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2}\Big(\frac{(x_n-\mu)^2}{\sigma^2}\Big)\Big)\\ &= -\frac{N}{2}\log(2\pi\sigma^2) + \sum_{n=1}^{N} -\frac{1}{2}\Big(\frac{(x_n-\mu)^2}{\sigma^2}\Big)\\ &= -\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2\end{aligned}$$

Color term: Mean Square Error (MSE)

$$\mathcal{L}_{\text{color}}(\mathbf{r}) = \left\|\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})\right\|_2^2,$$

Depth term: Gaussian negative log-likelihood

$$\mathcal{L}_{\text{depth}}(\mathbf{r}) = \begin{cases}\log\left(\hat{s}(\mathbf{r})^2\right) + \frac{(\hat{z}(\mathbf{r})-z(\mathbf{r}))^2}{\hat{s}(\mathbf{r})^2} & \text{if } P \text{ or } Q\\ 0 & \text{otherwise,}\end{cases}$$

where $P = |\hat{z}(\mathbf{r}) - z(\mathbf{r})| > s(\mathbf{r})\,,$

$Q = \hat{s}(\mathbf{r}) > s(\mathbf{r})\,.$

Depth term: case1 or 2 둘 중 하나만 만족 시,

Case1: predicted depth- target depth > standard deviation

Case2: predicted standard deviation> standard deviation

Dense Depth Priors for Neural Radiance Fields from Sparse Input Views

네트워크 특징 및 한계

1) Batch 1024개의 ray 당 256개의 sampling 진행

2) 기존 NeRF (4096개 ray당 256개 sampling)에 비해 연산량은 줄어들었지만,
NerfingMVS (1024개 ray당 64개 sampling)과 비교하면 연산량 감소는 이루지 못하였음

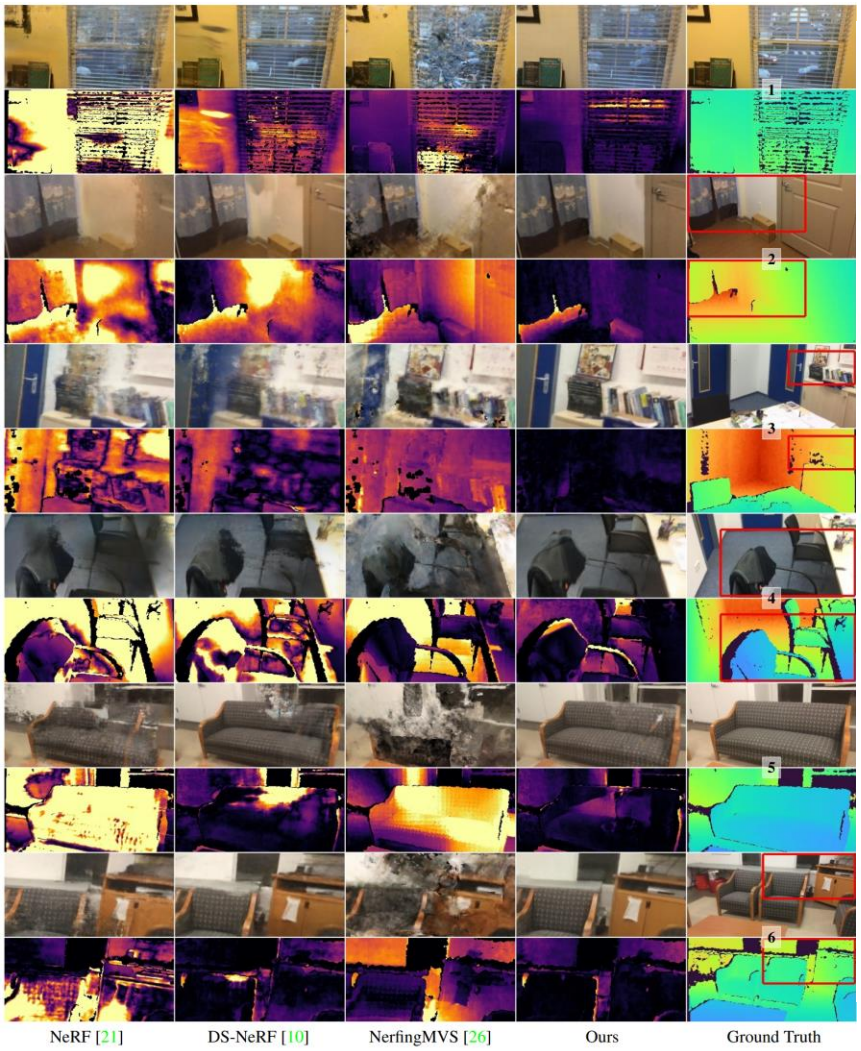
3) Coarse to fine 구조는 사용하지 않고, 한번의 MLP 연산

