Radiance-based Neural RGB-D Reconstruction

Siyu ZHANG

Research Engineer

ZJU-SenseTime Joint Lab of 3D Vision

Contents

- Preliminary： Nerf(**Ne**ural **R**adiance **F**ields) as scene representation

- Paper sharing:
  - iMAP: Implicit Mapping and Positioning in Real-Time
  - Neural RGB-D Surface Reconstruction

- Taking a ~~deeper~~ look

Nerf(Neural Radiance Fields)

- Radiance

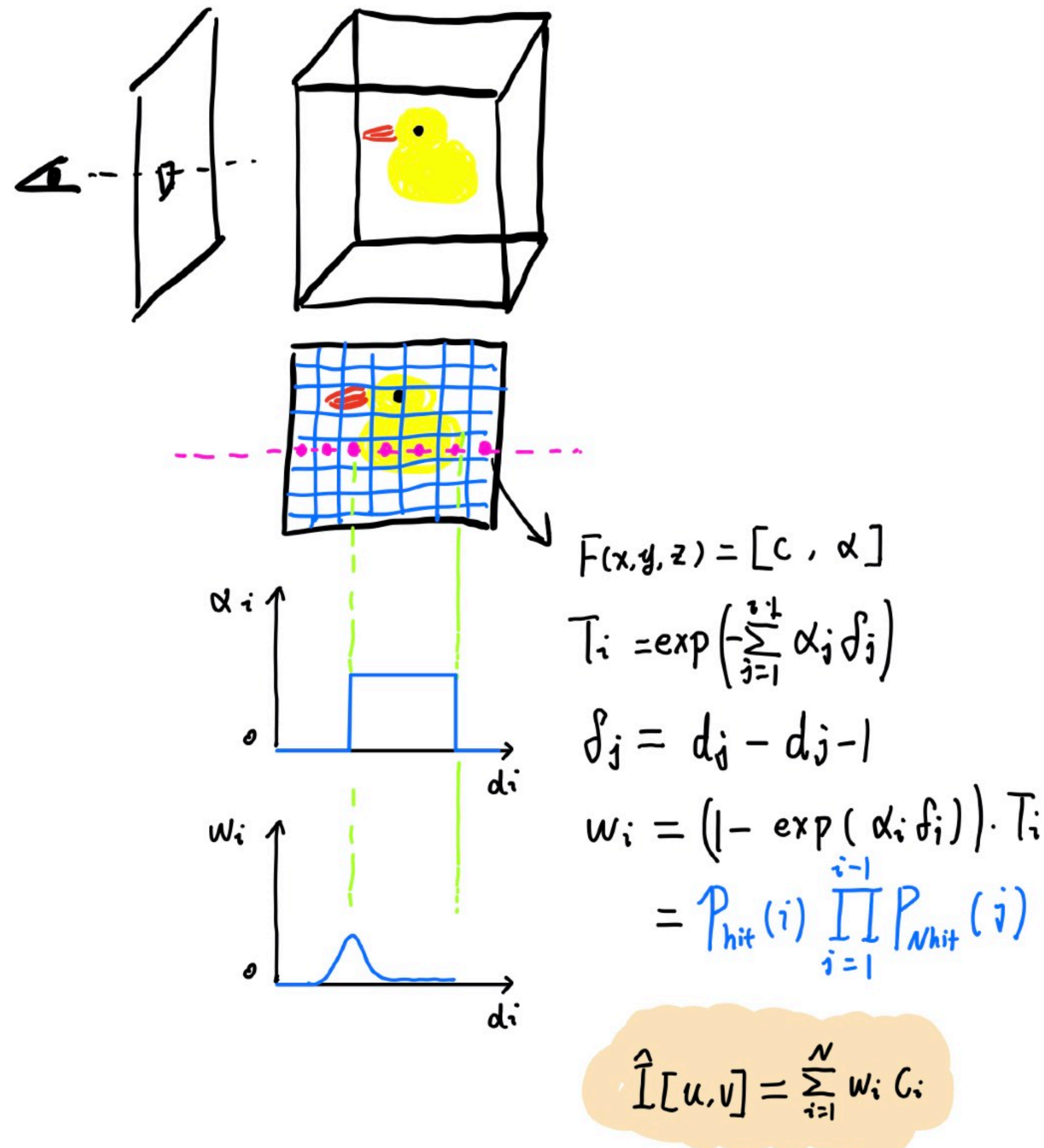
Definition: power per unit solid angle per projected unit area.

$$L(\mathrm{p}, \omega) \equiv \frac{\mathrm{d}^2\Phi(\mathrm{p}, \omega)}{\mathrm{d}\omega\, \mathrm{d}A \cos\theta}$$

Ref: GAMES-101

Nerf(Neural Radiance Fields)

- Radiance
- Radiance Field & Volume Rendering
  - **Input** camera pose and **output** RGB image
  - Method:
    - Sample points along the ray (given camera pose)
    - Query RGBα value for each point (given point coordinate)
    - Accumulate radiance along the ray



Ref: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

Nerf(Neural Radiance Fields)

- Radiance, Radiance Field & Volume Rendering

- How to *NEURALIZE* it?

Ref: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

Nerf(Neural Radiance Fields)

- Radiance, Radiance Field, Volume Rendering

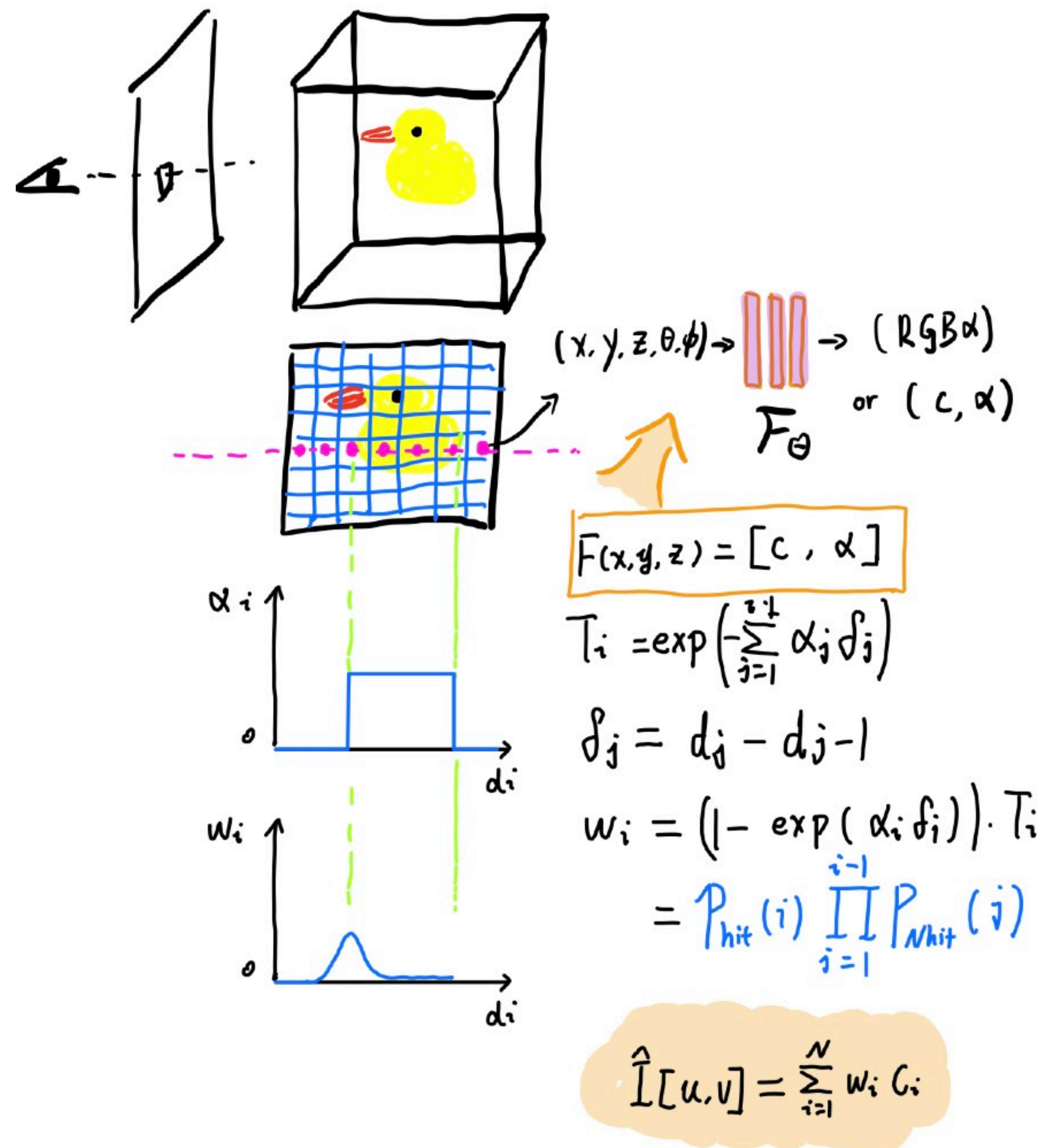
- How to *NEURALIZE* it?

