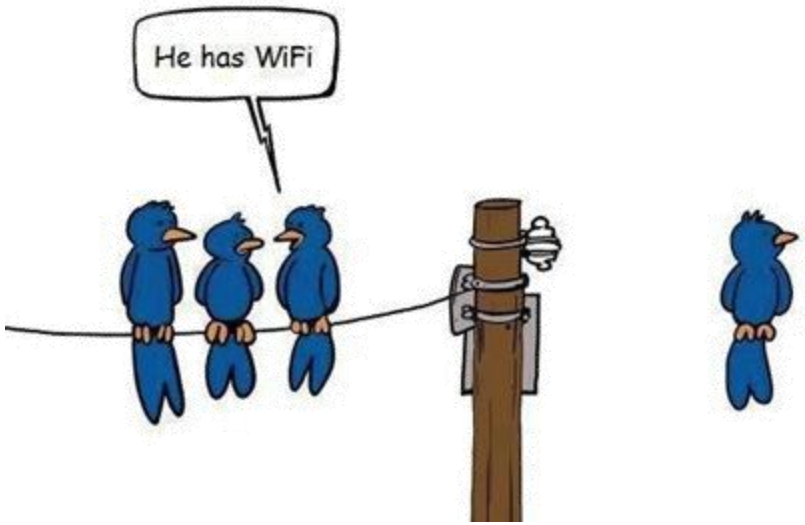
COM-405: Mobile Networks

Lecture 9.0: Wireless Sensing & Imaging

Haitham Hassanieh



Laboratory of SEnsing
& Networking Systems

EPFL

Imaging Non-Line-of-Sight 3D geometry with Radio Frequency Signals

VENUE & DATE

Mobile Network
Nov 18th , 2025

PRESEDENTED BY

Jiachen Lu
Laboratory of Sensing and Networking Systems

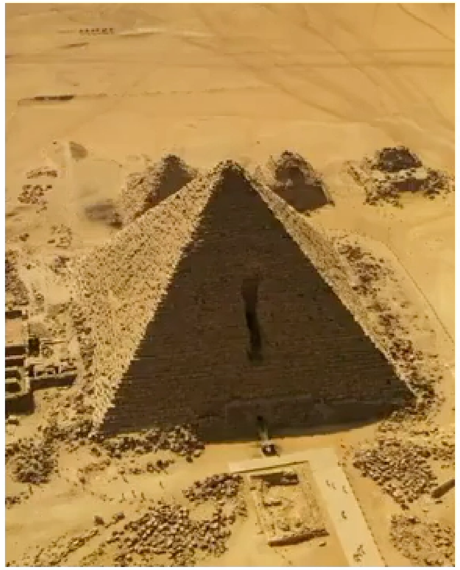
École polytechnique fédérale de Lausanne