네트워크 의의

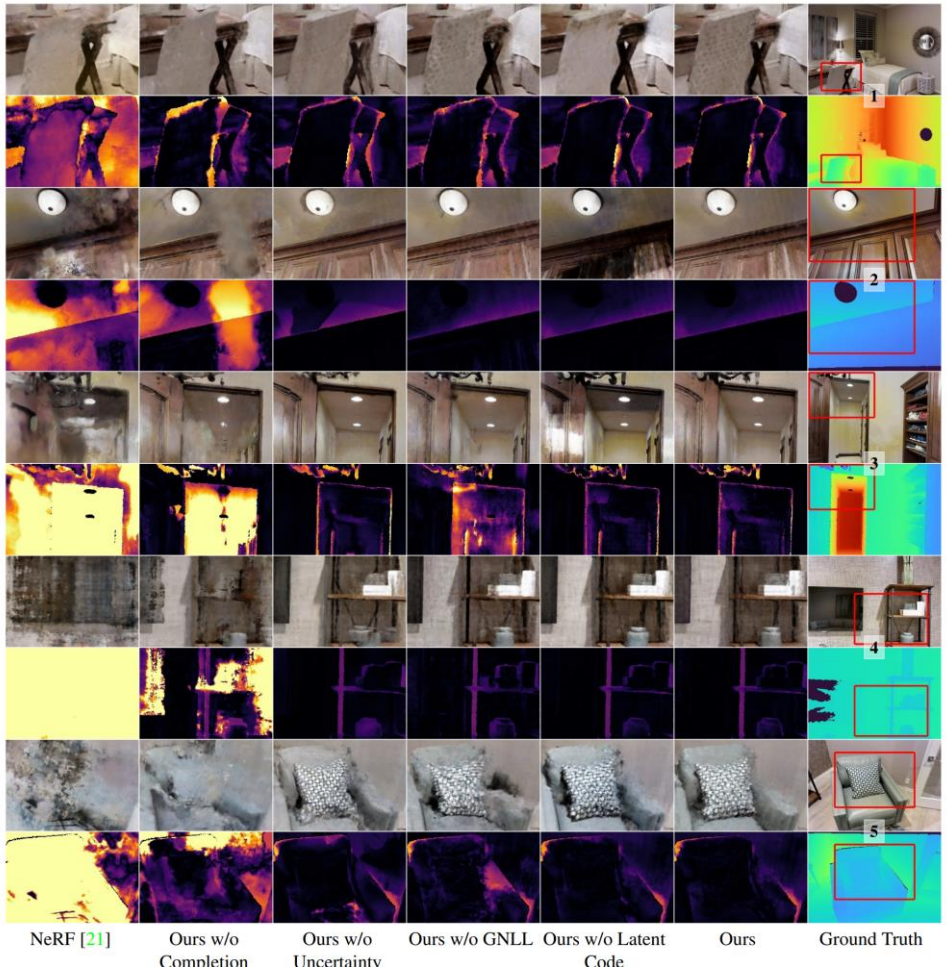
1) 하지만 적은 양의(18-20 train, 8 test) 데이터로 room scaled-rendering이 가능함
→ NeRF는 room-scale dataset이 아닌데도 최소 25장- 100장 필요
→ NerfingMVS는 해당 부분에서 취약한 한계가 있었음
a. 모든 scene의 COLMAP sparse input 필요
b. Error map을 모든 scene에 대해서 구한 후 평균을 취했다는 점에서 depth range 오류 존재

2) Inference 시에 COLMAP sparse input 필요 없음

Dense Depth Priors for Neural Radiance Fields from Sparse Input Views



ScanNet 결과



Matterport3D 결과

Dense Depth Priors for Neural Radiance Fields from Sparse Input Views

Method	scene 0616		scene 0521		scene 0000		scene 0158	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
NSVF [25]	15.71	0.704	27.73	0.892	**23.36**	0.823	**31.98**	**0.951**
SVS [40]	**21.38**	**0.899**	**27.97**	**0.924**	21.39	**0.914**	29.43	**0.953**
NeRF [33]	15.76	0.699	24.41	0.871	18.75	0.751	29.19	0.928
Ours	18.07	0.748	**28.07**	0.901	22.10	0.880	30.55	**0.948**
Method	scene 0316		scene 0553		scene 0653		scene 0079	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
NSVF [25]	**22.29**	0.917	31.15	0.947	28.95	0.929	26.88	0.887
SVS [40]	20.63	**0.941**	30.95	**0.968**	27.91	**0.965**	25.18	**0.923**
NeRF [33]	17.09	0.828	30.76	0.950	30.89	0.953	25.48	0.896
Ours	20.88	0.899	**32.56**	**0.965**	**31.43**	**0.964**	**27.27**	**0.916**

Method	PSNR ↑	SSIM ↑	LPIPS ↓
NeRF [33]	28.62	0.909	0.319
Ours	**31.55**	**0.942**	**0.200**

NerfingMVS: scene 8개 정량적 평가

NerfingMVS vs 제안 기법 비교

데이터 셋은 모두 ScanNet으로 진행

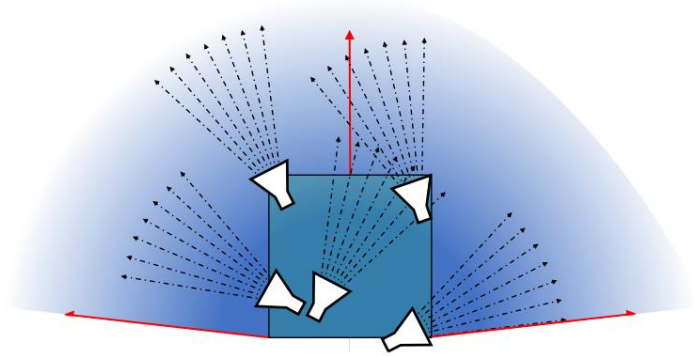
(1) Scene에 따라 정량적 결과 차이 존재

(2) 제안기법이 room 전체 reconstruction 가능

Method	PSNR↑	SSIM↑	LPIPS↓	Depth RMSE ↓
NeRF [21]	19.03	0.670	0.398	1.163
DS-NeRF [10]	20.85	0.713	0.344	0.447
NerfingMVS [26]	16.29	0.626	0.502	0.482
Ours w/o Completion	20.43 (22.10)	0.707	0.366	0.526
Ours w/o Uncertainty	20.09 (22.21)	0.714	0.308	0.279
Ours w/o GNLL	20.80 (22.23)	0.733	0.312	0.275
Ours w/o Latent Code	20.87	0.726	**0.293**	0.243
Ours	**20.96** (**22.30**)	**0.737**	0.294	**0.236**