    - Inference: Volume rendering

        - Sample points along the ray (given camera pose)

        - Query RGBα value for each point **with MLP** (given point coordinate **and view direction**)

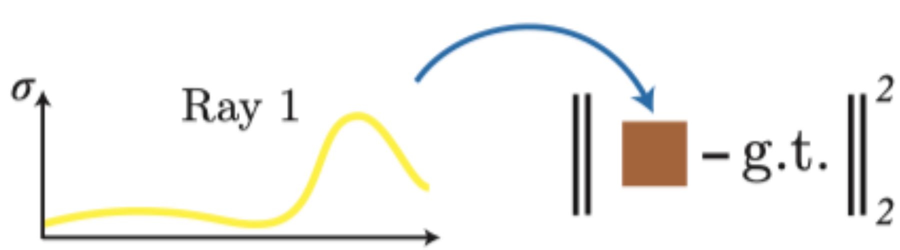
        - Accumulate radiance along the ray

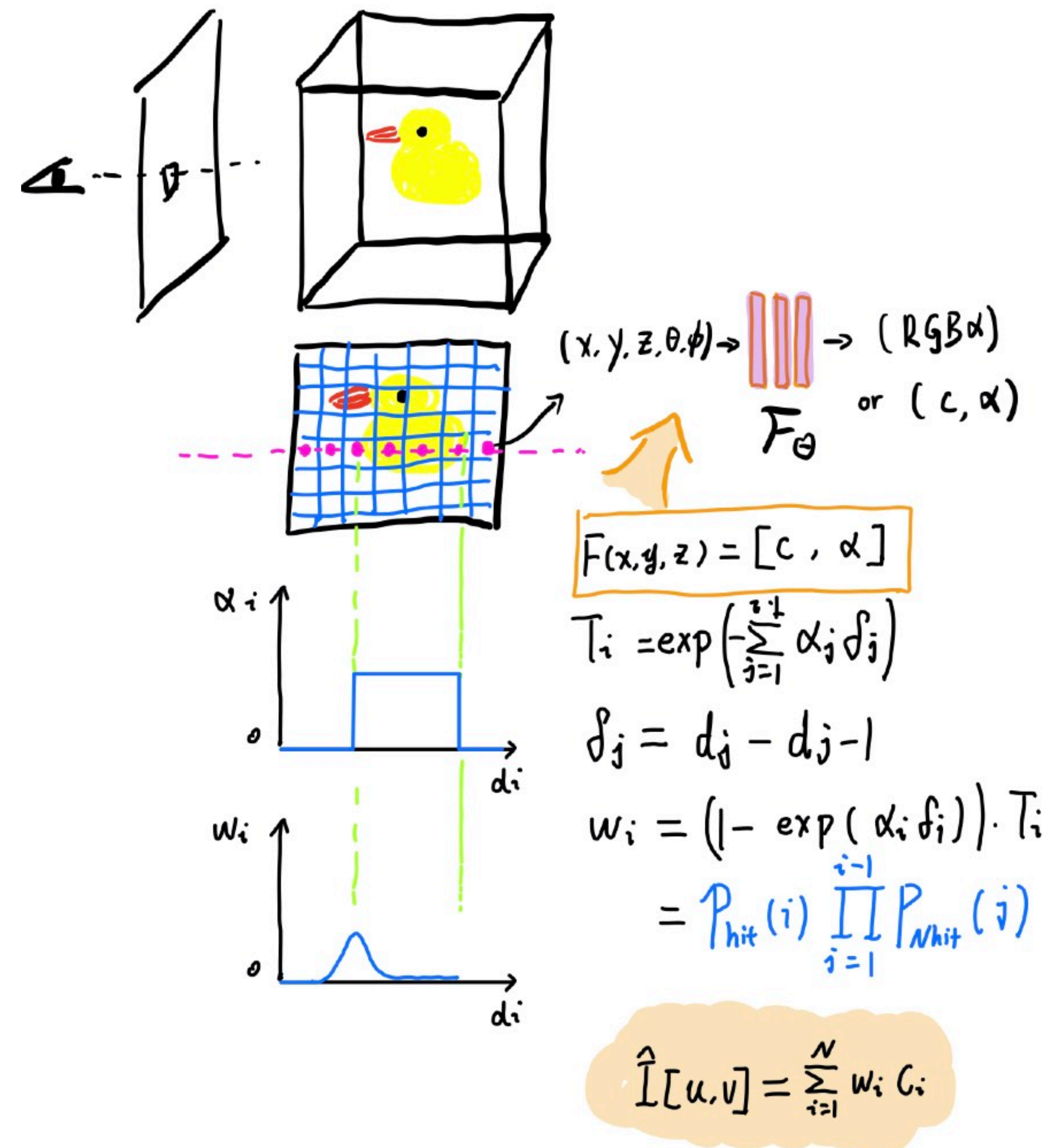


Ref: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

Nerf(Neural Radiance Fields)

- Radiance, Radiance Field, Volume Rendering
- How to *NEURALIZE* it?
    - Inference: Volume rendering
    - Training: Sample N rays and supervise with color value

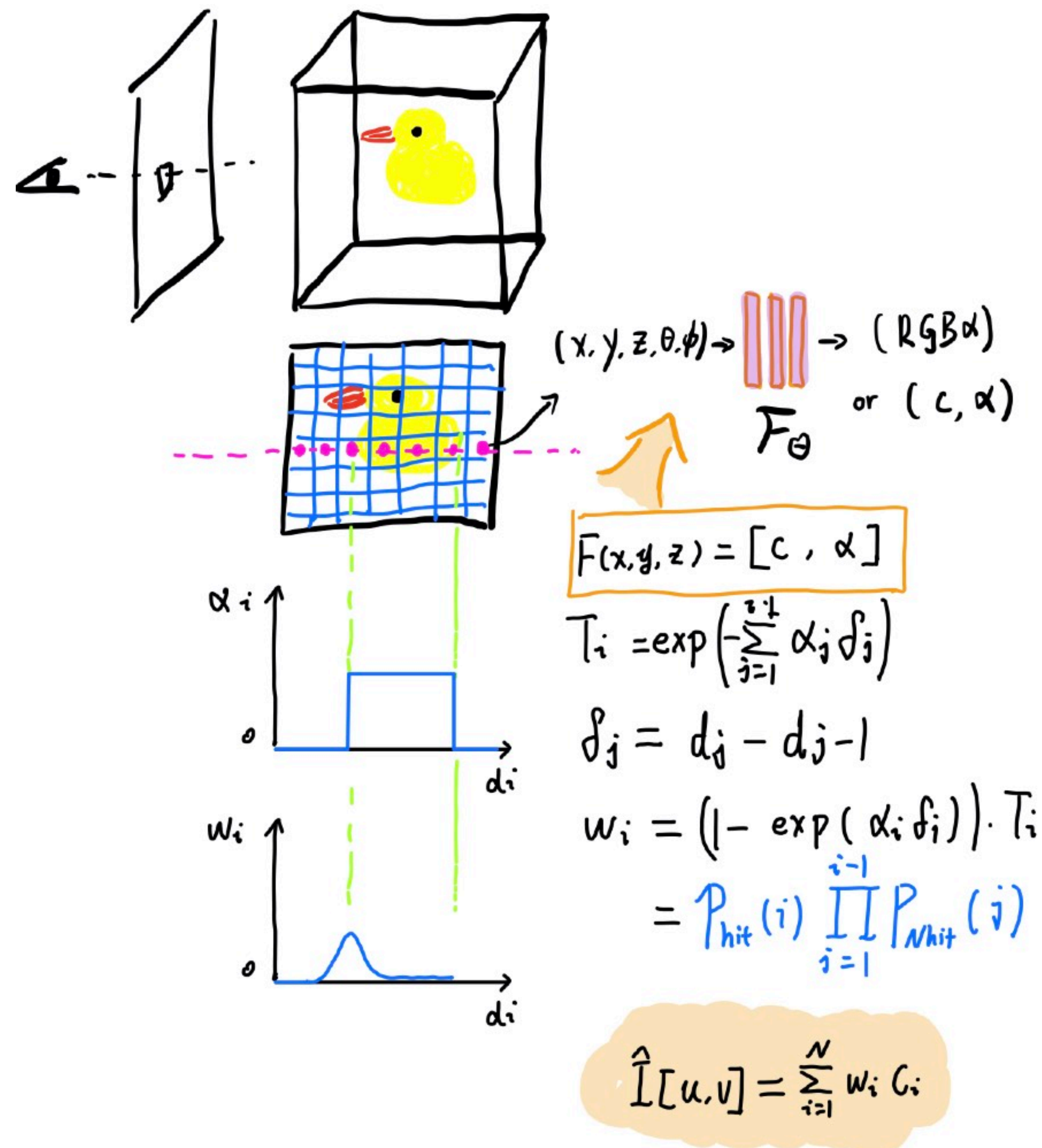




Ref: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

Nerf(Neural Radiance Fields)