Shape from X

Shape from X

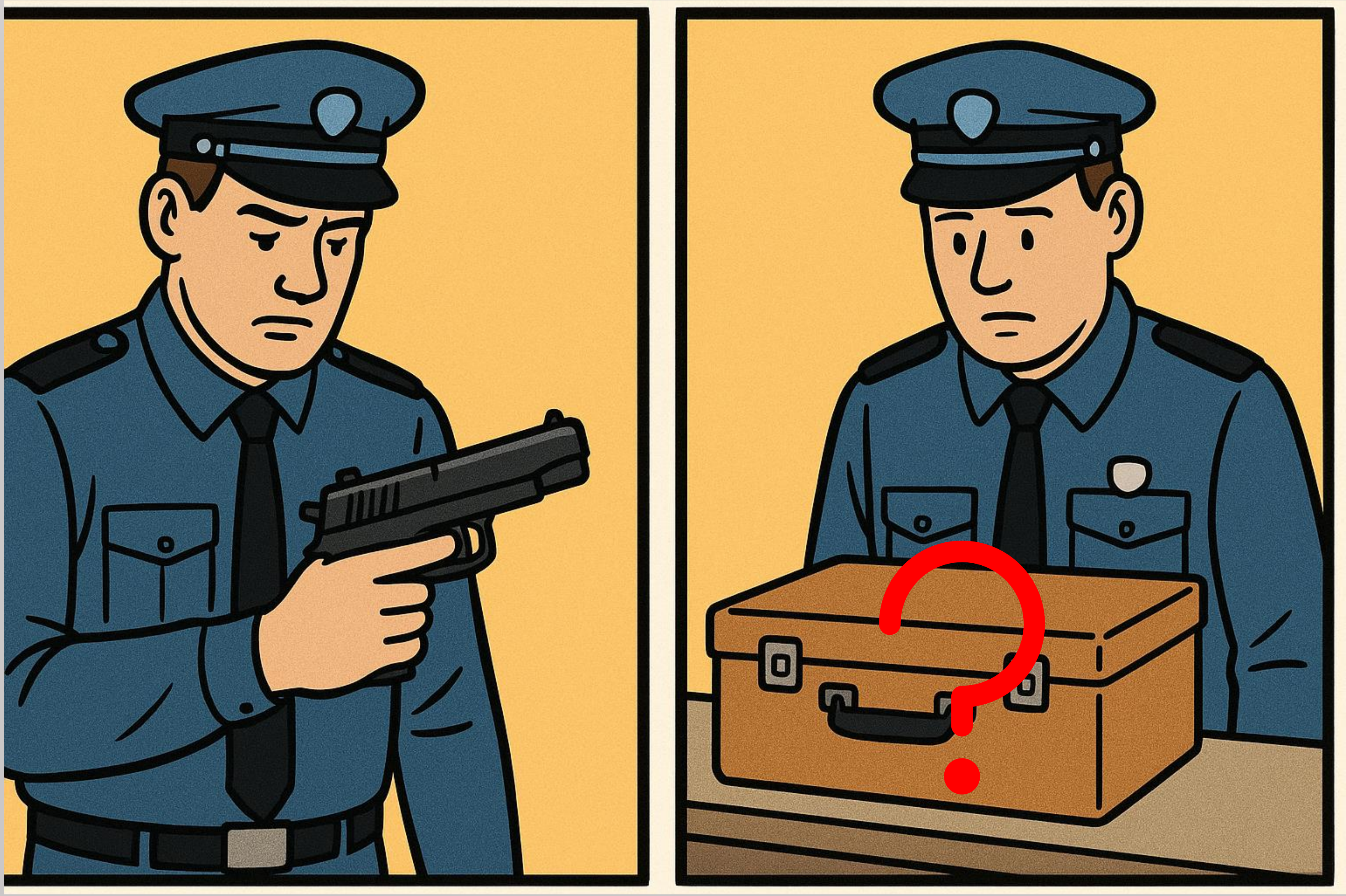
Shape from X

32 Views

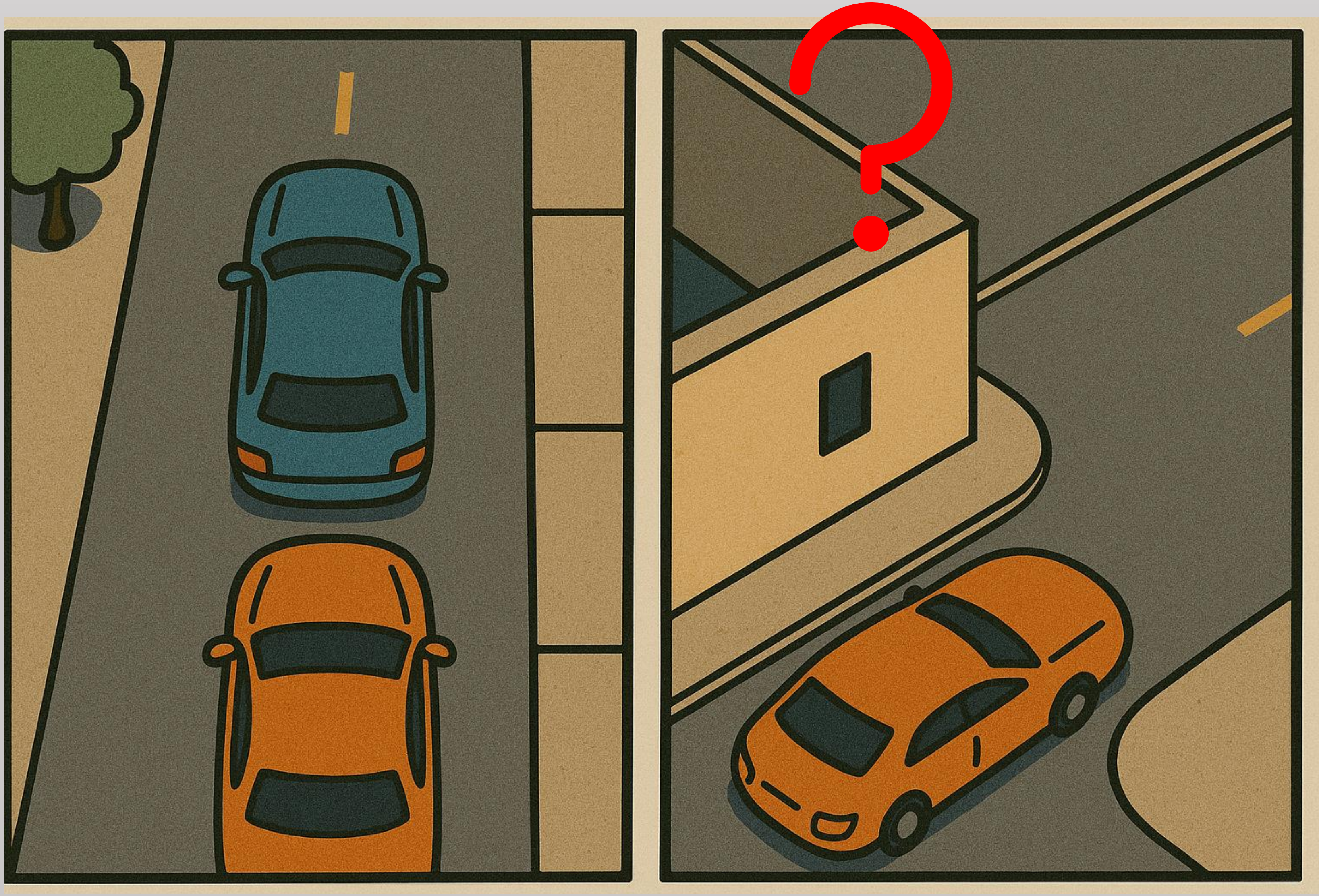


X is Visible Light

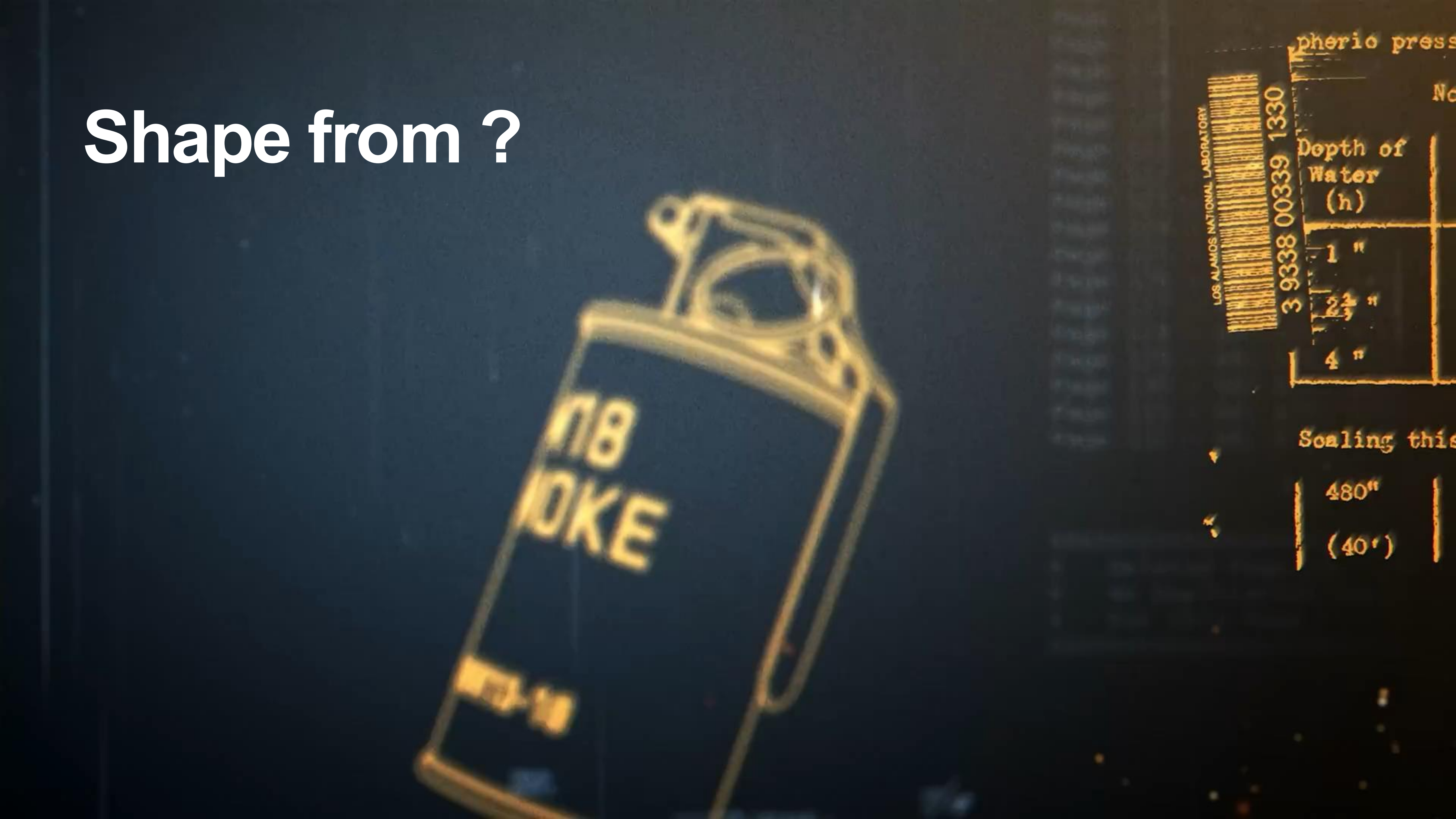
Shape from ?



Shape from ?



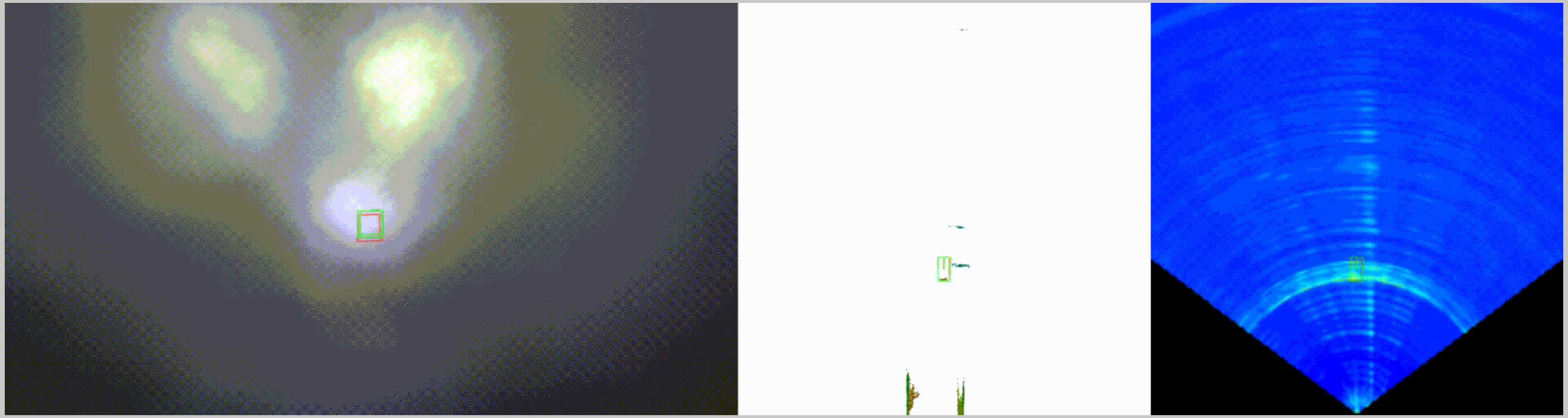
Shape from ?

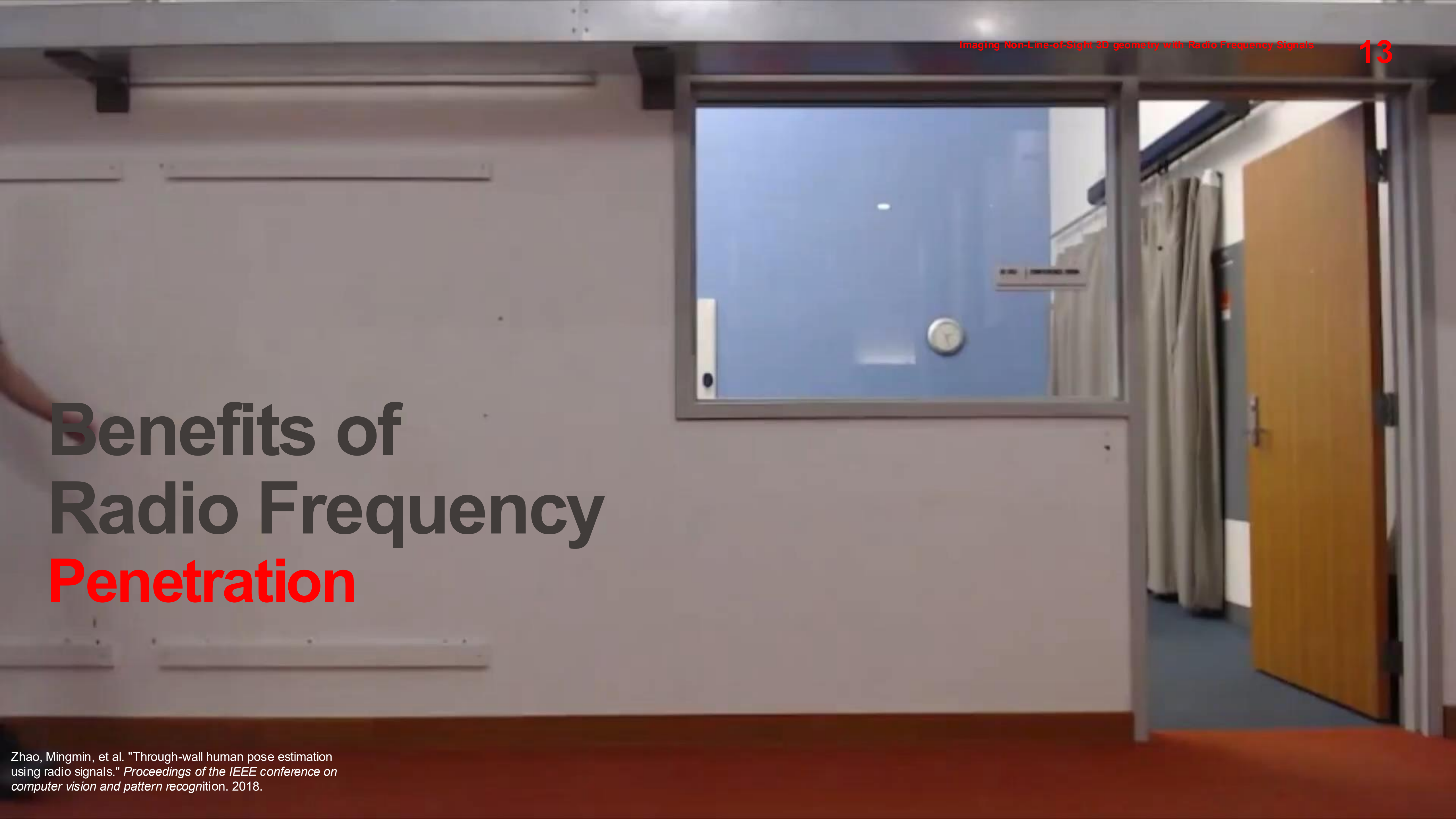


LOS ALAMOS NATIONAL LABORATORY
3 9338 00339 1330
Depth of Water (h)
1 "
4 "
Scaling this
480"
(40')