제안 기법: scene 3개 정량적 평가

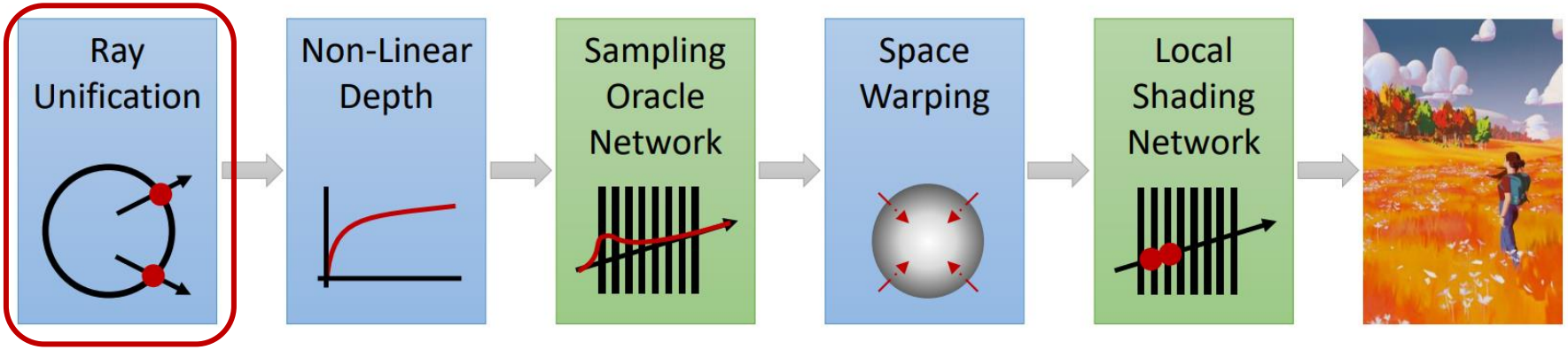
DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks



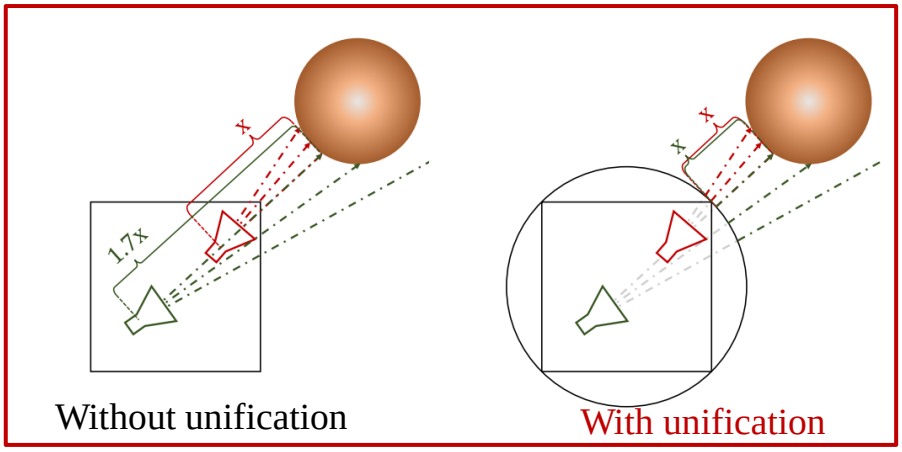
View cells: 특정 bounding box 안에서 특정 각도 내의 모든 view ray를 포집할 수 있는 범위 의미

Why? 깊은 범위의 depth range를 갖는 데이터 셋에 대해 가상시점 합성을 하기 위해서

DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks

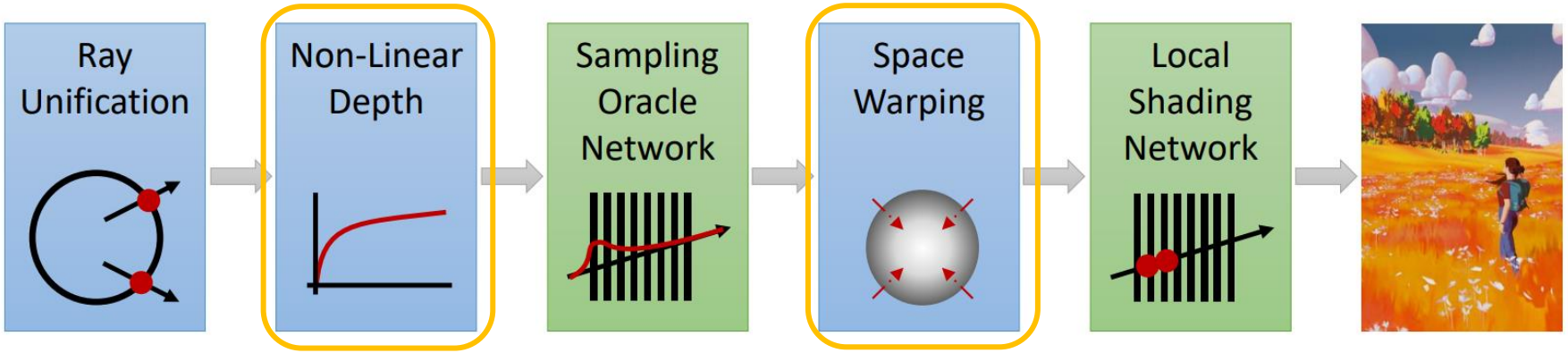


Ray unification: 같은 ray 선상의 정보는 같은 depth로 표현할 수 있도록 조절



View cell의 외부에서 다각도로 바라 보았을 때, 같은ray 선상 정보는 같은 depth가 될 수 있도록 view cell 외부에 외접 sphere 설치

DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks



Non-linear depth sampling

1) Uniform sampling : 균일한 간격으로 depth sampling 진행

$$\mathbf{x}(d_i) = \mathbf{o} + d_i \cdot \mathbf{r}$$

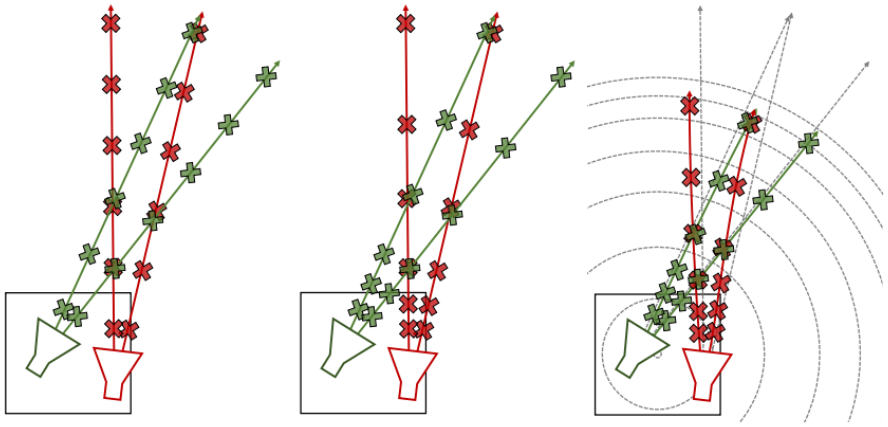
$$d_i = \left(d_{min} + i \cdot \frac{(d_{max} - d_{min})}{N} \right), \; i = [0, 1, 2, \cdots, N],$$