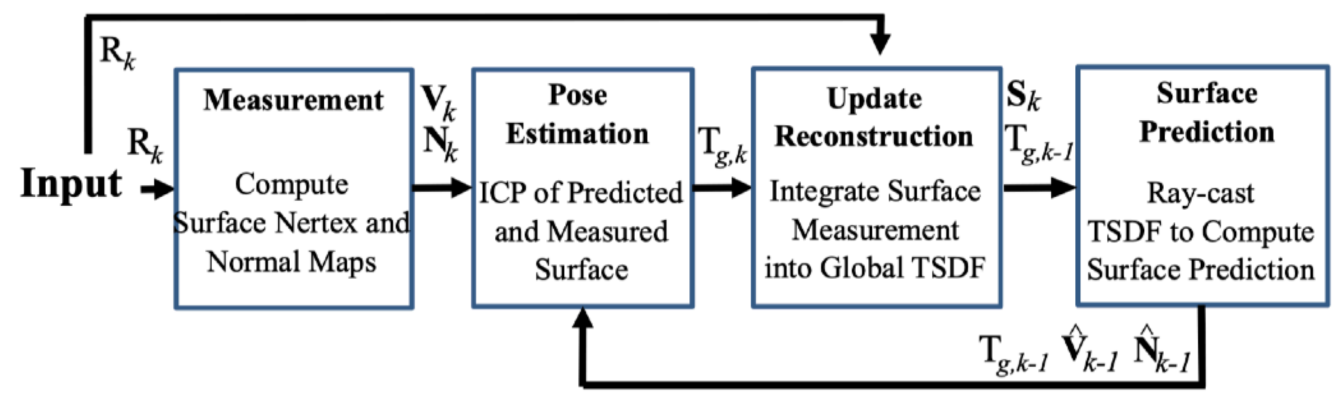
- Radiance, Radiance Field, Volume Rendering
- How to *NEURALIZE* it?
  - Inference: Volume rendering
  - Training: Sample N rays and supervise with color value
- AMAZING Part about NeRF: Directly **model out coming radiance** without knowing incoming radiance



Ref: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

iMAP

- Preliminary: RGB-D Reconstruction
- Input: RGB image + Depth
- Output: Reconstructed scene (in TDSF, Surfel, or NeRF)
- Key Steps:
  - Camera tracking: track camera pose w.r.t. global map
  - Local map building: reconstruction for local region
  - Global map integration: fuse local reconstruction to global reconstruction



Ref: KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera

iMAP

- In a nutshell: **RGB-D Reconstruction** using **NeRF** as scene representation (instead of TSDF or Surfel)
- Motivation: Leverage NeRF representation for RGB-D reconstruction
- Contribution: Real-Time RGB-D reconstruction with NeRF representation

- Input: RGB image + Depth
- Output: Reconstructed scene (in TDSF, Surfel, or NeRF)
- Key Steps:
  - Camera tracking
  - Local map building
  - Global map integration

iMAP

- In a nutshell: **RGB-D Reconstruction** using **NeRF** as scene representation (instead of TSDF or Surfel)

- Input & Output

- Input: RGB image + Depth
- Output: Reconstructed scene (in TDSF, Surfel, or NeRF)
- Key Steps:
  - Camera tracking
  - Local map building
  - Global map integration

iMAP

- In a nutshell: Real-time **RGB-D Reconstruction** using **NeRF** as scene representation (instead of TSDF or Surfel)
- Input & Output
- Camera pose: assumed to be obtained with existing approach (known)

- Input: RGB image + Depth
- Output: Reconstructed scene (in TDSF, Surfel, or NeRF)
- Key Steps:
  - Camera tracking
  - Local map building
  - Global map integration

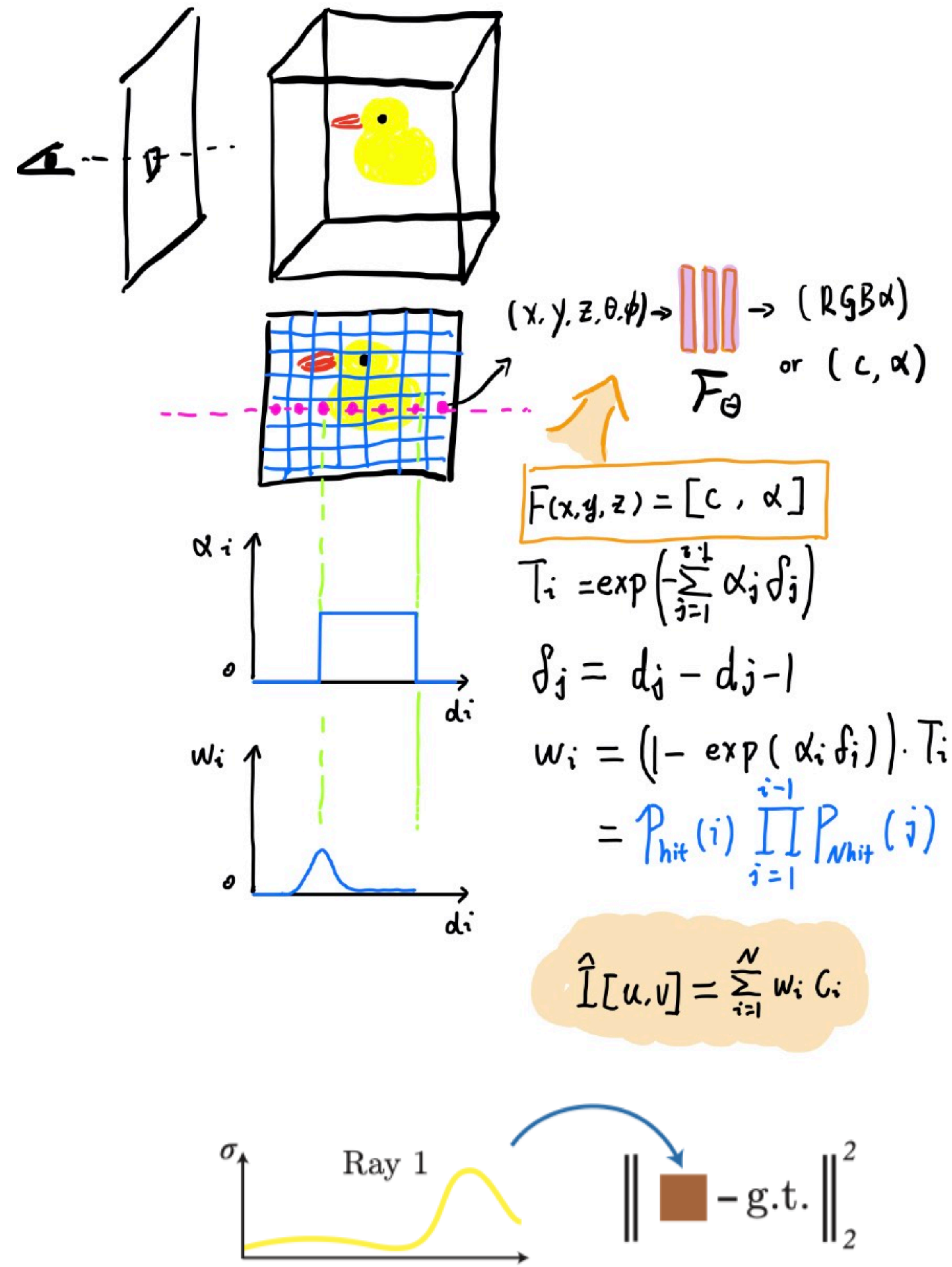
iMAP

- In a nutshell: **RGB-D Reconstruction** using **NeRF** as scene representation (instead of TSDF or Surfel)
- Input & Output
- Camera pose: assumed to be obtained with existing approach (known)
- **Key of the paper**: How to reconstruct Neural Radiance Field w/o additional supervision
  - i.e. How to train NeRF together with camera tracking?

- Input: RGB image + Depth
- Output: Reconstructed scene (in TDSF, Surfel, or NeRF)
- Key Steps:
  - Camera tracking
  - **Local map building**
  - **Global map integration**

iMAP

- How to <u>reconstruct Neural Radiance Field w/o additional supervision</u>
  - NeRF recap: Volume rendering
    - Training data: RGB image + camera pose
  - Steps:
    - Camera pose tracking
    - Keyframe selection: the lower overlap the better
    - Train the network with selected keyframe (with camera pose)