X is Radio Frequency

Benefits of Radio Frequency
Penetration

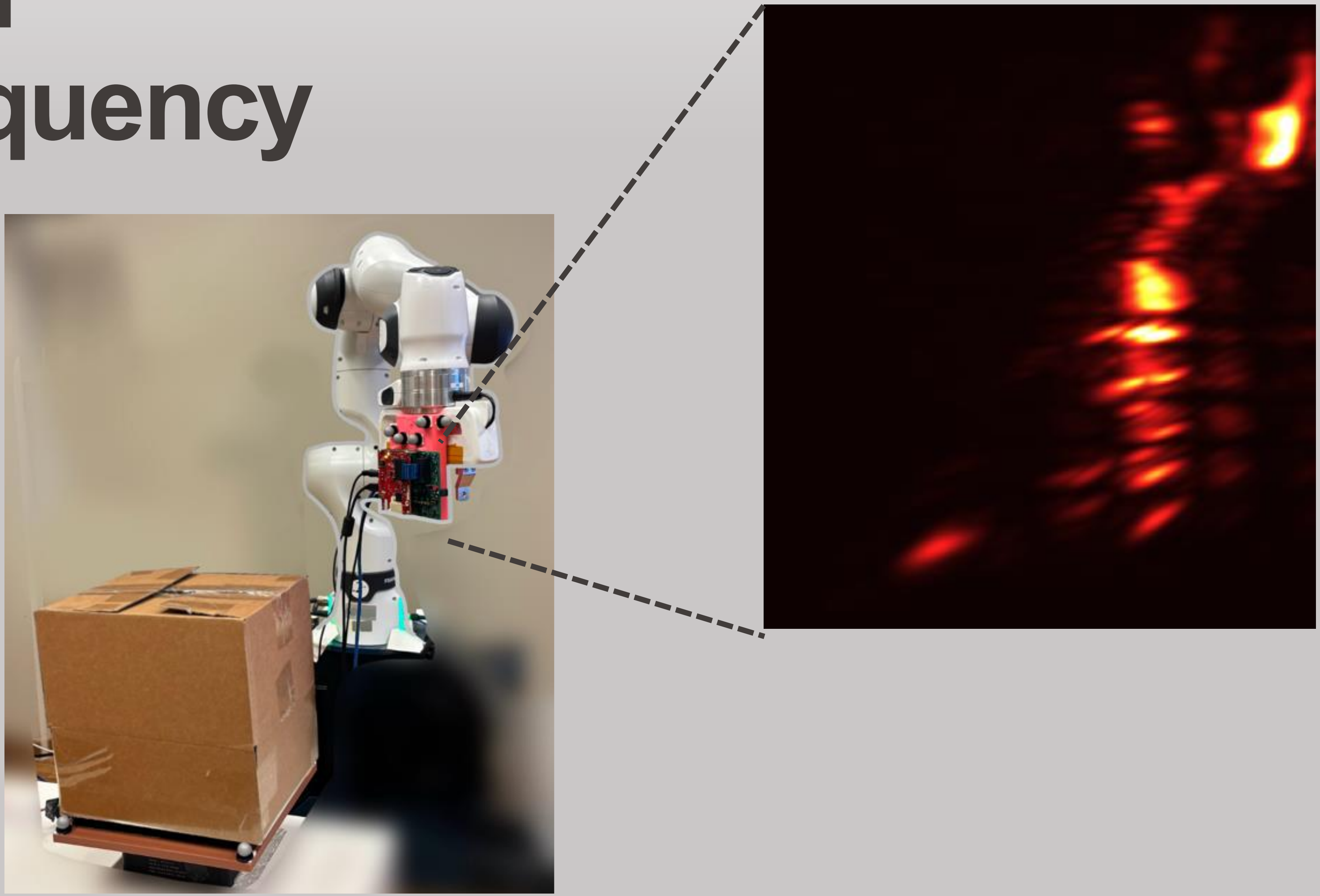




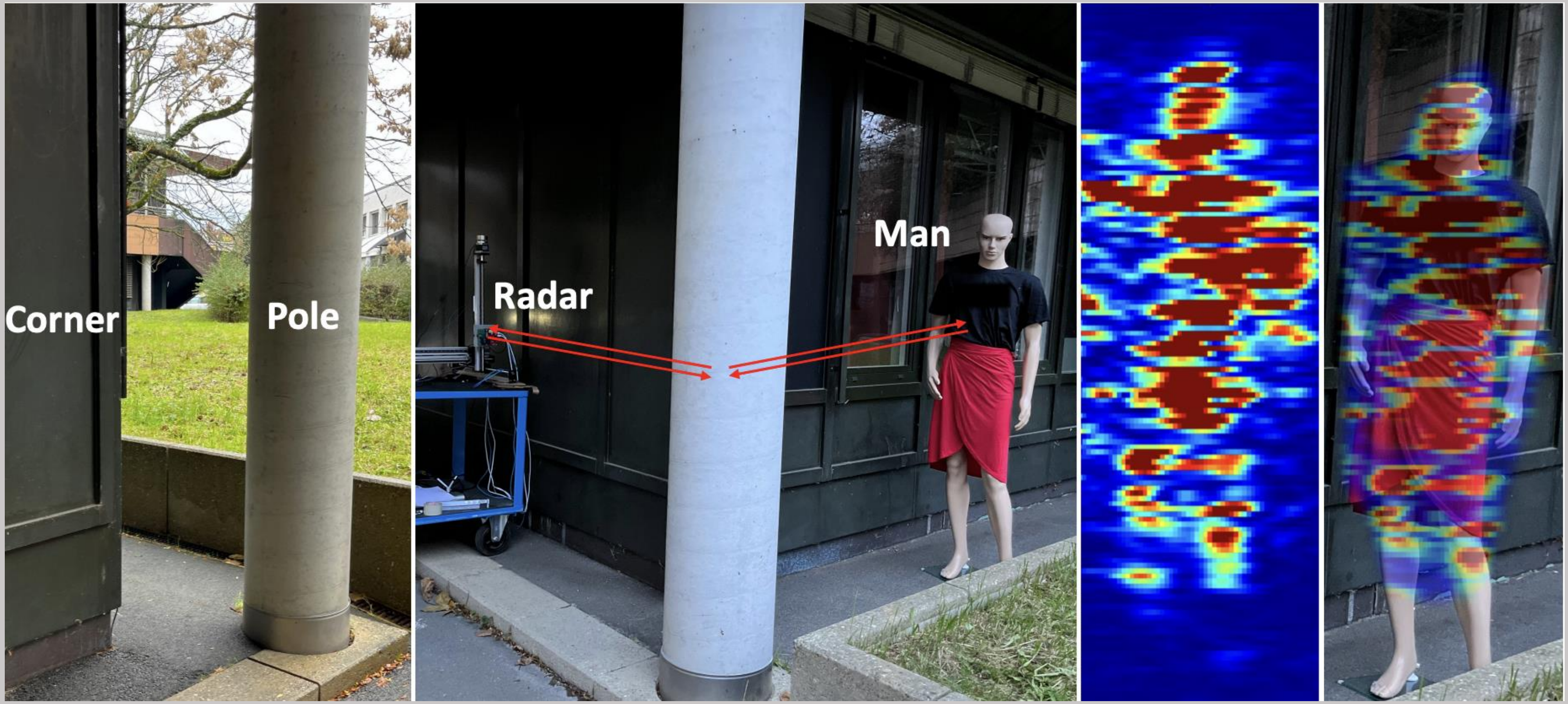
Benefits of Radio Frequency
Penetration

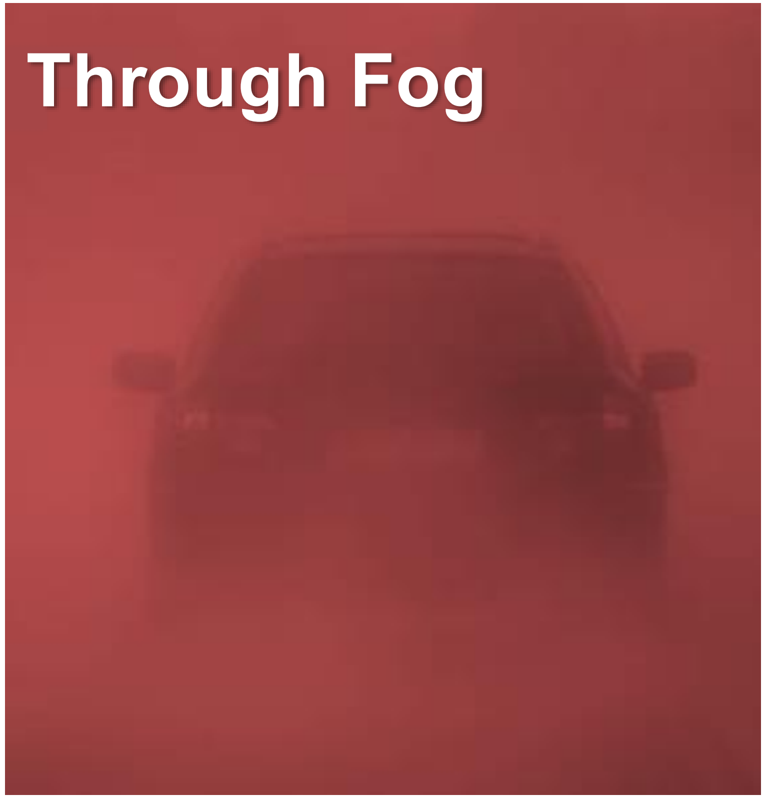
Zhao, Mingmin, et al. "Through-wall human pose estimation using radio signals." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

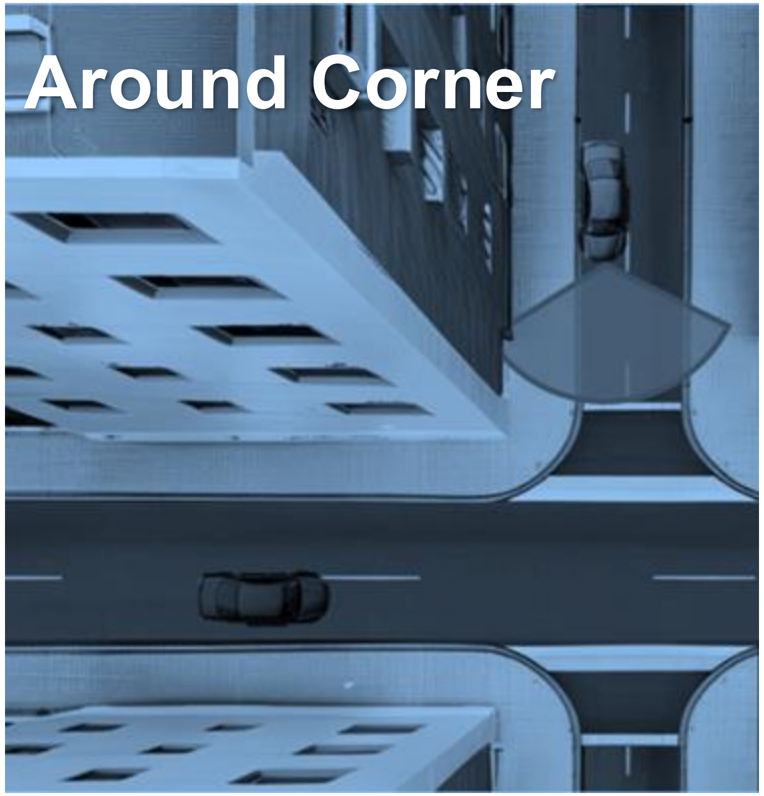
Benefits of Radio Frequency
Penetration



Benefits of Radio Frequency
Reflection







Non-Line-of-Sight (NLoS)

using Radio Frequency Signals

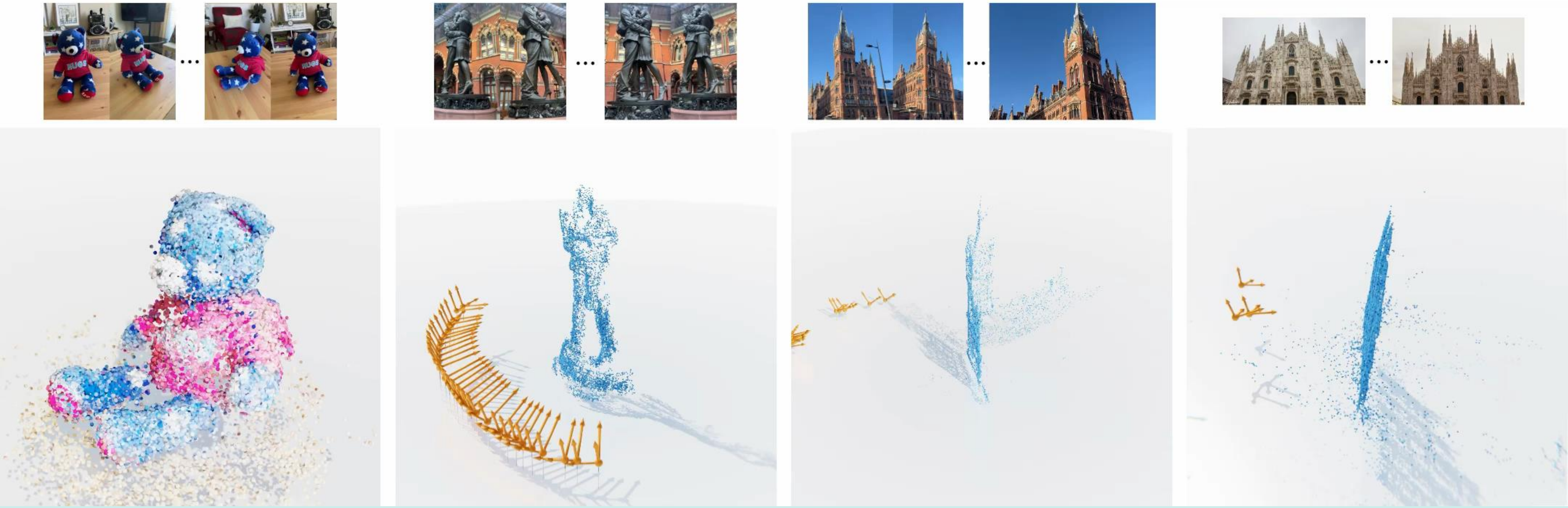
Can we reconstruct high resolution shape from radio Frequency Signals?