2) Non-linear sampling (logarithmic): 가까이 있는 건 dense, 멀리 있는 건 sparse하게 log sampling

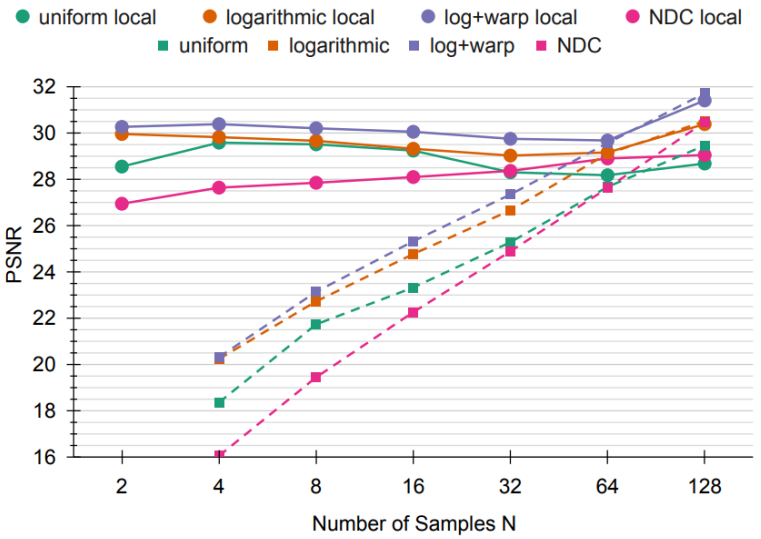
$$\tilde{d}_i = d_{min} + \frac{\log(d_i - d_{min} + 1)}{\log(d_{max} - d_{min} + 1)} \cdot (d_{max} - d_{min})$$

$$\mathbf{f}(d_i) = \text{encode}\left(\frac{\mathbf{x}(d_i) - \mathbf{c}}{d_{max}} \right).$$

DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks



(a) *uniform* **(b)** *logarithmic* **(c)** *log+warp*



Non-linear depth sampling

3) NDC(Normalized device coordinate) sampling

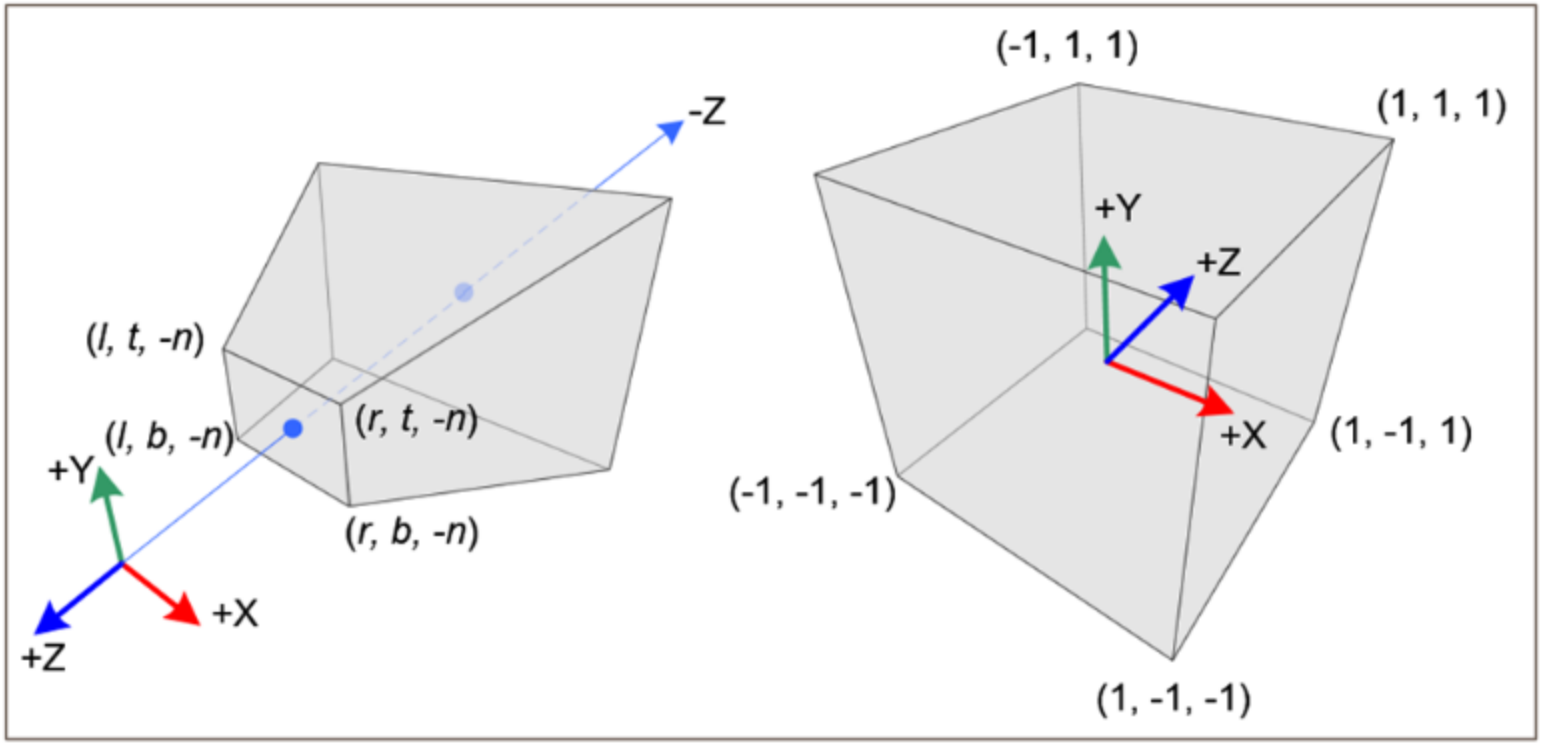
Affine transform

기존 NeRF에서 forward-facing 실사 데이터 셋에 적용하기위해 **카메라 기준 좌표** → **기준 큐브 좌표** $[-1, 1]^3$