iMAP

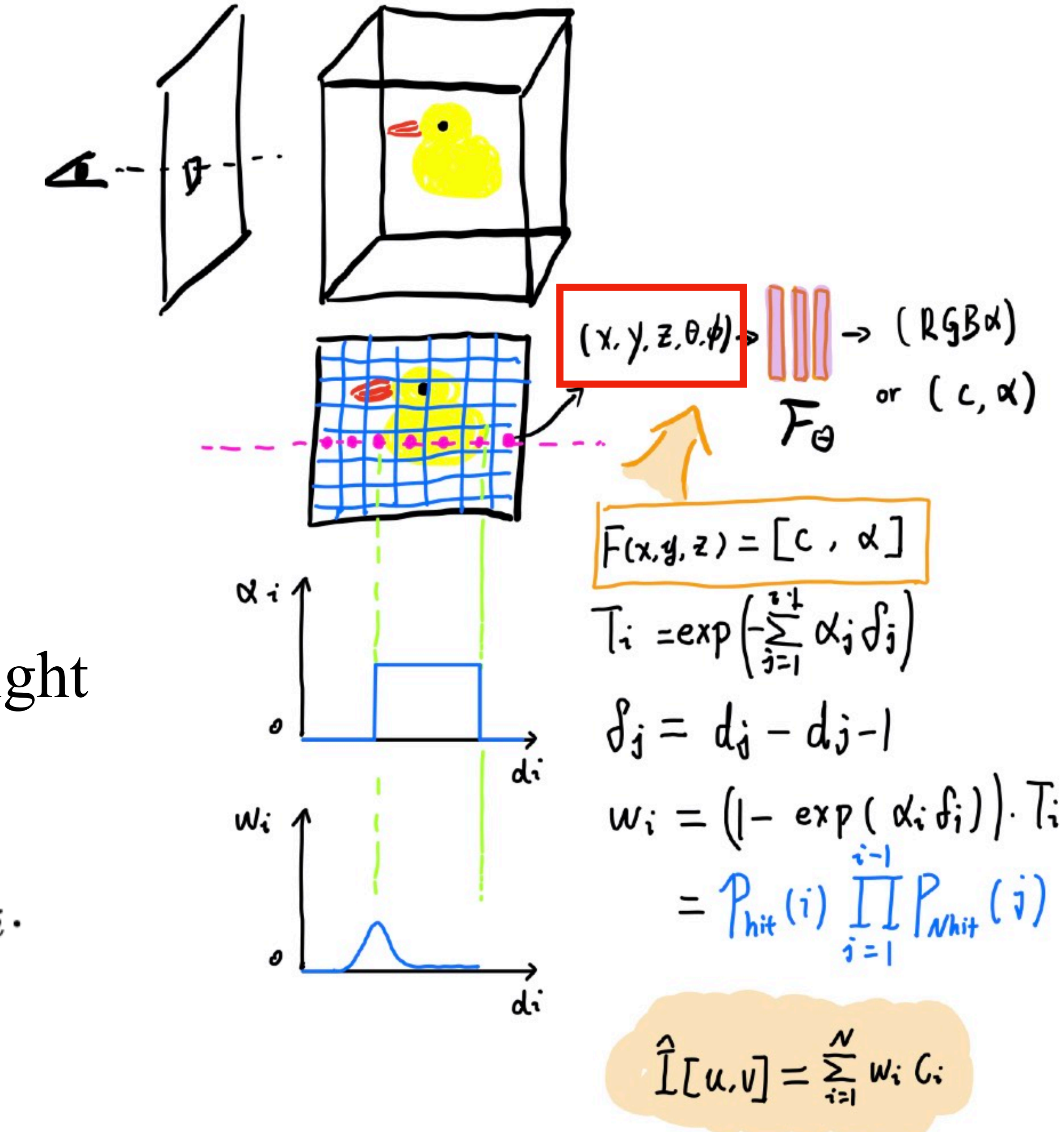
- More details…
  - Differences compared to NeRF
    - Does not model <u>view independent rendering</u>
    - Additional geometric loss: down-weight depth loss at uncertain regions

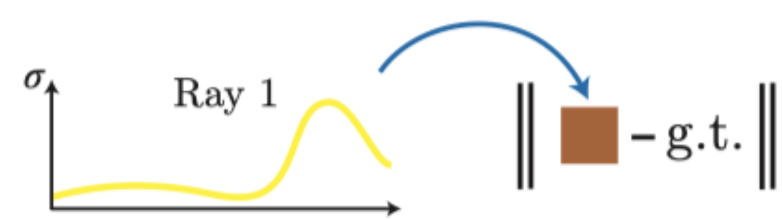
$$\hat{D}[u,v] = \sum_{i=1}^{N} w_i d_i, \quad \hat{I}[u,v] = \sum_{i=1}^{N} w_i \mathbf{c}_i.$$

$$\hat{D}_{var}[u,v] = \sum_{i=1}^{N} w_i (\hat{D}[u,v] - d_i)^2.$$

$$L_g = \frac{1}{M} \sum_{i=1}^{W} \sum_{(u,v) \in s_i} \frac{e_i^g[u,v]}{\sqrt{\hat{D}_{var}[u,v]}}.$$

$$\min_{\theta, \{T_i\}} (L_g + \lambda_p L_p).$$



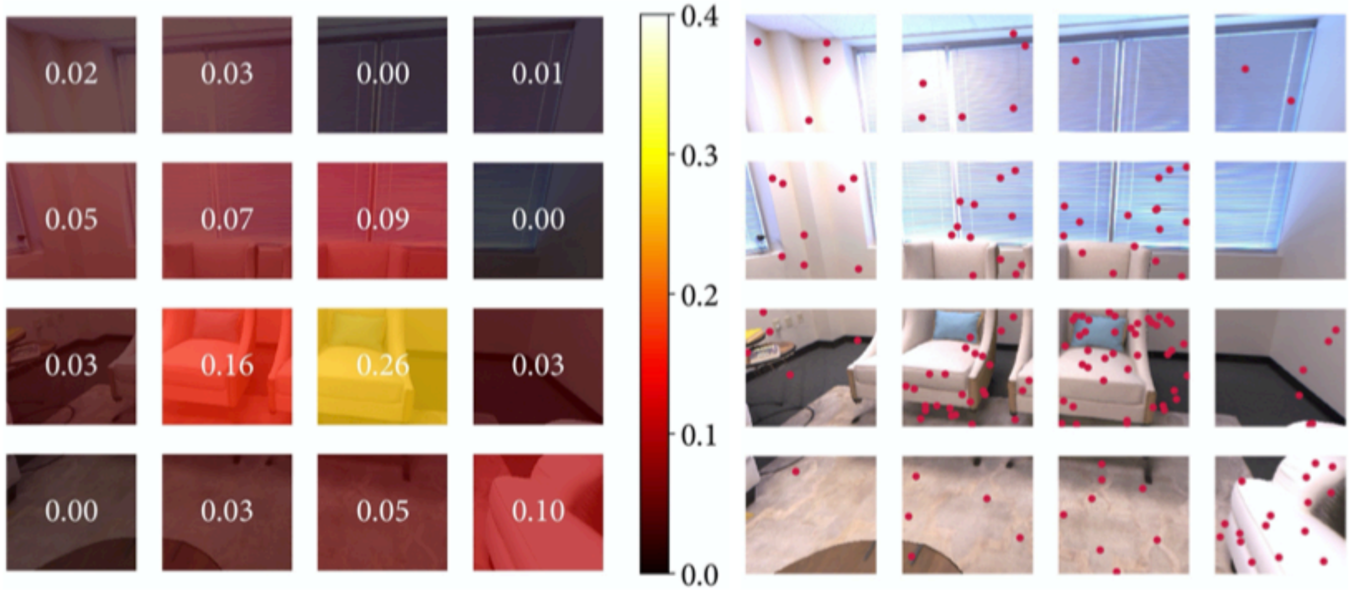


iMAP

- More details…
  - Differences compared to NeRF
  - Keyframe Selection: whether nor to add current frame to training set
    - Portion of current frame that is already explainable by existing model (measured by **normalized depth error**)

$$P = \frac{1}{|s|} \sum_{(u,v) \in s} \mathbb{1} \left(\frac{\left| D[u,v] - \hat{D}[u,v] \right|}{D[u,v]} < t_D \right).$$

iMAP

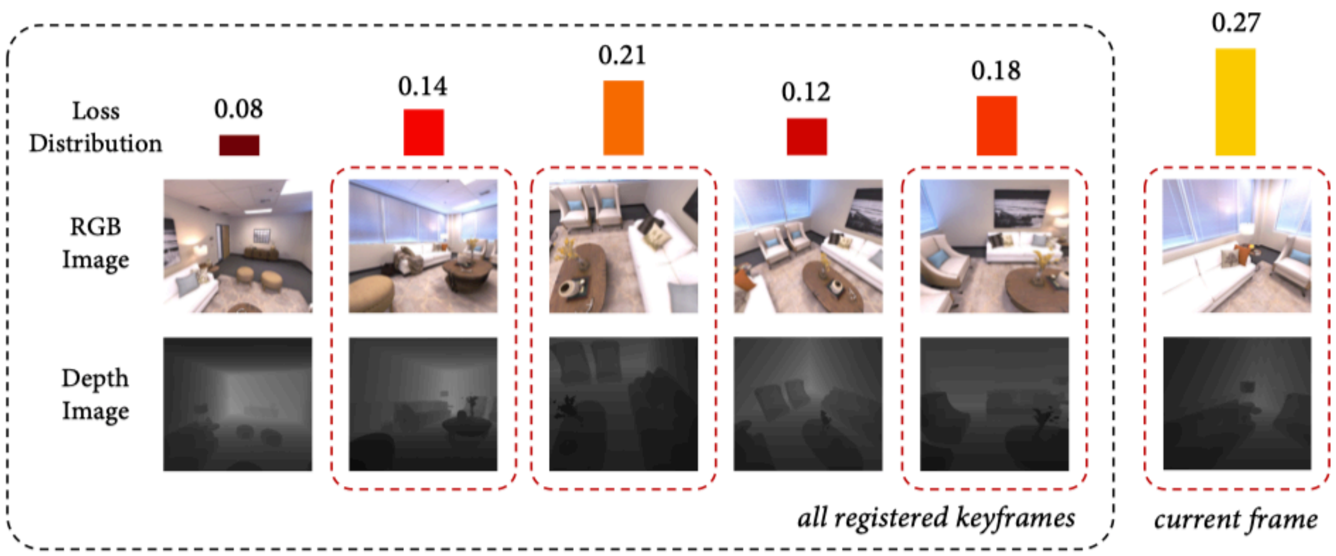


- More details
  - Keyframe selection
  - Image active sampling:

    —— strategy for pixel sampling as supervision

    - Uniformly sample for the first time (one sample per [8x8] grid)
    - Normalize loss to get probability of been sampled for each region

iMAP



- More details
  - Keyframe selection
  - Image active sampling
  - Keyframe active sampling & Bounded Keyframe Selection
    - For each iteration of training:
      - Random sample keyframes according to loss distribution
      - Always include last keyframe

iMAP

- Evaluation of reconstruction: Replica dataset, use iMAP pose

		room-0	room-1	room-2	office-0	office-1	office-2	office-3	office-4	Avg.
iMAP	**# Keyframes**	11	12	12	10	11	10	14	11	13.37
	Acc. [cm]	3.58	3.69	4.68	5.87	3.71	4.81	4.27	4.83	4.43
	Comp. [cm]	5.06	4.87	5.51	6.11	5.26	5.65	5.45	6.59	5.56
	Comp. Ratio [< 5cm %]	83.91	83.45	75.53	77.71	79.64	77.22	77.34	77.63	79.06
TSDF Fusion	**Acc.** [cm]	4.21	3.08	2.88	2.70	2.66	4.27	4.07	3.70	3.45
	Comp. [cm]	5.04	4.35	5.40	10.47	10.29	6.43	6.26	4.78	6.63
	Comp. Ratio [< 5cm %]	76.90	79.87	77.79	79.60	71.93	71.66	65.87	77.11	75.09

- Quantitive result: marginal improvement

iMAP

- Evaluation of reconstruction: Replica dataset, use iMAP pose

(a) Chair (b) Back of Objects (c) Small Objects (d) Black Chair

- Qualitative result
  - Better in hole filing (<u>learn the unobservable geometric structures</u> by color supervision)
  - More cohesive, less artifacts

iMAP

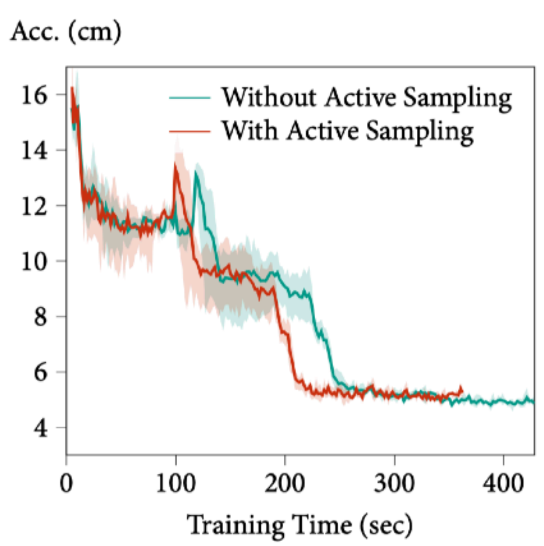
- Evaluation of camera tracking: TUM RGB-D dataset

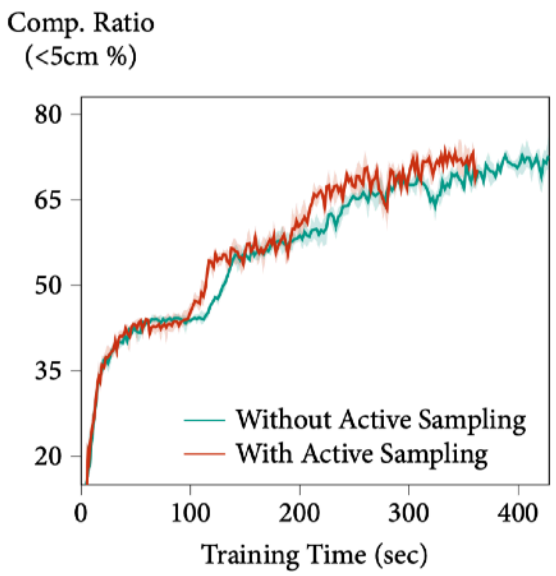
	fr1/desk (cm)	fr2/xyz (cm)	fr3/office (cm)
iMAP	4.9	2.0	5.8
BAD-SLAM	1.7	1.1	1.73
Kintinuous	3.7	2.9	3.0
ORB-SLAM2	1.6	0.4	1.0

Table 3: ATE RMSE in cm on TUM RGB-D dataset.

- Ablation study

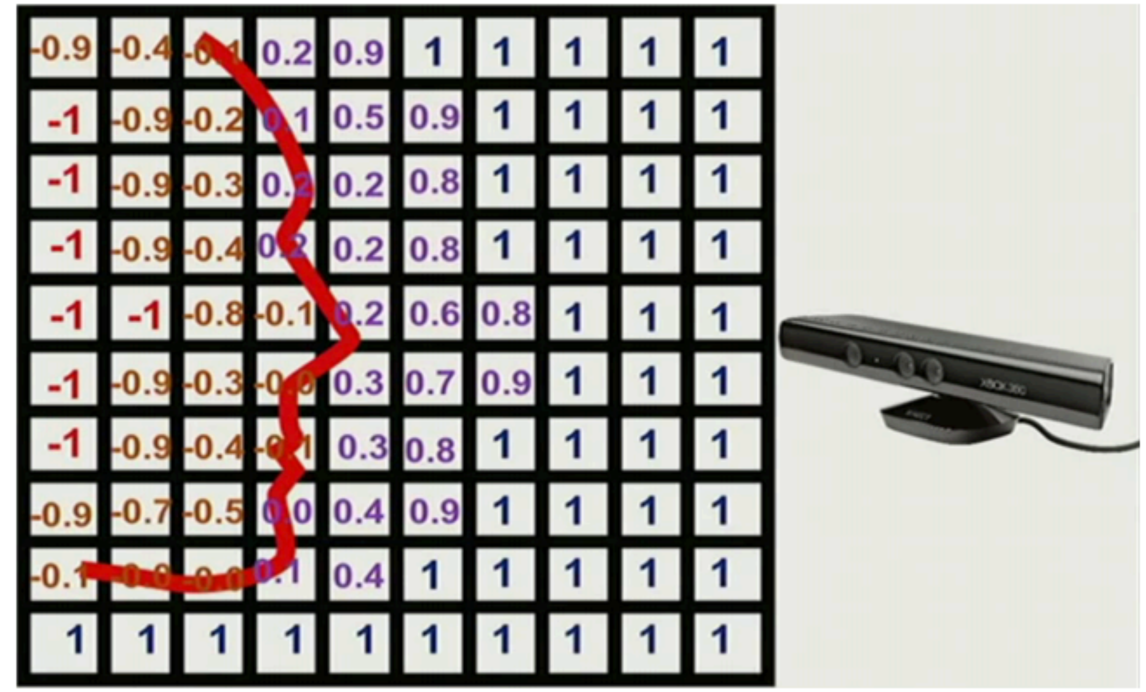
	Default	Width		Window		Pixels	
		128	512	3	10	100	400
Tracking Time [ms]	101	80	173	84	144	74	160
Joint Optim. Time [ms]	448	357	777	373	647	340	716
Comp. Ratio [<5cm %]	77.22	75.79	76.91	75.82	77.35	77.33	77.49





Neural RGB-D Surface Reconstruction

- Preliminary about TSDF (Truncated Signed Distance Function)
  - **Signed**: Positive if outside model, negative otherwise
  - **Distance**: value = distance to the surface
  - **Truncated**: equal to a fix value when far enough to surface
  - **Function**: input position (x, y, z), output distance value