Two steps: from Light to RF

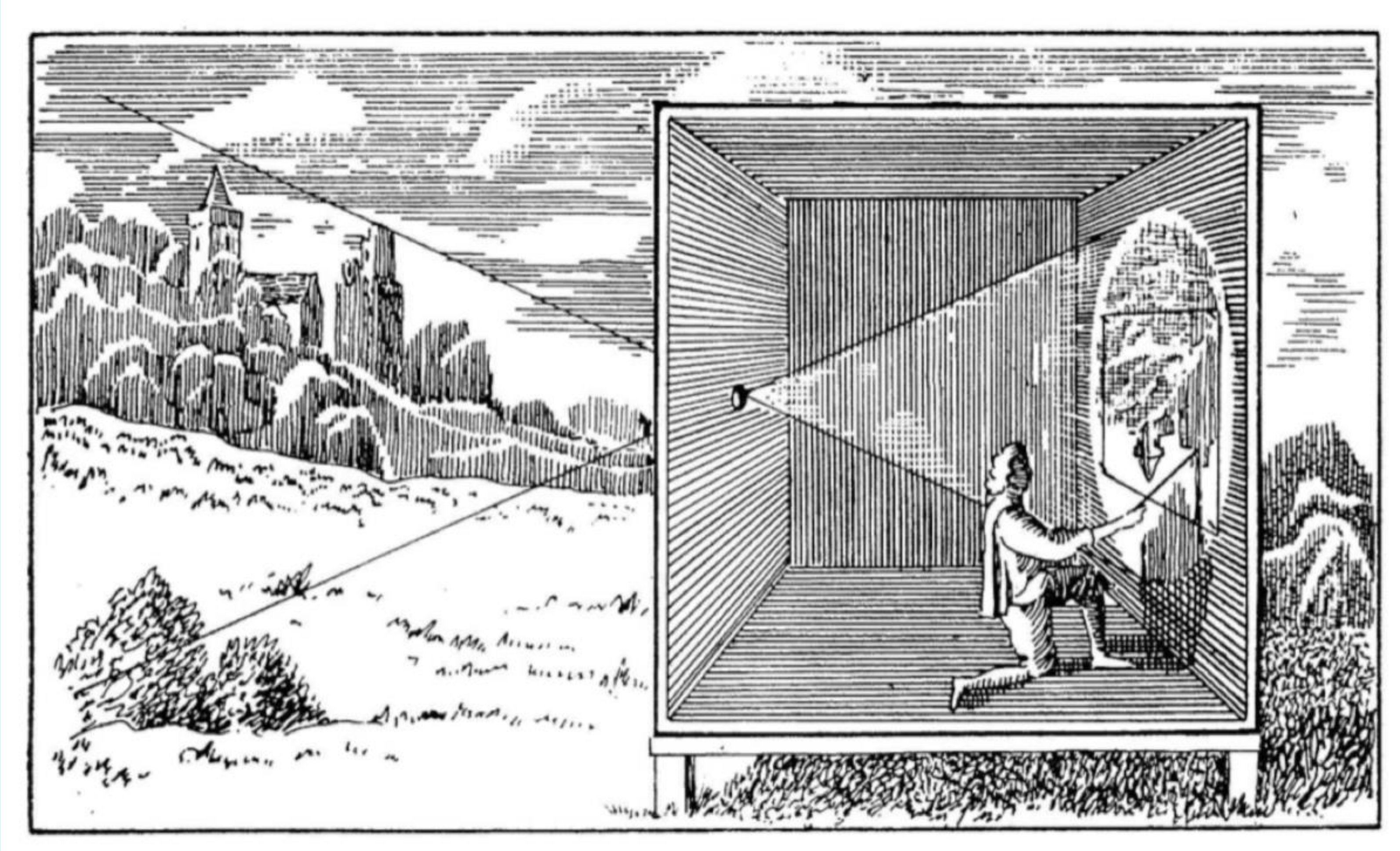
How to reconstruct surfaces from visible light?

How to reconstruct surfaces from RF signals?