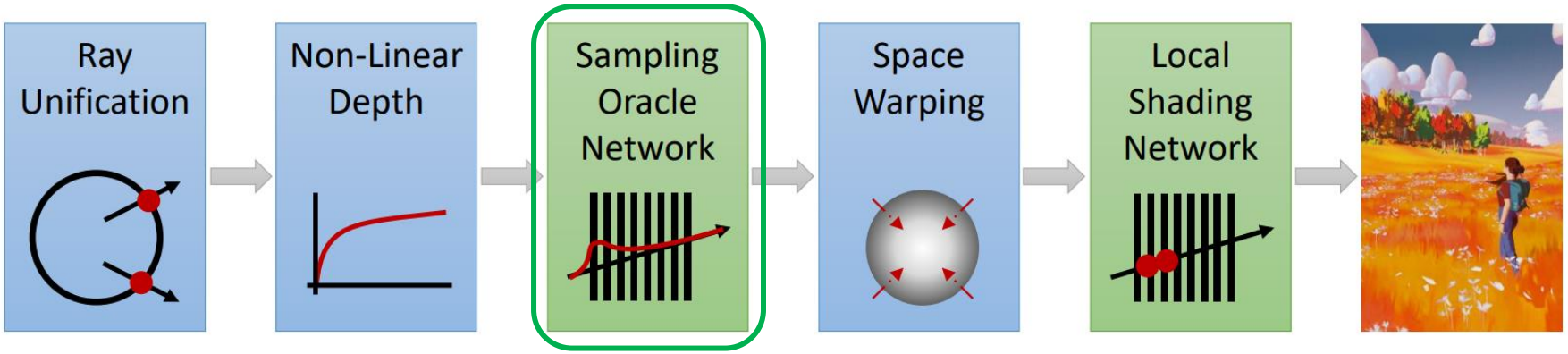
4) Space Warping (log+warp) : 가까이 있는 건 dense, 멀리 있는 건 sparse하게 log-space sampling 되어 foreground 에서만 표현이 잘되는 한계를 개선하기 위해 radial distorted- warping 적용

$$\tilde{\mathbf{f}} = \text{encode}\left((\mathbf{x}(\tilde{d}_i) - \mathbf{c}) \cdot W(\mathbf{x}(\tilde{d}_i) - \mathbf{c})\right) \qquad W(\mathbf{p}) = \frac{1}{\sqrt{|\mathbf{p}| \cdot d_{max}}}$$

NDC 참고자료



DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks



Sampling Oracle Network : Depth를 sampling prior로 단순히 사용하는 대신, classification depth oracle 네트워크 생성

기존 prior 방식은 surface representation에 의존 → compact 한 representation 사용 의미 무색

→각 ray에 확률적으로 객체가 존재 할 만한 최적의 sampling 위치는?

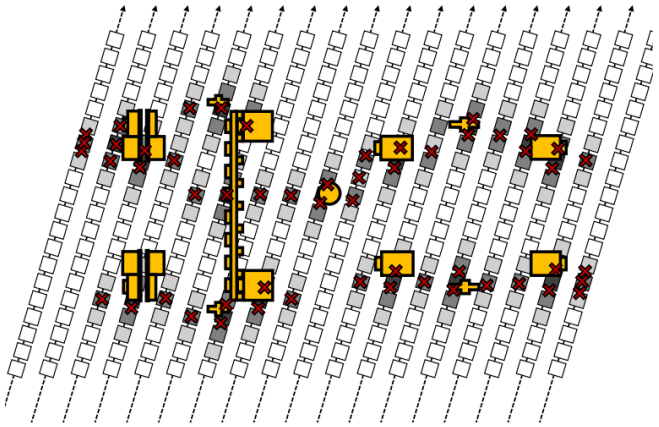
↓

Classification network 도입

↓

Foreground와 background 사이에 depth discontinuity가 발생하기 때문에, 최적의 위치를 찾는 것은 쉽지 않음 → 최적 position에 여유 oracle 부여 → **여러 ray label을 classification으로 부여**

DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks



Depth Oracle 이란?

기존 NeRF의 coarse network의 sampling의 느낌으로 , 임의로 생성한 depth 단위의 particle들

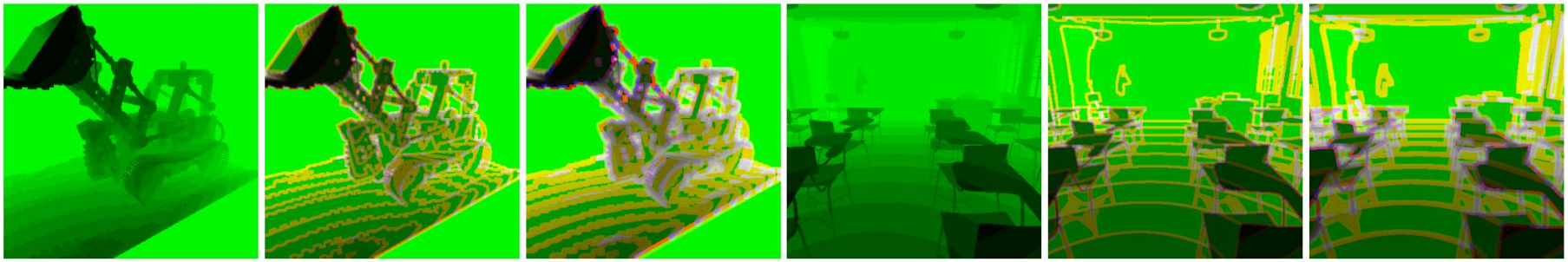
→ 최적의 sample 장소들을 뽑기 위해 sample보다 **큰 단위의** depth particle을 생성