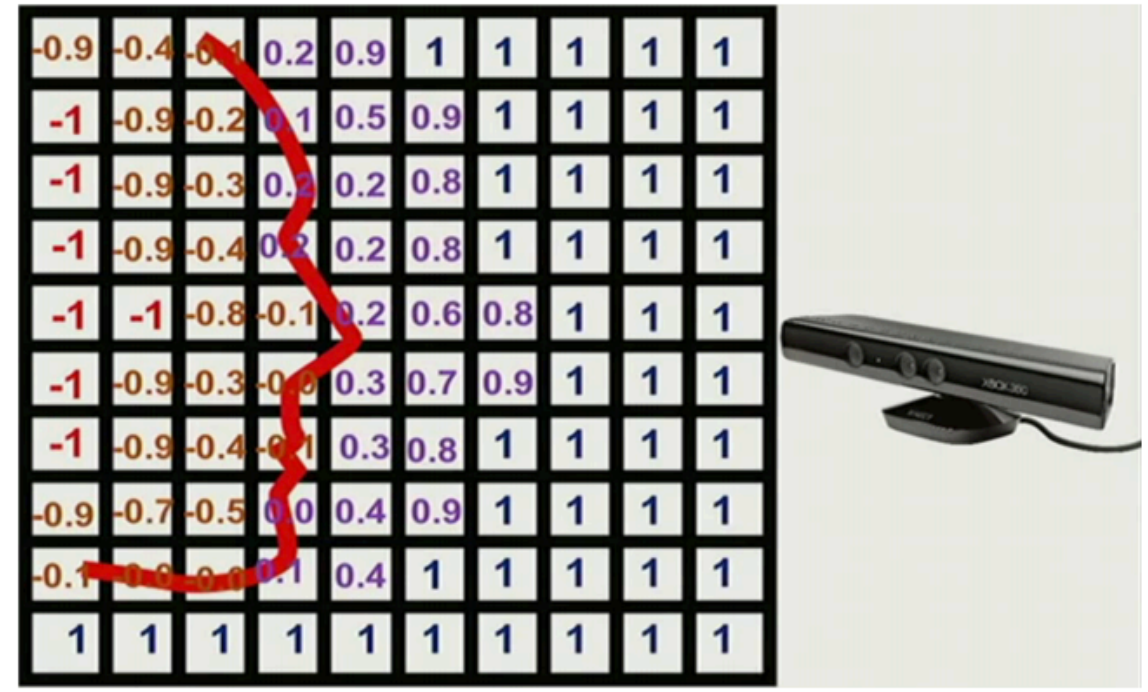


Neural RGB-D Surface Reconstruction

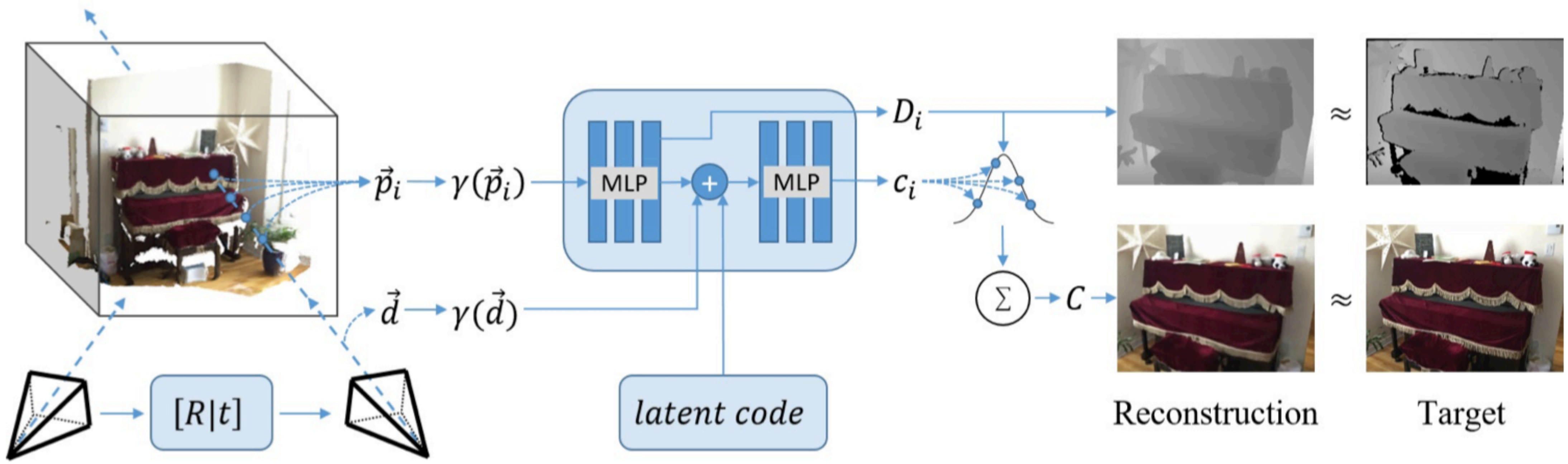
- Preliminary about TSDF (*Truncated* Signed **Distance** FUNCTION)

- Convert to mesh: Marching Cubes

- Convert to depth: surface rendering



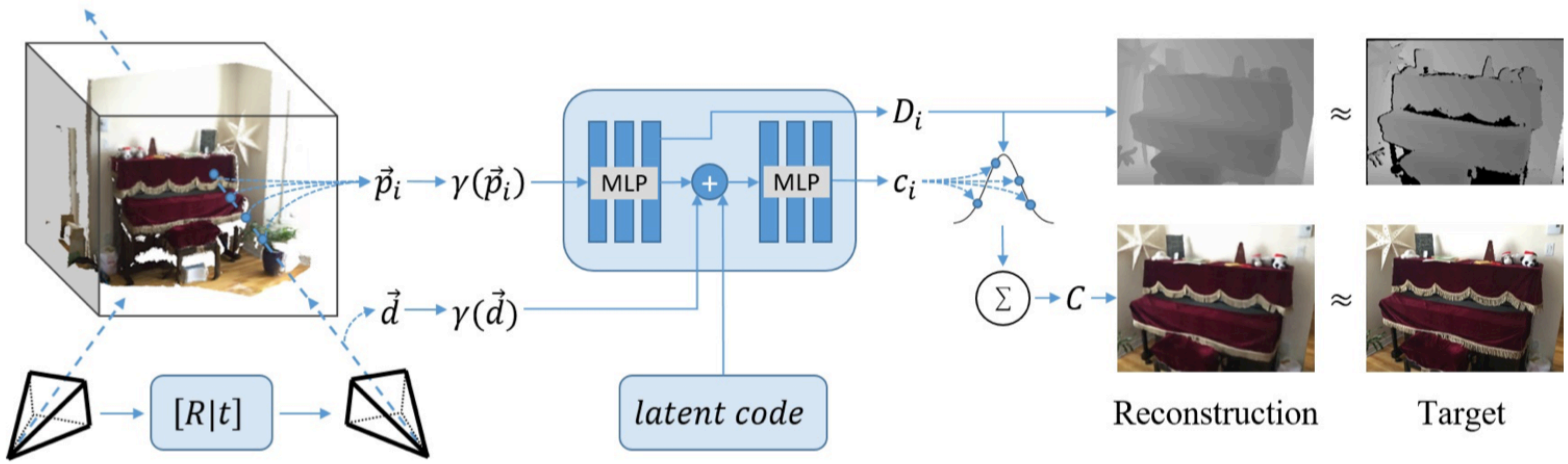
Neural RGB-D Surface Reconstruction



- **In a nutshell:** Improvement of NeRF representation by:

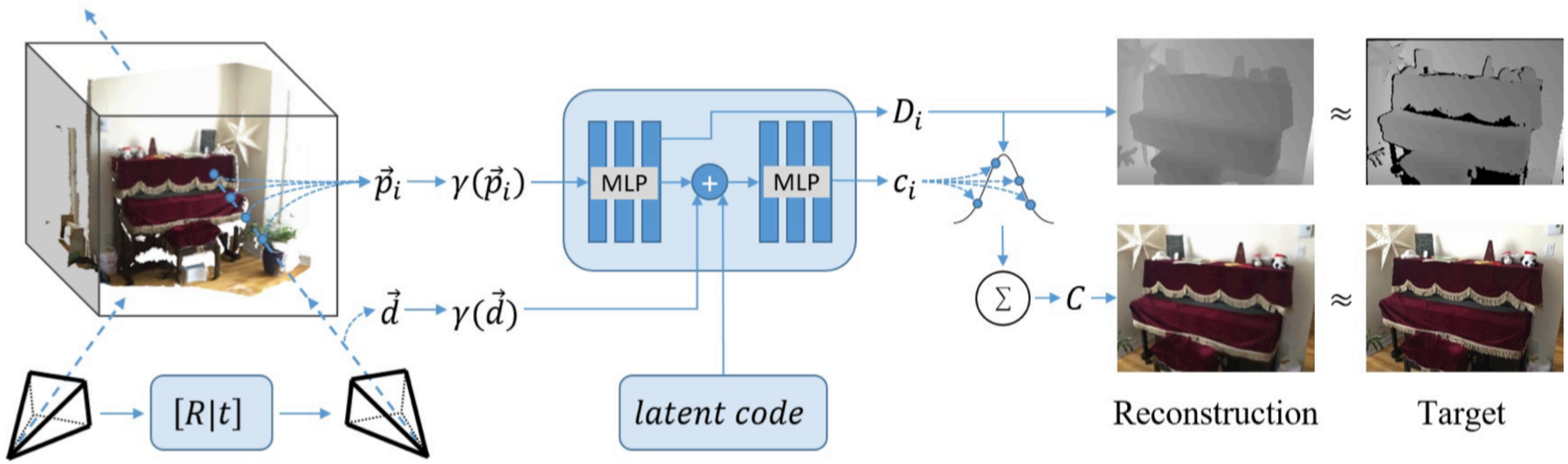
- Motivation: TSDF provide better geometric detail, make use of color supervision

Neural RGB-D Surface Reconstruction



- **In a nutshell:** Improvement of NeRF representation by:
  - Estimate <u>TSDF + color</u> instead of <u>volume density + color</u> (for **hard boundary** and better shape)
  - Additional depth supervision (given RGB-D input)
  - Input latent code to control to correct for effects like auto-white balancing (<u>guarantee the same color for the same position</u>)