What is 3D Reconstruction? Sparse -> Density



How to get an Image

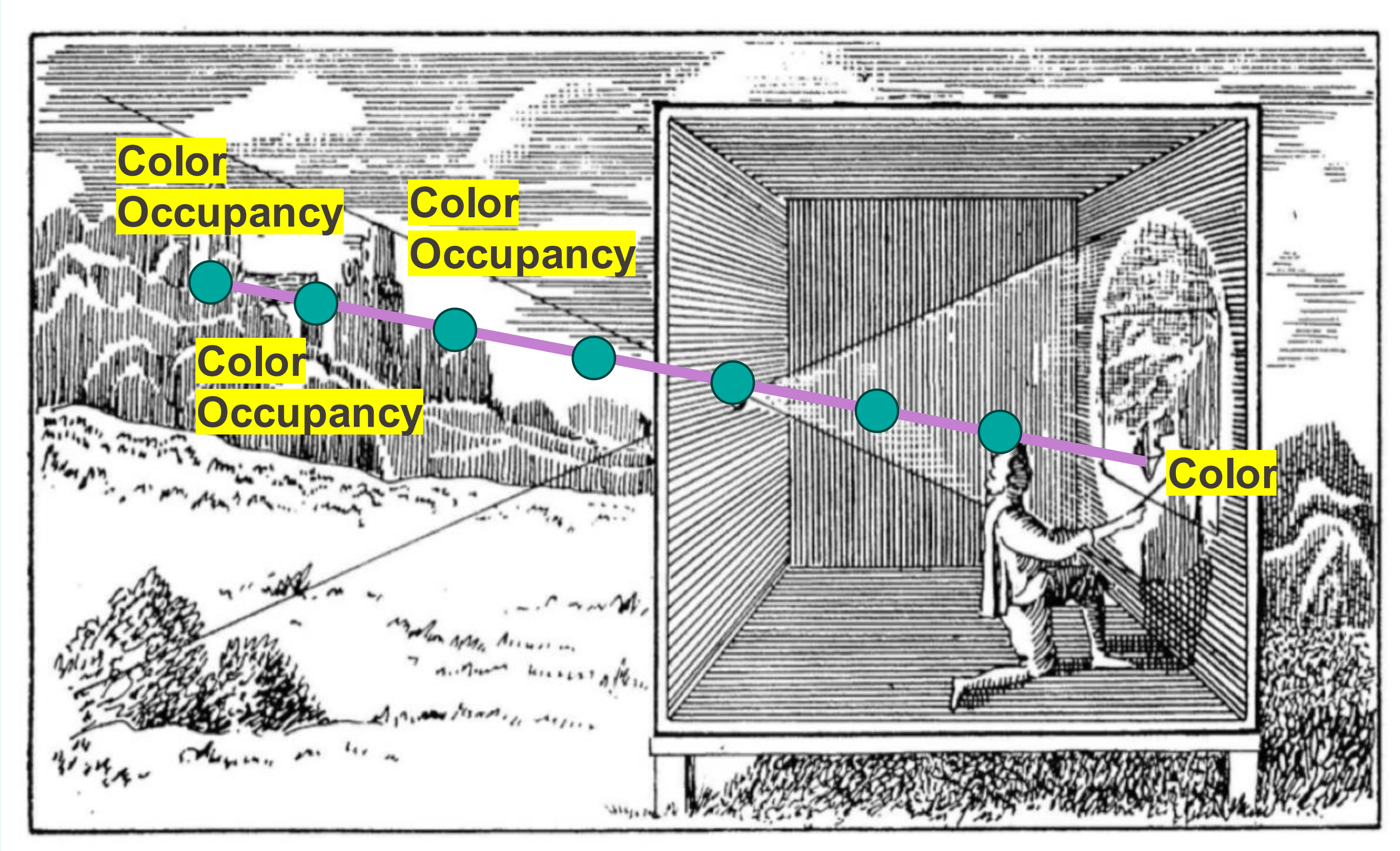


How to get an Image

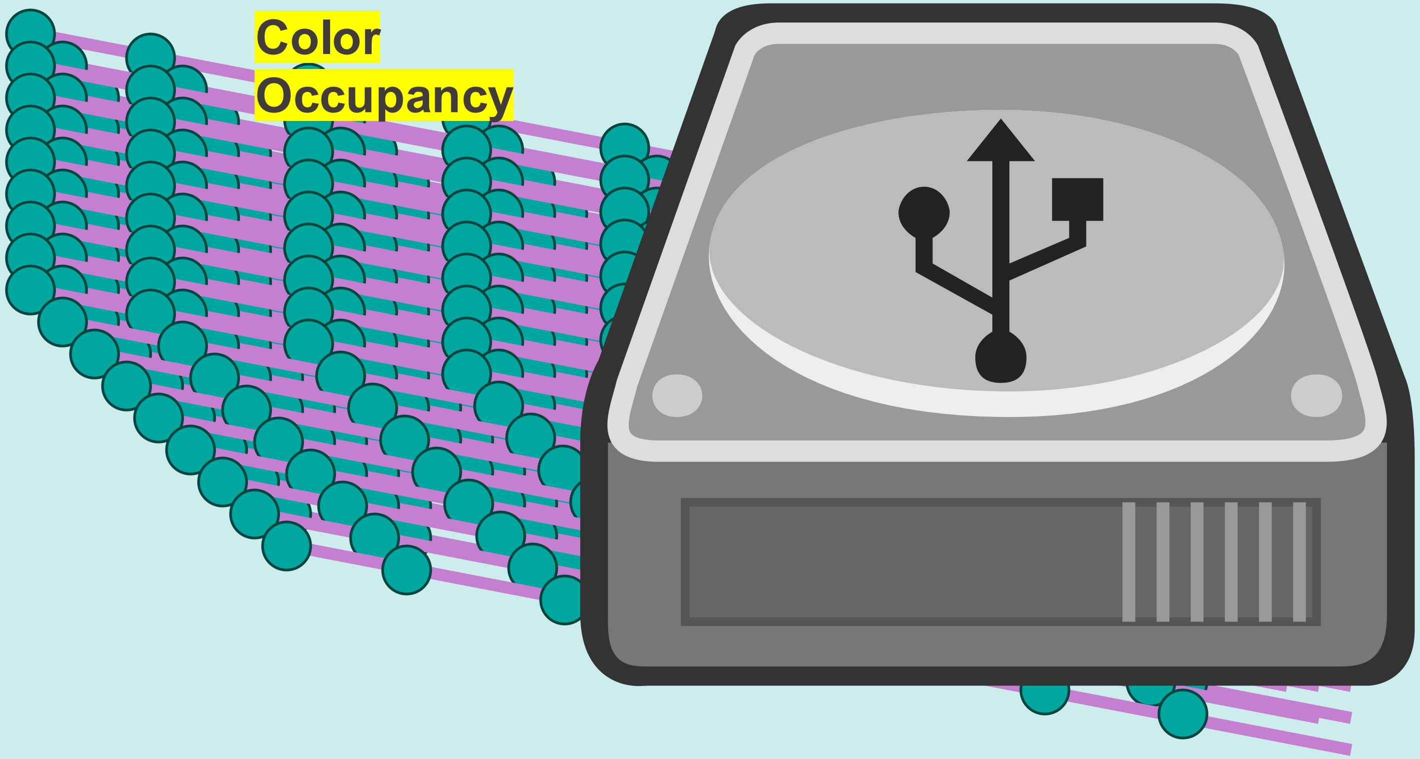
How to get an Image

How to get an Image

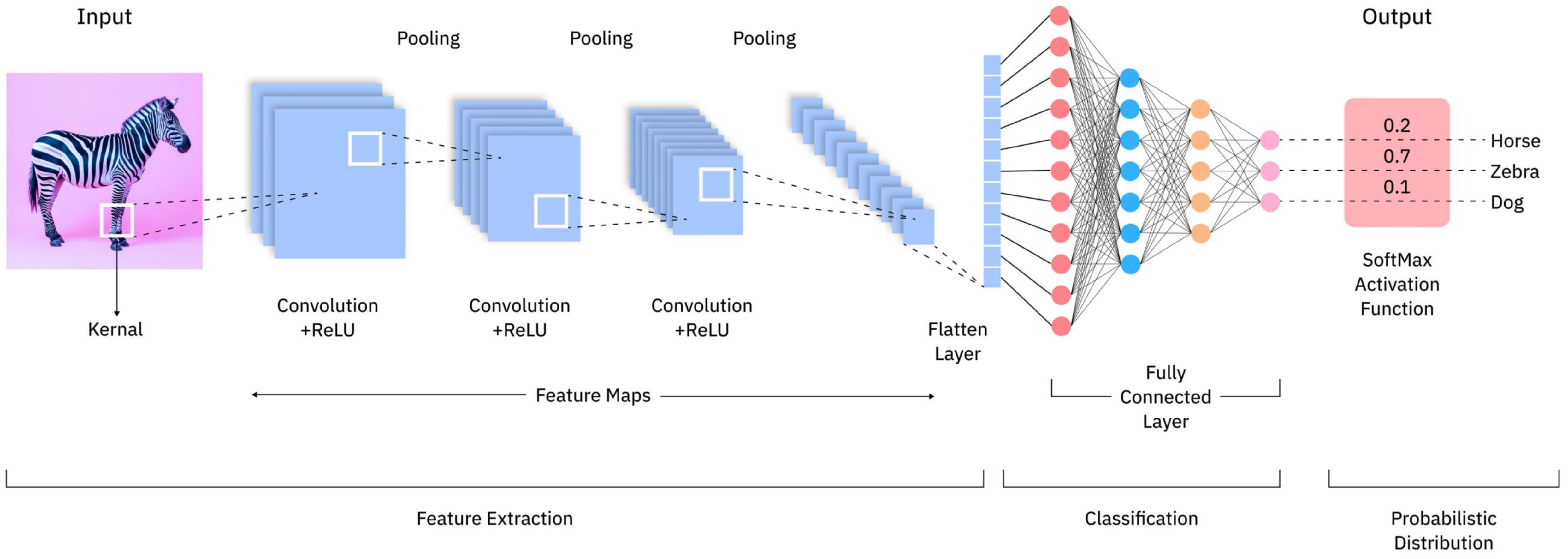
How to get an Image



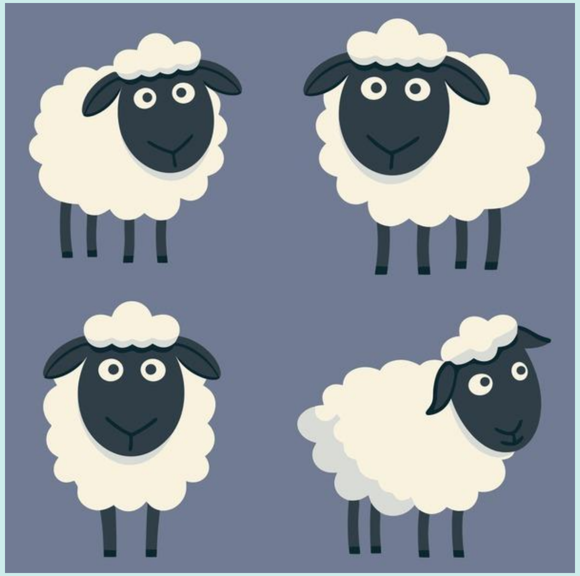
If you can **remember** occupancy for all points, you can do reconstruction.