→ Classification을 통해 유의미한 depth추출 + depth scale의 일관성 확보

→ 각각의 depth는 non-real valued depth

실제 depth 값을 이용하는 것보다, scale 일관성을 갖는 임의의 depth label 값이 geometric 보장

DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks



(a) Bulldozer $K = 1$ **(b)** Bulldozer $K = 5$ **(c)** Bulldozer $K = 9$ **(d)** Classroom $K = 1$ **(e)** Classroom $K = 5$ **(f)** Classroom $K = 9$

Depth Ray Classification 순서: binary 분류 → multi-class 분류 →1D label smoothing

$$C_{x,y}(z) = \begin{cases} 1, & \text{if } d_z \leq d_s < d_{z+1} \\ 0, & \text{otherwise,} \end{cases}$$

기대 확률이 높을 것 같은 ray segment 범위 [dz,dz+1]에 특정 GT depth (d_s)가 포함되어 있다면 1, 아니면 0으로 표시
→**Binary depth ray생성**

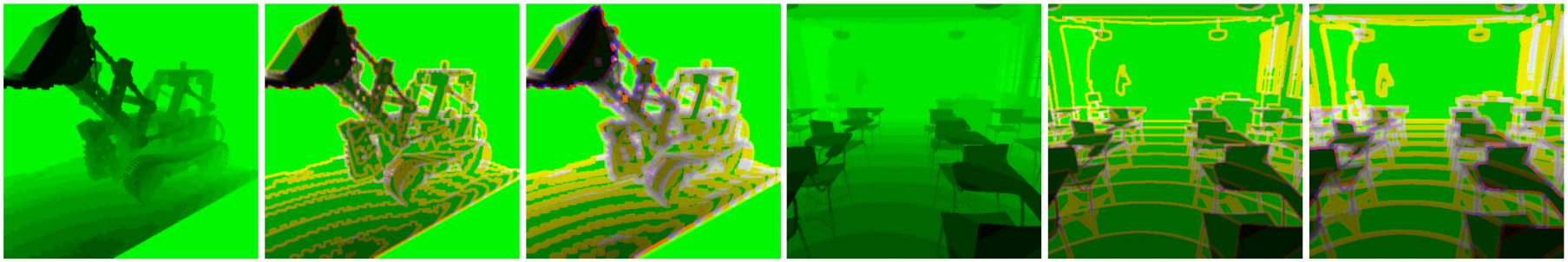
$$\hat{C}_{x,y}(z) = \max_{i,j \in \pm \lfloor K/2 \rfloor} \left(C_{x+i,y+j}(z) - \frac{\sqrt{i^2 + j^2}}{\sqrt{2} \cdot \lfloor K/2 \rfloor} \right)$$

좀더 scale-consistent한 결과를 도출하기 위해 인접 ray의 거리ex)0,1,2,3에 따라 lower contribution 을 갖는 filter도입
→**Multi-class depth ray 생성 ex)4x4=16class**

$$\grave{C}(z) = \min \left(\sum_{i=-\lfloor Z/2 \rfloor}^{\lfloor Z/2 \rfloor} \hat{C}(z+i) \frac{\lfloor Z/2 \rfloor + 1 - |i|}{\lfloor Z/2 \rfloor + 1}, 1 \right)$$

Depth discontinuity에서 hard boundary 피하도록 Label smoothing filter 적용

DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks



(a) Bulldozer $K = 1$ **(b)** Bulldozer $K = 5$ **(c)** Bulldozer $K = 9$ **(d)** Classroom $K = 1$ **(e)** Classroom $K = 5$ **(f)** Classroom $K = 9$

초록색: Single-valued depth sampled ray with depth, 회색: multiple-valued depth but similar, 다른색: multiple depth and different

Multi-class ray classification이 의미

(1) 높은 class 픽셀일 수록 가깝고, 많은 ray oracle 을 받음

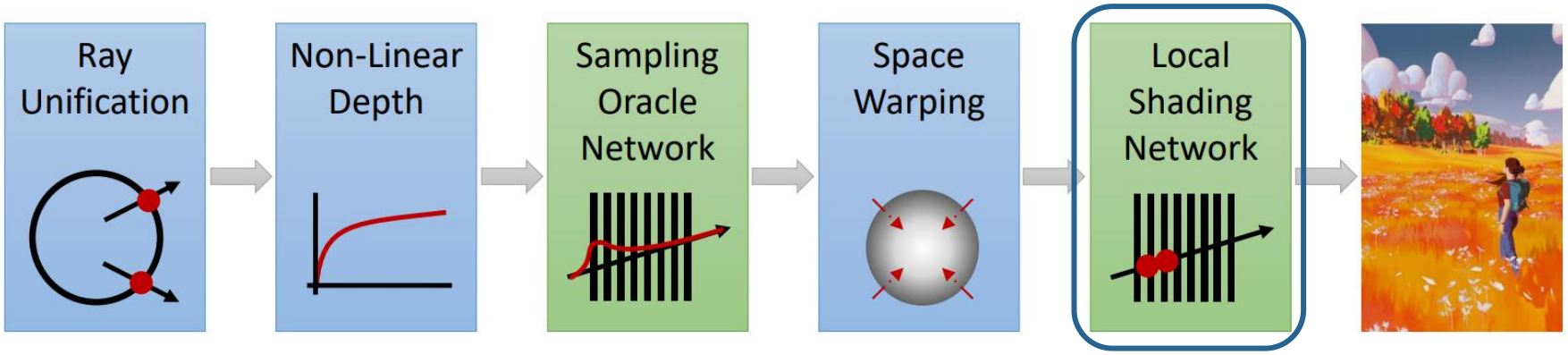
(2) 반면, 낮은 class 일수록 멀고, ray oracle 을 거의 받지 못해 정보 skip 가능