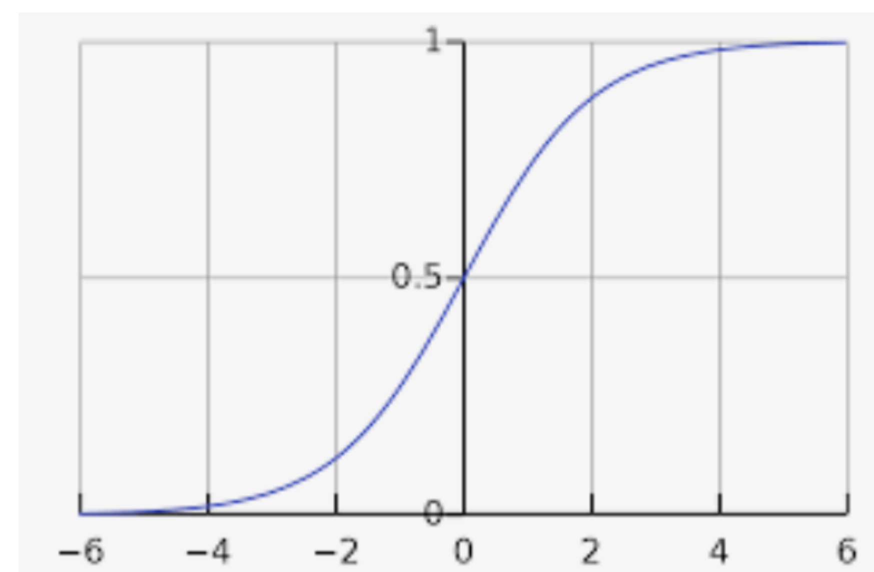
Neural RGB-D Surface Reconstruction



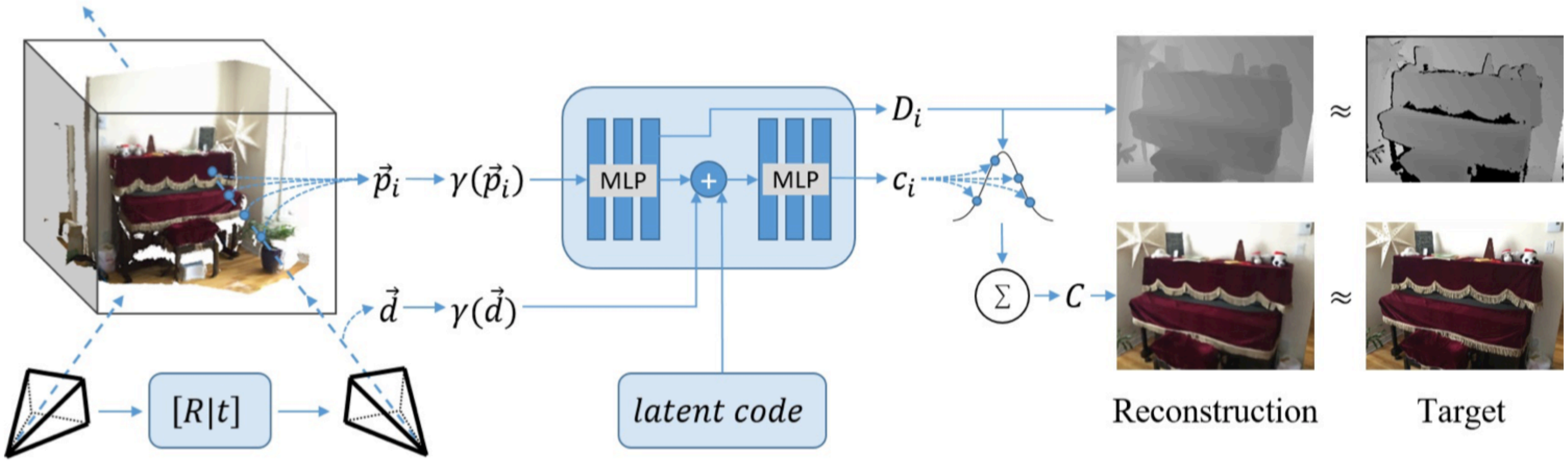
- Volume rendering for TSDF

$$w_i = \sigma(s \cdot D_i) \cdot \sigma(-s \cdot D_i).$$

$$\hat{D}[u,v] = \sum_{i=1}^{N} w_i d_i, \quad \hat{I}[u,v] = \sum_{i=1}^{N} w_i \mathbf{c}_i.$$



Neural RGB-D Surface Reconstruction



- Loss
  - Color objective (4)
    - Same as NeRF
  - Free-space objective (5)
    - Predicted TSDF value must be 1
  - Singed distance objective (6)
    - Predicted distance value must be closed to true distance value

$$\mathcal{L}(\mathcal{P}) = \sum_{b=0}^{B-1} \lambda_1 \mathcal{L}^b_{rgb}(\mathcal{P}) + \lambda_2 \mathcal{L}^b_{fs}(\mathcal{P}) + \lambda_3 \mathcal{L}^b_{tr}(\mathcal{P}).$$

$$\mathcal{L}^b_{rgb}(\mathcal{P}) = \frac{1}{|P_b|} \sum_{p \in P_b} (C_p - \hat{C}_p)^2. \qquad (4)$$

$$\mathcal{L}^b_{fs}(\mathcal{P}) = \frac{1}{|P_b|} \sum_{p \in P_b} \frac{1}{|S^{fs}_p|} \sum_{s \in S^{fs}_p} (D_s - 1)^2. \qquad (5)$$

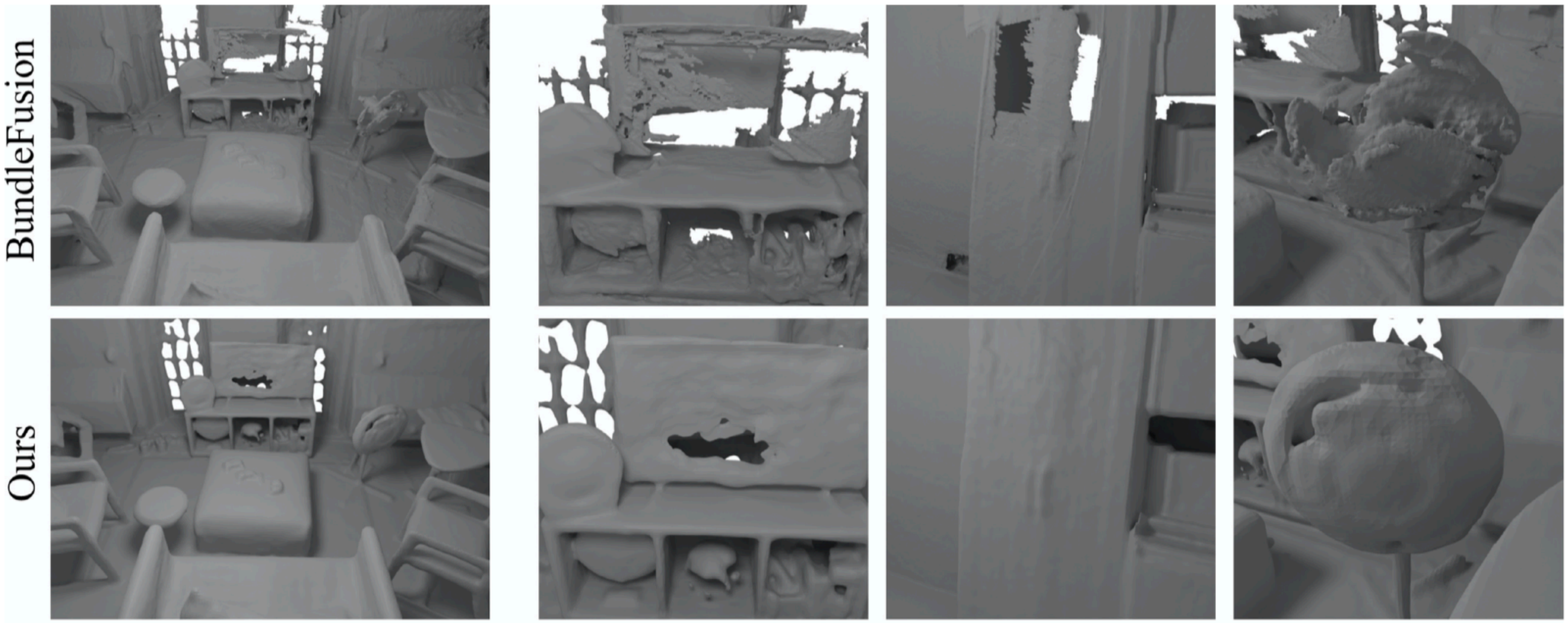
$$\mathcal{L}^b_{tr}(\mathcal{P}) = \frac{1}{P_b} \sum_{p \in P_b} \frac{1}{|S^{tr}_p|} \sum_{s \in S^{tr}_p} (D_s - \hat{D}_s)^2. \qquad (6)$$

Neural RGB-D Surface Reconstruction

Method	**C-ℓ_1** $\downarrow$	**IoU** $\uparrow$	**NC** $\uparrow$	**F-score** $\uparrow$
BundleFusion	0.062	0.528	0.869	0.701
COLMAP + Poisson	0.083	0.512	0.840	0.688
NeRF + Depth	0.073	0.385	0.716	0.619
Ours	**0.027**	**0.744**	**0.910**	**0.909**