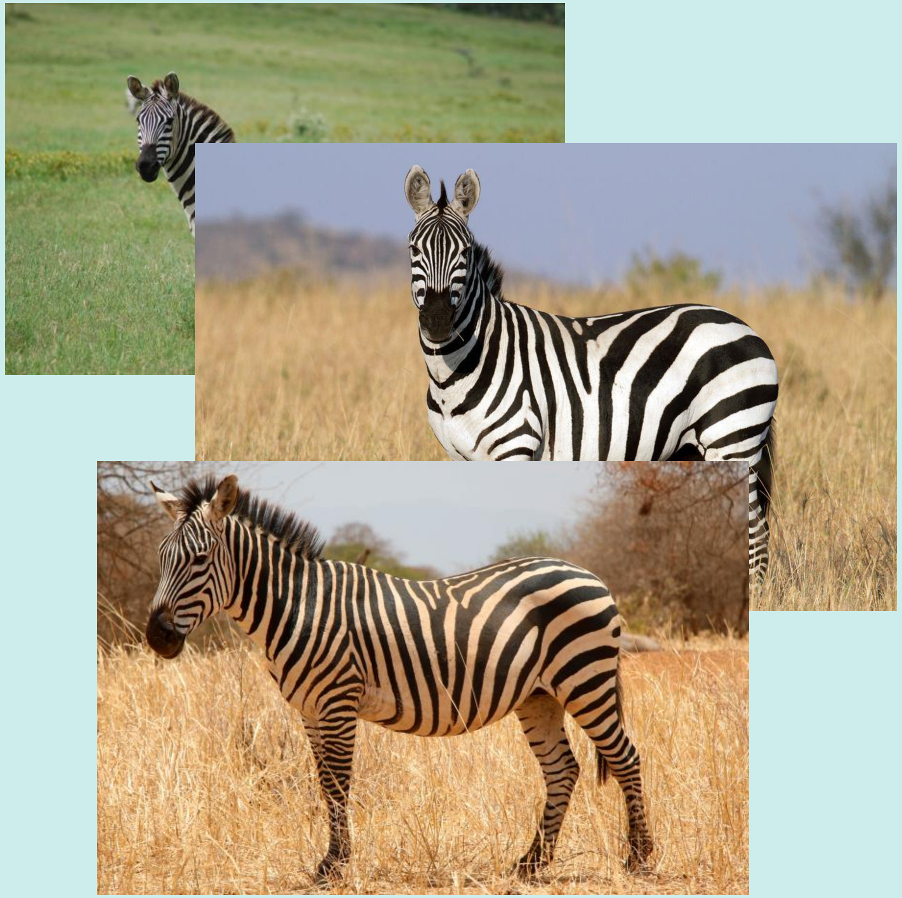
Neural Network

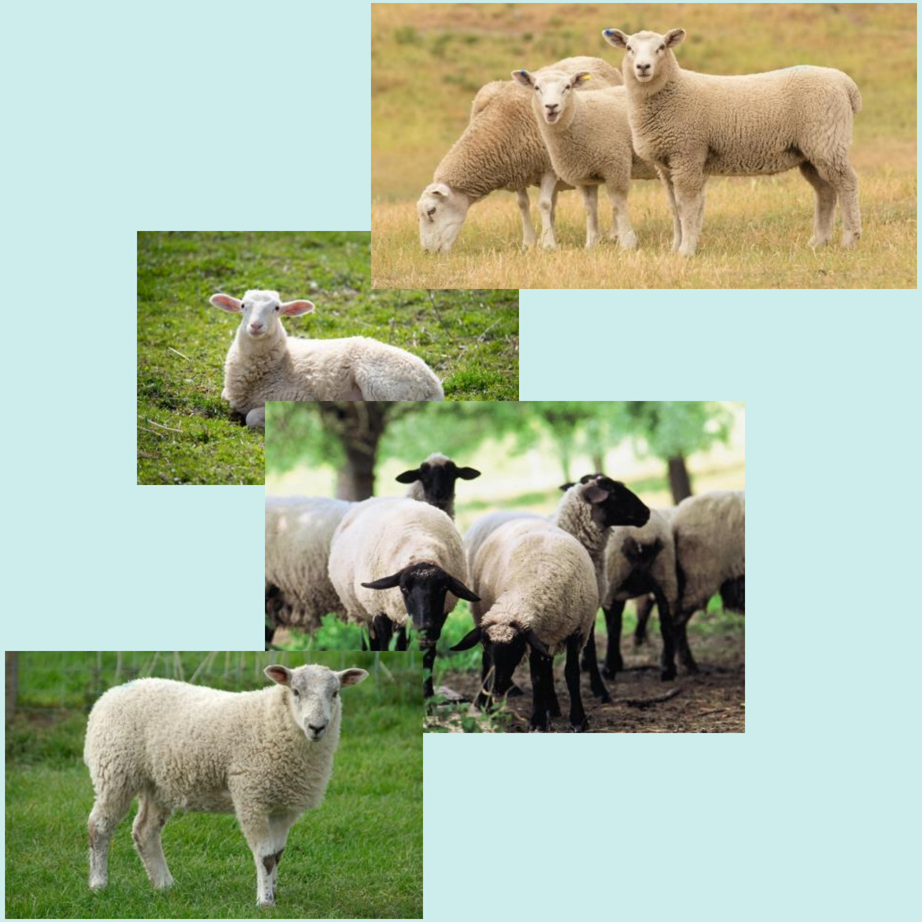


Neural Network is a General Memory



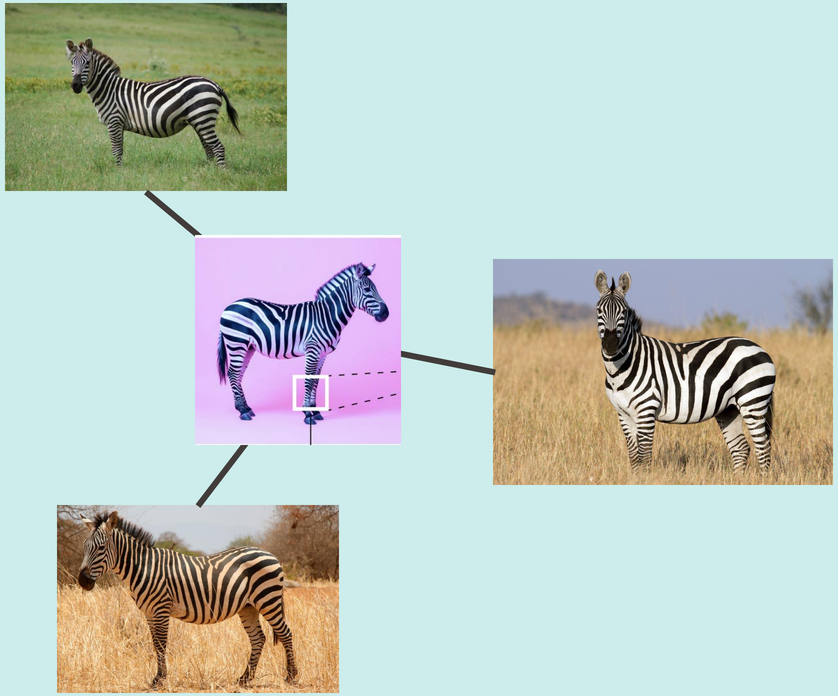
Neural Network is a General Memory



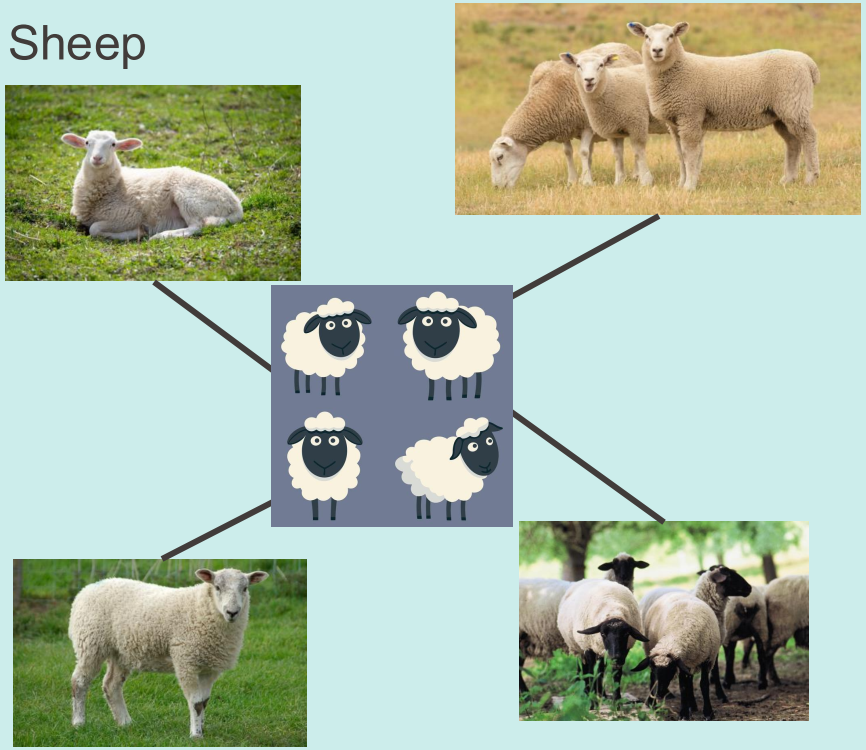


Neural Network is a General Memory

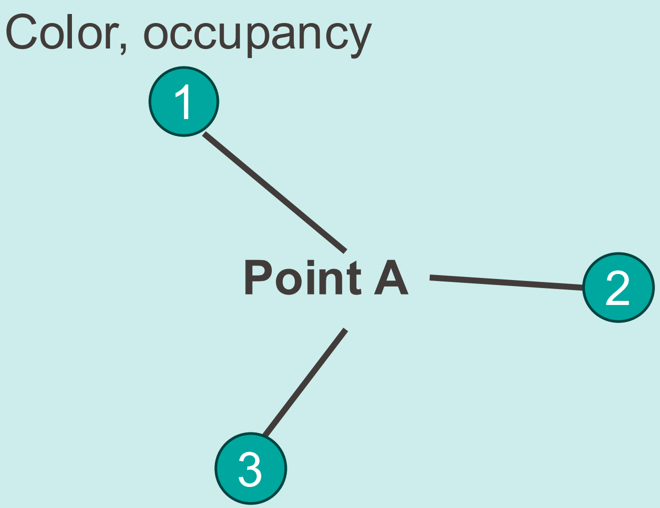
Zebra

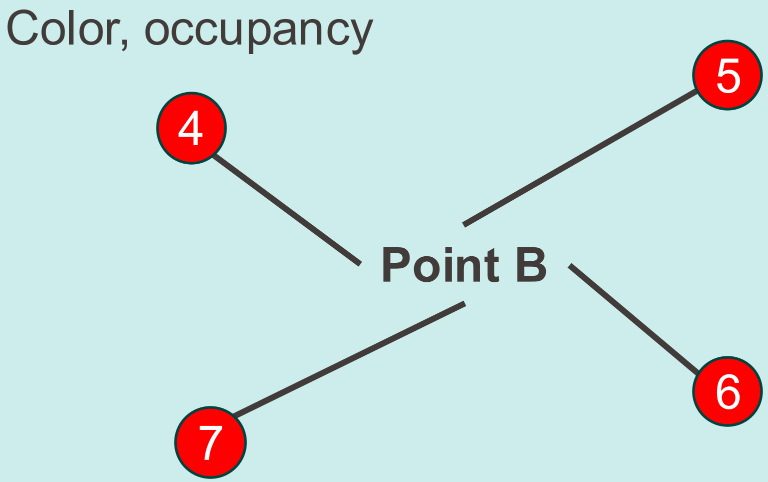


Sheep



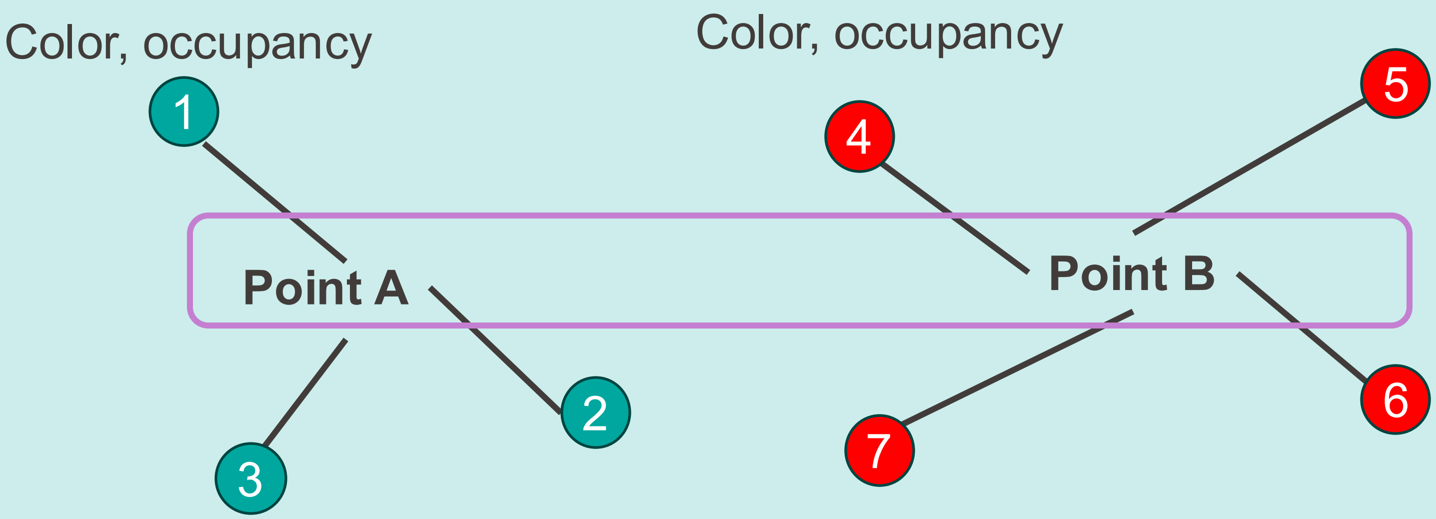
Neural Network is a General Memory





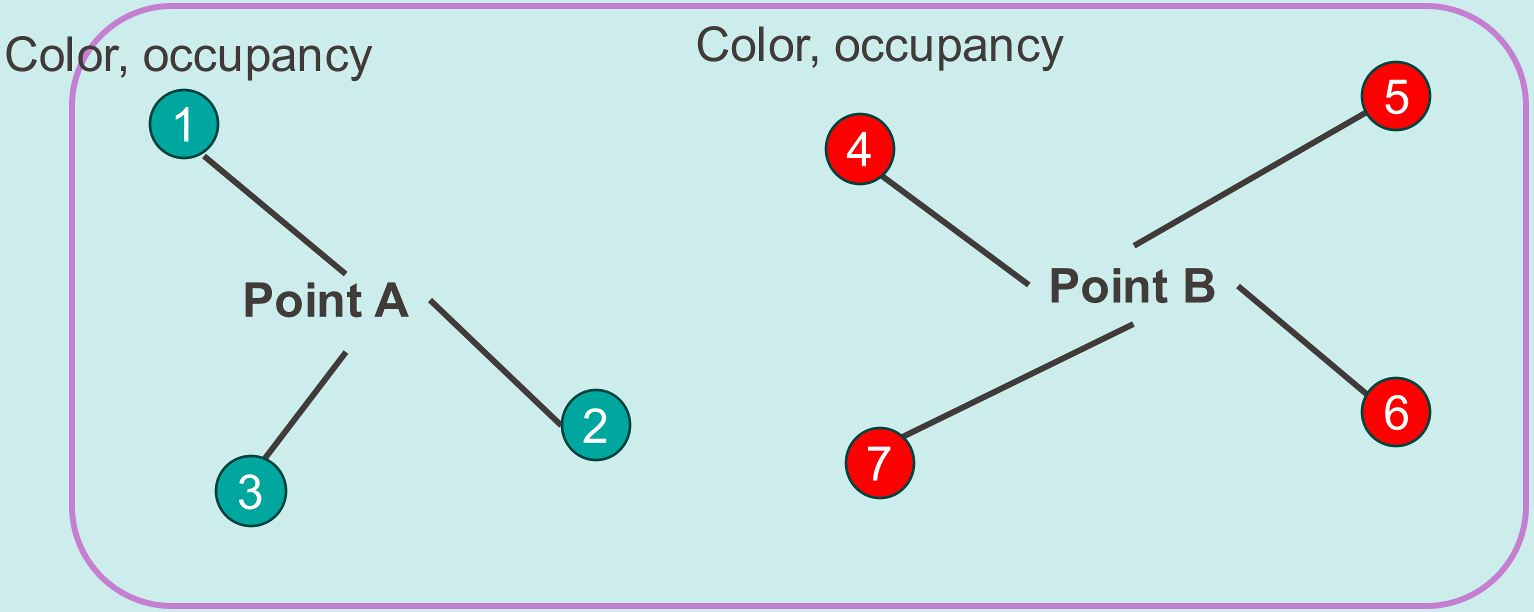
If you can remember all the zebras from few images, you can remember all the 3D points from few points.