(3) 각 oracle의 depth가 비슷하면 회색, 다르면 이상한 색

(4) 많은 free oracle (empty oracle)을 부여함으로써 최적 공간 sampling 확률 높임

- Binary classification을 통해 False negative label이 전부 사라졌고, False Positive label의 비율이 상승함으로 써 free sampling이 많이 포함

DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks



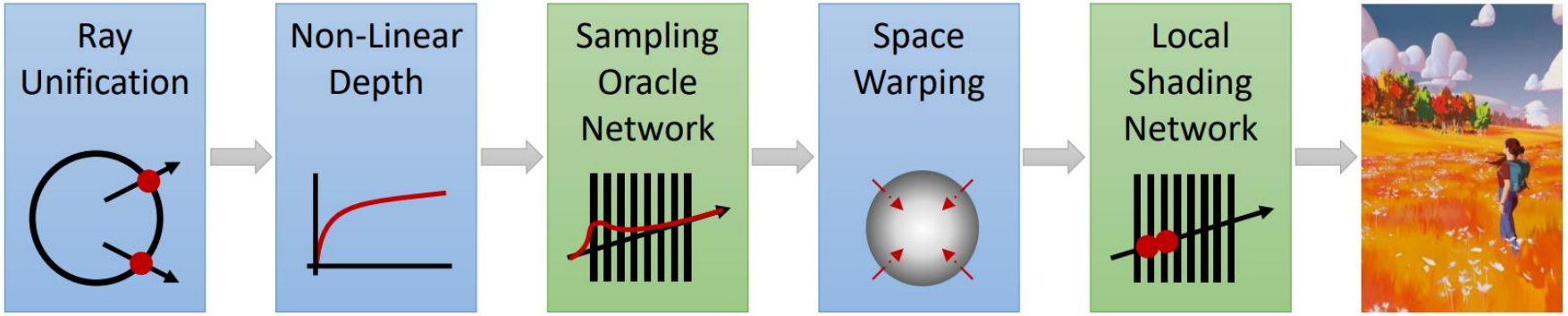
Local shading Network : NeRF의 fine network 대신에, depth oracle의 3d position을 MLP에 넣어줌

	PSNR ↑				FLIP ↓			
Method \ N	2	4	8	16	2	4	8	16
SD	26.686	27.401	28.220	29.085	0.092	0.084	0.078	0.073
SD unified	27.423	28.052	28.825	29.554	0.085	0.079	0.074	0.071
K-1 Z-1 I-1	27.325	28.697	30.068	31.145	0.082	0.073	0.065	0.061
K-5 Z-1 I-1	28.685	30.521	31.988	32.982	0.075	0.066	0.059	0.055
K-5 Z-1 I-128	29.956	31.746	32.951	33.760	0.067	**0.061**	**0.056**	**0.053**
K-5 Z-5 I-128	**30.071**	**31.842**	33.027	33.836	**0.067**	**0.061**	**0.056**	**0.053**
K-9 Z-1 I-1	28.831	30.881	32.617	33.495	0.075	0.065	0.057	0.055
K-9 Z-1 I-128	29.299	31.645	**33.125**	**33.994**	0.071	**0.061**	**0.056**	**0.053**
K-9 Z-9 I-128	28.737	30.847	32.261	33.302	0.076	0.066	0.060	0.056

1. Neighborhood 픽셀 사이즈 $(K - X)$
2. Smoothing 필터 사이즈 $(Z - X)$
3. Depth oracle 3d point 개수 $(I - X)$

$(K - 5, Z - 5, I - 128)$의 최적 값으로 training 진행

DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks



Total network 학습법

1) $loss_{MSE}$: RGB output에 대해 일반적인 MSE loss 사용

2) $loss_O$: 불투명 loss로 누적 불투명도가 1이상이도록 학습

→샘플 수가 작으면, 배경색을 누적연산에 포함시켜서 검게 되는데, 이를 방지하기 위해서 누적 불투명도가1이상이도록 학습 진행

$$\text{loss} = \alpha \cdot \text{loss}_{\text{MSE}} + \beta \cdot \text{loss}_O.$$

$$\text{loss}_O = \begin{cases} 0, & \text{if } \sum_{i=1}^{X} \delta_i \geq 1 \\ \left(\left(\sum_{i=1}^{X} \delta_i\right) - 1\right)^2, & \text{otherwise,} \end{cases}$$