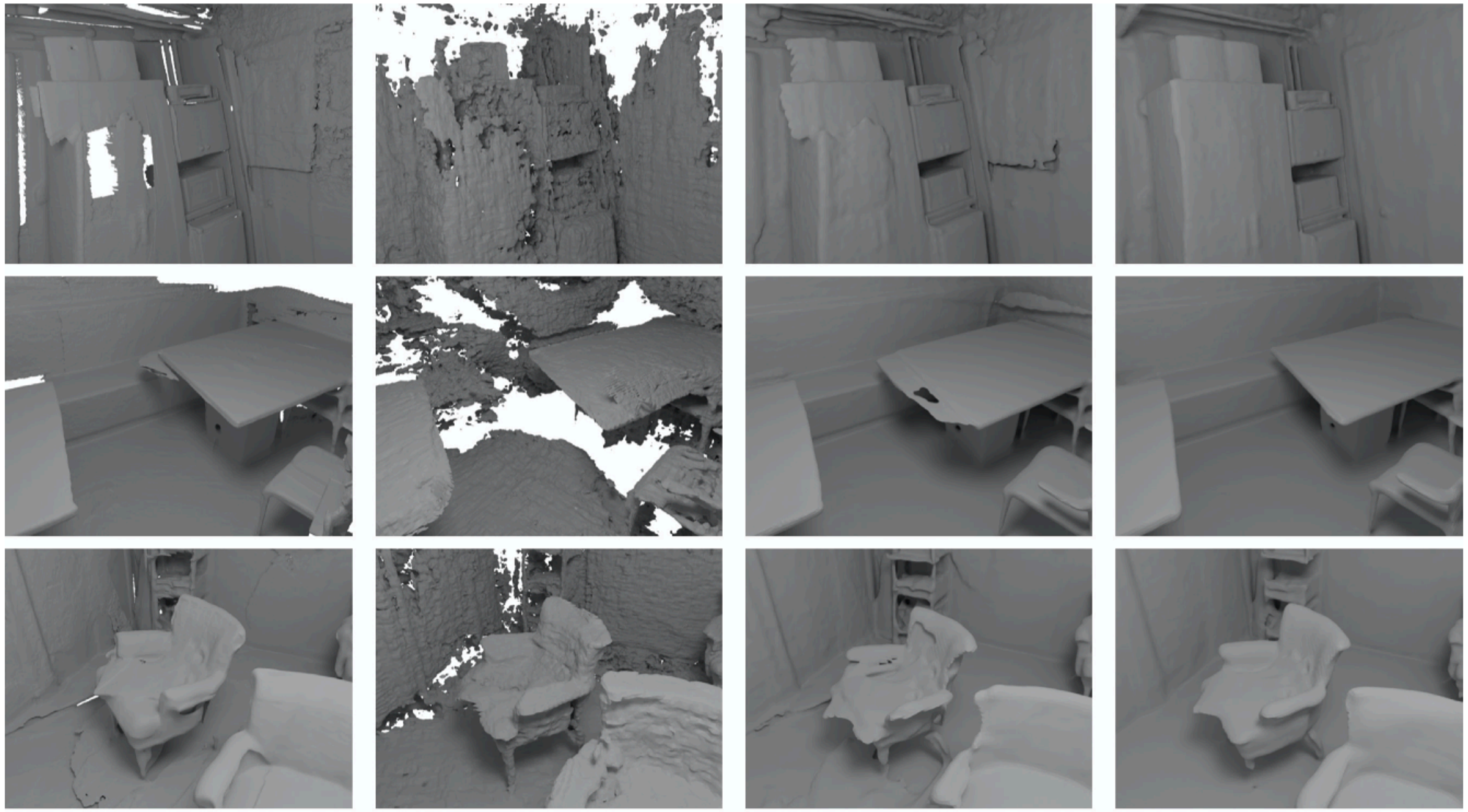
- Results
  - Quantitative evaluation on Synthetic dataset
  - Use BundleFusion pose
  - Evident improvement

Neural RGB-D Surface Reconstruction



- Qualitative Results
    - Better in hole filling

Neural RGB-D Surface Reconstruction



- Ablation studies

Comparison

	iMAP	Neu-Surf
Representation	Neural Radiance Filed (view independent)	TSDF + Neural Radiance Filed
Supervision	Color + Depth	Color + Depth
Real-Time	Yes	No

Taking a ~~DEEPER~~ Look!

- Modeling REAL PRB rendering
    - Camera center & view
    - Surface position, normal
    - Material BRDF (Cook-Torrance Model)
    - Incoming radiance: very hard to control & capture

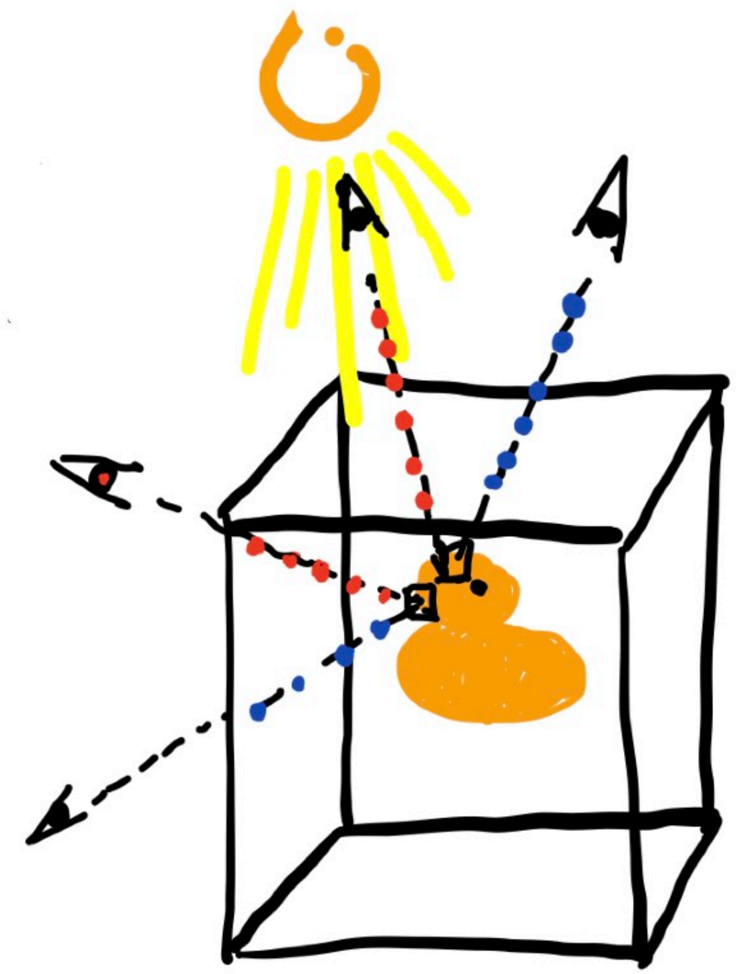
$$L(\hat{\boldsymbol{x}}, \boldsymbol{w}^o) = L^e(\hat{\boldsymbol{x}}, \boldsymbol{w}^o) + \int_\Omega B(\hat{\boldsymbol{x}}, \hat{\boldsymbol{n}}, \boldsymbol{w}^i, \boldsymbol{w}^o) L^i(\hat{\boldsymbol{x}}, \boldsymbol{w}^i)(\hat{\boldsymbol{n}} \cdot \boldsymbol{w}^i)\, d\boldsymbol{w}^i = M_0(\hat{\boldsymbol{x}}, \hat{\boldsymbol{n}}, \boldsymbol{v}), \quad (5)$$

Ref: GAMES-101

** Credit to Jiaming for this part*

Taking a ~~DEEPER~~ Look!

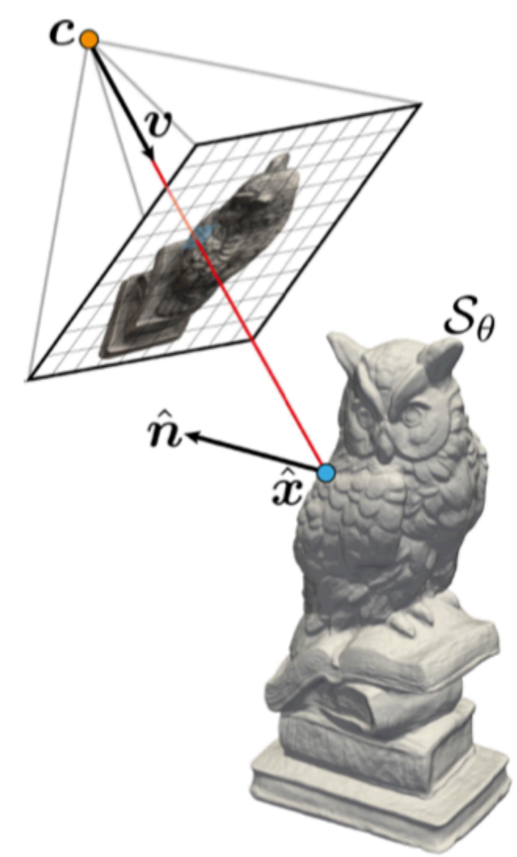
- Approximation of PBR by Deep Implicit Rendering
  - NeRF-style:
    - learn radiance along each possible ray
    - specular effect (caused by non-lambertion material or ambient light) **is encoded in the ray along certain direction**



** Credit to Jiaming for this part*

Taking a ~~DEEPER~~ Look!

- Approximation of PBR by Deep Implicit Rendering
  - NeRF-style: encode material and lighting along the ray
  - IDR-style:
    - Predict ambient light (as latent vector) with implicit network
    - Model material BRDF, ambient lighting, incoming lighting all inside Neural Render model (MLP) at intersection point



** Credit to Jiaming for this part*

Acknowledge

- Thanks to the help of Jiaming, Zhiyuan, Yifan and Huangdi for the thoughtful discussion and generously sharing of knowledge and insight

Thanks for your Attention

Siyu ZHANG

Research Engineer

ZJU-SenseTime Joint Lab of 3D Vision

zhangsiyu1@sensetime.com