Neural Radiance Fields (NeRF)



Training Set

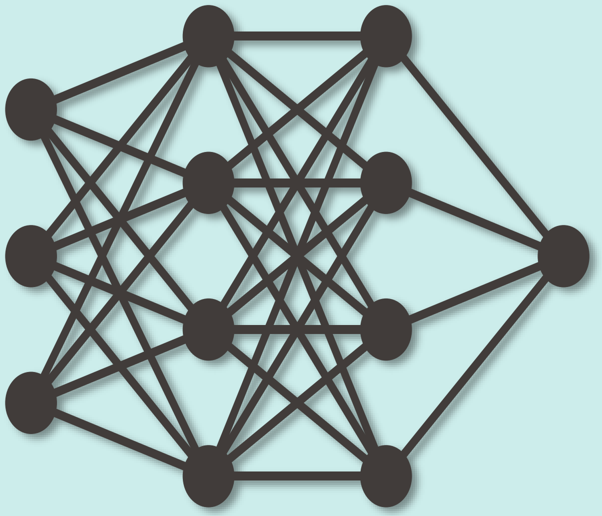
Neural Radiance Fields (NeRF)



Test Set

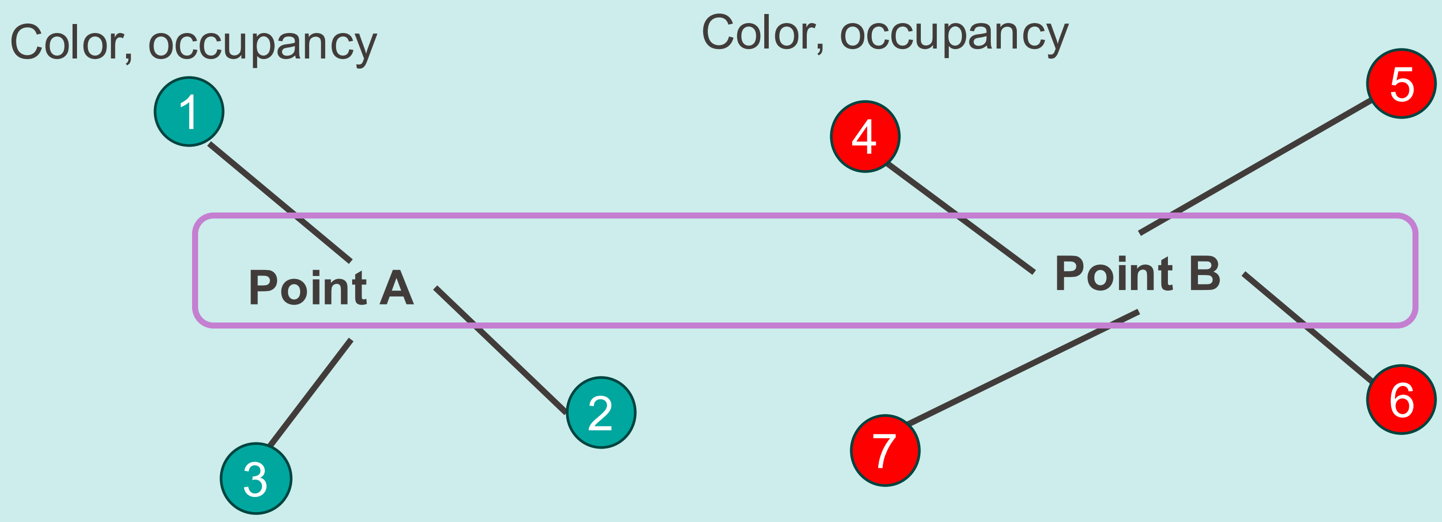
Neural Radiance Fields (NeRF)

Position of points



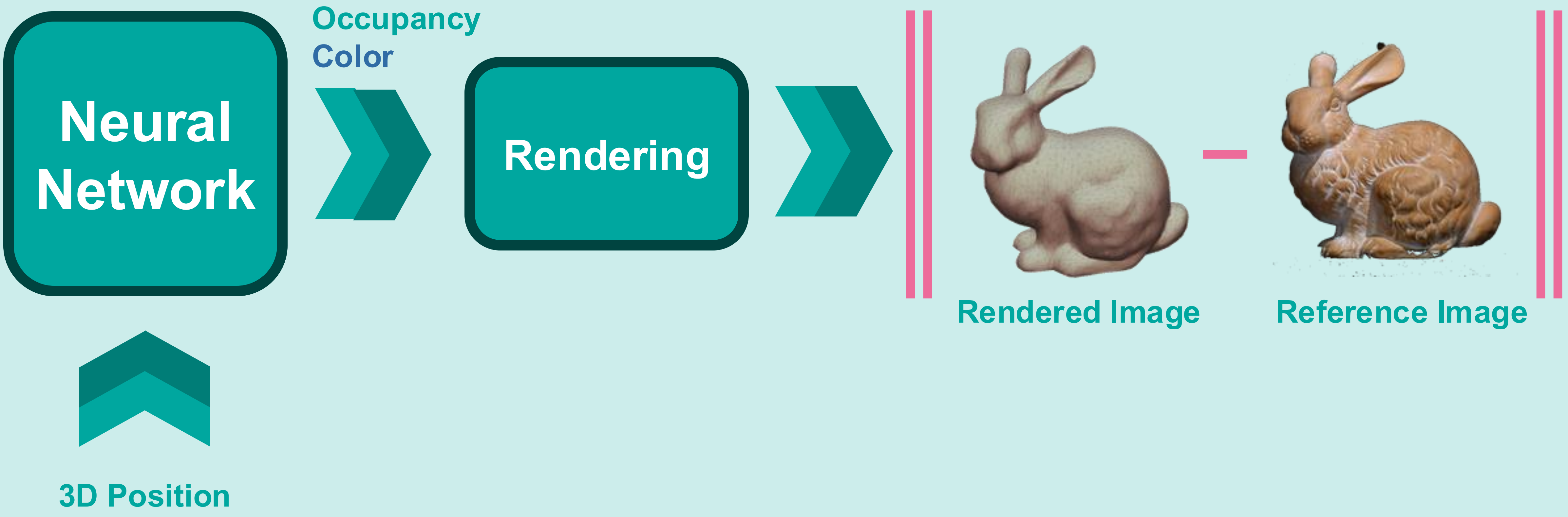
Color
Occupancy

No explicit color or occupancy for you to train.

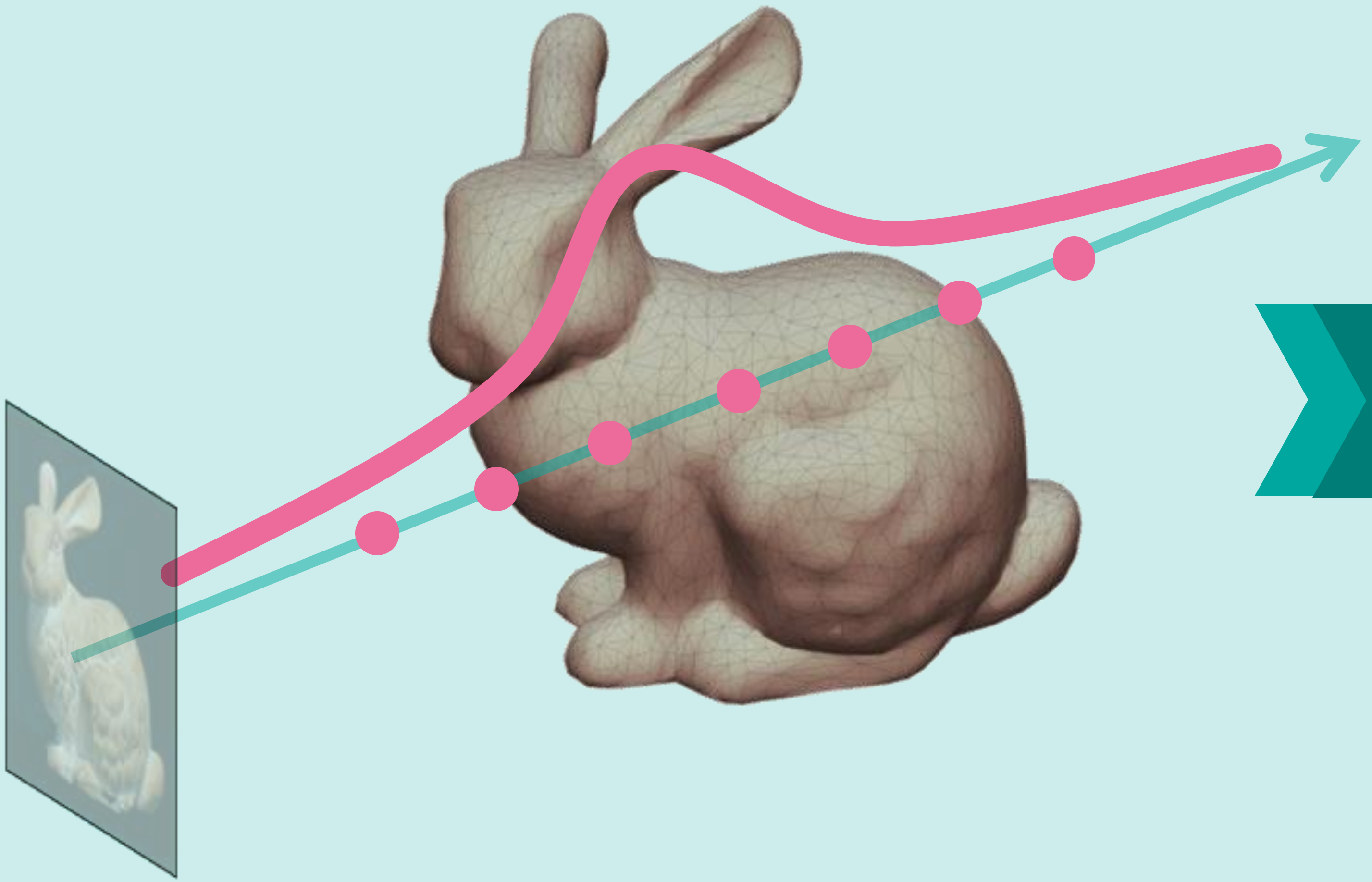


Training Set

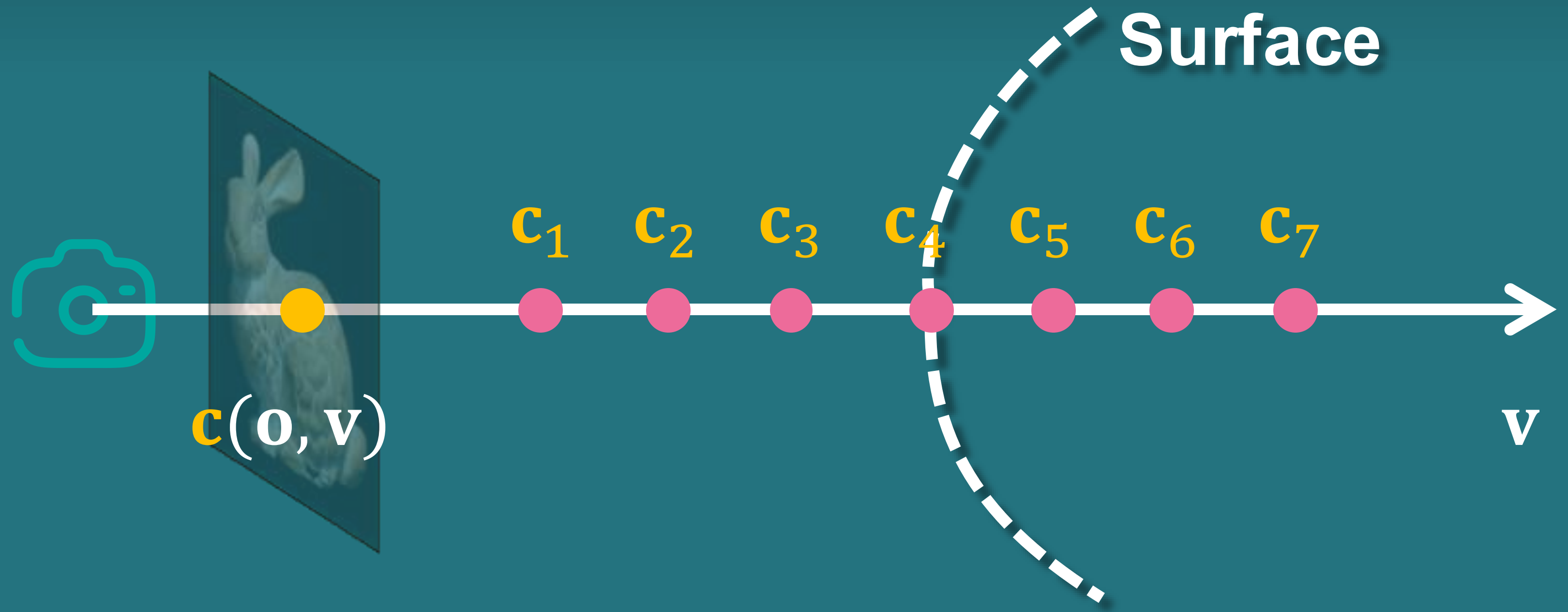
Neural Implicit 3D Reconstruction



How to Render 2D from 3D?