네트워크 특징: cashing기법을 사용하지 않는 기법 중 가장 빠름

1) Batch 4096개의 ray 당 [2,4,8,16]sampling 진행

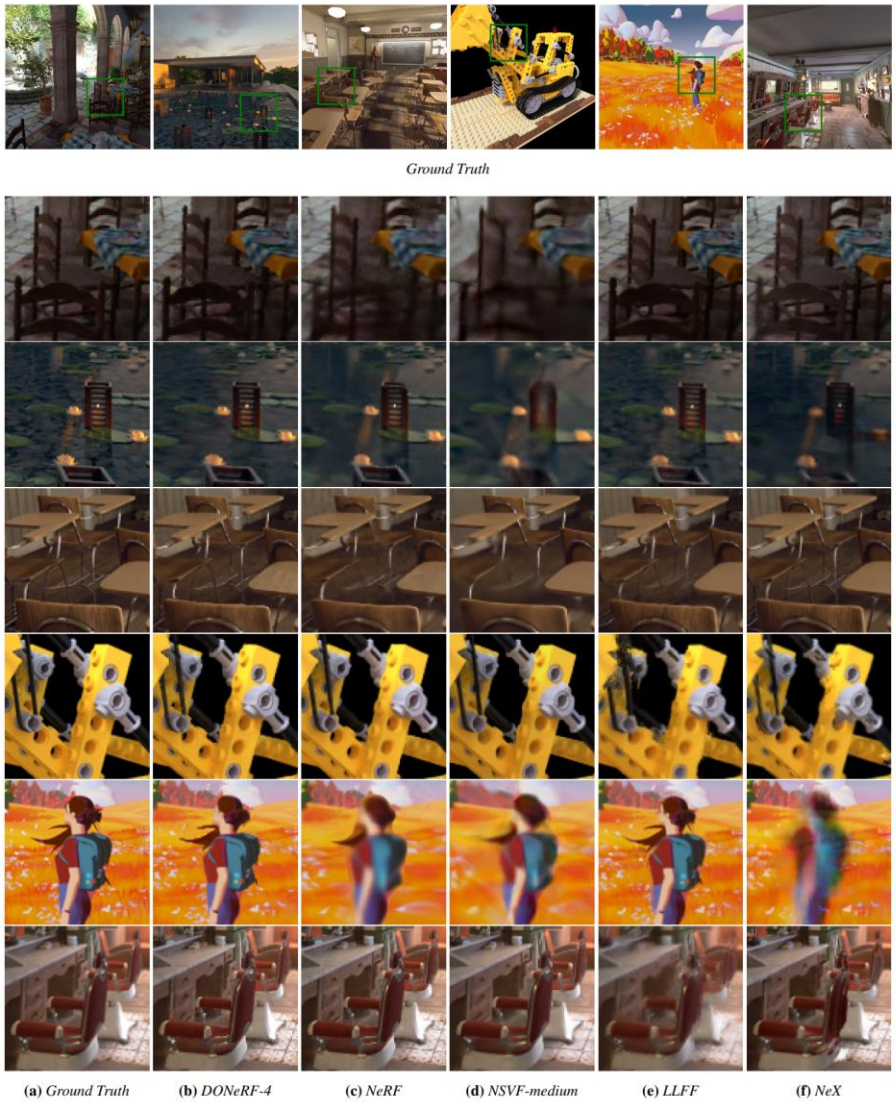
2) 기존 NeRF (4096개 ray당 256개 sampling)에 비해 15-78배 빨라졌고,
NerfingMVS (1024개 ray당 64개 sampling)과 비교하면 sampling 16개일 때만 기준 빠를 것으로 예상, 나머지는 훨씬 빨라 짐
Dense Depth Priors for Neural Radiance Fields from Sparse Input Views와 비교해도 (1024개의 ray 당 256개) 가장 빠름

입력 영상 개수 비교

1) Large depth forward-facing scene에 맞게 210 training images, 60 test images 필요

2) Inward facing인 NeRF는 최소 25장-100장 필요
NerfingMVS는 ScanNet 내의 모든 frame이 필요함
Dense Depth Priors for Neural Radiance Fields from Sparse Input Views는 가장 적은 양의 ScanNet 데이터 셋으로 (18-20 train, 8 test) 데이터로 room scaled-rendering이 가능함

DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks



DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks

			San Miguel		*Pavillon*		*Classroom*		*Bulldozer*		*Forest*		*Barbershop*		Average	
Method	Storage [MiB]	MFLOP per pixel	PSNR	FLIP	PSNR	FLIP	PSNR	FLIP	PSNR	FLIP	PSNR	FLIP	PSNR	FLIP	PSNR	FLIP
DONeRF-2	3.6	2.70	26.01	.094	30.50	.103	31.66	.061	30.15	.063	29.29	.082	29.41	.074	29.50	.079
DONeRF-2-noGT	3.6	2.70	25.33	.098	29.84	.103	30.11	.067	26.92	.077	28.36	.089	29.01	.075	28.26	.085
DONeRF-4	3.6	4.36	27.41	.080	31.07	.098	33.43	.058	33.46	.048	30.63	.077	30.84	.065	31.14	.071
DONeRF-4-noGT	3.6	4.36	26.19	.090	30.69	.096	31.44	.061	29.78	.060	29.31	.086	30.42	.067	29.64	.077
DONeRF-8	3.6	7.66	28.65	.071	31.46	.096	35.23	.048	35.88	.039	**32.09**	**.070**	31.72	.060	32.50	.064
DONeRF-8-noGT	3.6	7.66	26.88	.086	31.56	.091	33.19	.055	32.96	.047	29.98	.084	31.73	.062	31.05	.071
DONeRF-16	3.6	14.29	29.67	.065	31.79	.094	**36.27**	**.045**	**36.98**	**.036**	31.32	.074	32.15	.059	**33.03**	**.062**
DONeRF-16-noGT	3.6	14.29	27.70	.078	32.22	**.088**	34.63	.049	35.41	.040	30.74	.079	32.80	.057	32.25	.065
NeRF	3.2	211.42	25.19	.117	29.54	.115	34.02	.056	36.83	.038	23.90	.151	**33.63**	**.052**	30.52	.088
NeRF (log + warp)	3.2	211.42	28.98	.074	**32.88**	.089	35.19	.051	36.22	.040	28.97	.101	33.60	.055	32.64	.068
NSVF-small	**2.3**	74.66	24.00	.132	29.42	.110	31.00	.070	25.75	.167	23.79	.159	27.72	.094	26.95	.122
NSVF-medium	4.6	132.03	25.07	.110	29.81	.105	33.04	.055	26.51	.163	25.08	.135	29.62	.077	28.19	.108
NSVF-large	8.3	187.52	25.73	.097	30.48	.099	34.06	.051	33.14	.042	26.05	.119	30.61	.061	30.01	.078
LLFF	4130.6	**.03**	24.53	.106	27.50	.123	24.87	.114	24.76	.114	22.19	.148	24.13	.129	24.66	.122
NeX	88.8	.06	28.07	.094	26.28	.174	30.34	.085	29.20	.072	20.95	.220	22.98	.152	26.30	.133
NeX-MLP	89.0	42.71	**30.68**	**.060**	30.41	.102	34.10	.046	34.03	.046	24.65	.125	29.45	.075	30.55	.076

Q & A)

감사합니다!