Embedded Color



$$\mathbf{c}(\mathbf{o},\mathbf{v}) = \sum_{i=1}^{n} w_i \mathbf{c}_i$$

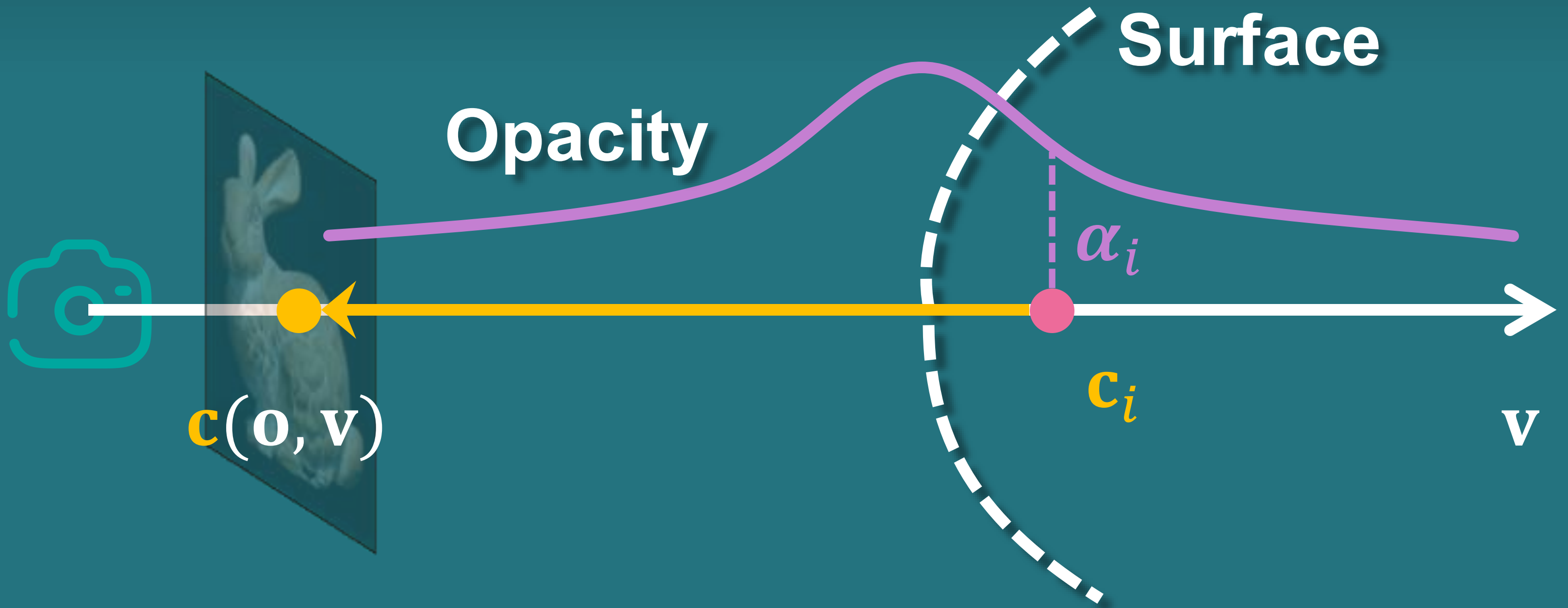
$\mathbf{o}$: Camera Center

$\mathbf{v}$: Ray Direction

$\mathbf{c}$: Color

w: Weight

Opacity

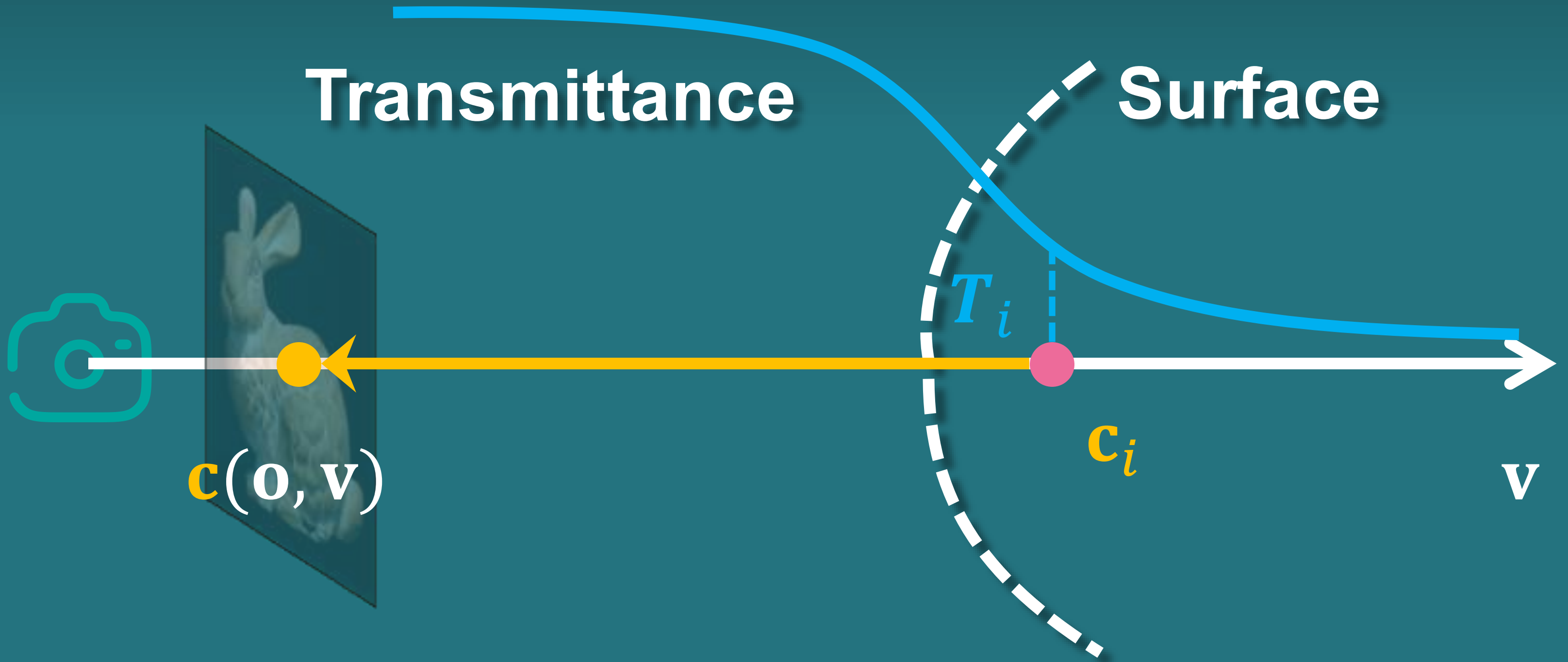


$$\mathbf{c}(\mathbf{o}, \mathbf{v}) = \sum_{i=1}^{n} w_i \mathbf{c}_i$$

*how much light **absorbed** at that point.*

$\mathbf{c}$: Color
w: Weight
α: Opacity

Accumulated Transmittance



$$\mathbf{c}(\mathbf{o}, \mathbf{v}) = \sum_{i=1}^{n} w_i \mathbf{c}_i$$

how much light ***passes through*** *the scene before reaching the camera*

$\mathbf{c}$: Color

w: Weight

α: Opacity

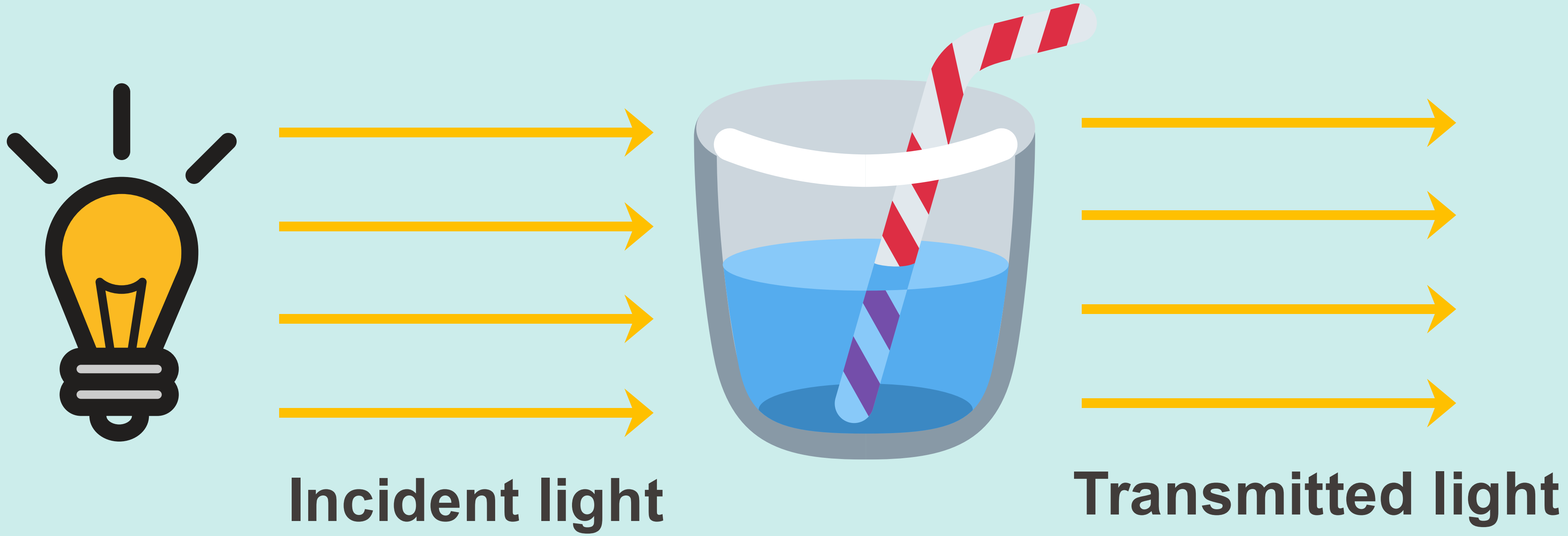
T: Accumulated Transmittance

Volumetric Rendering

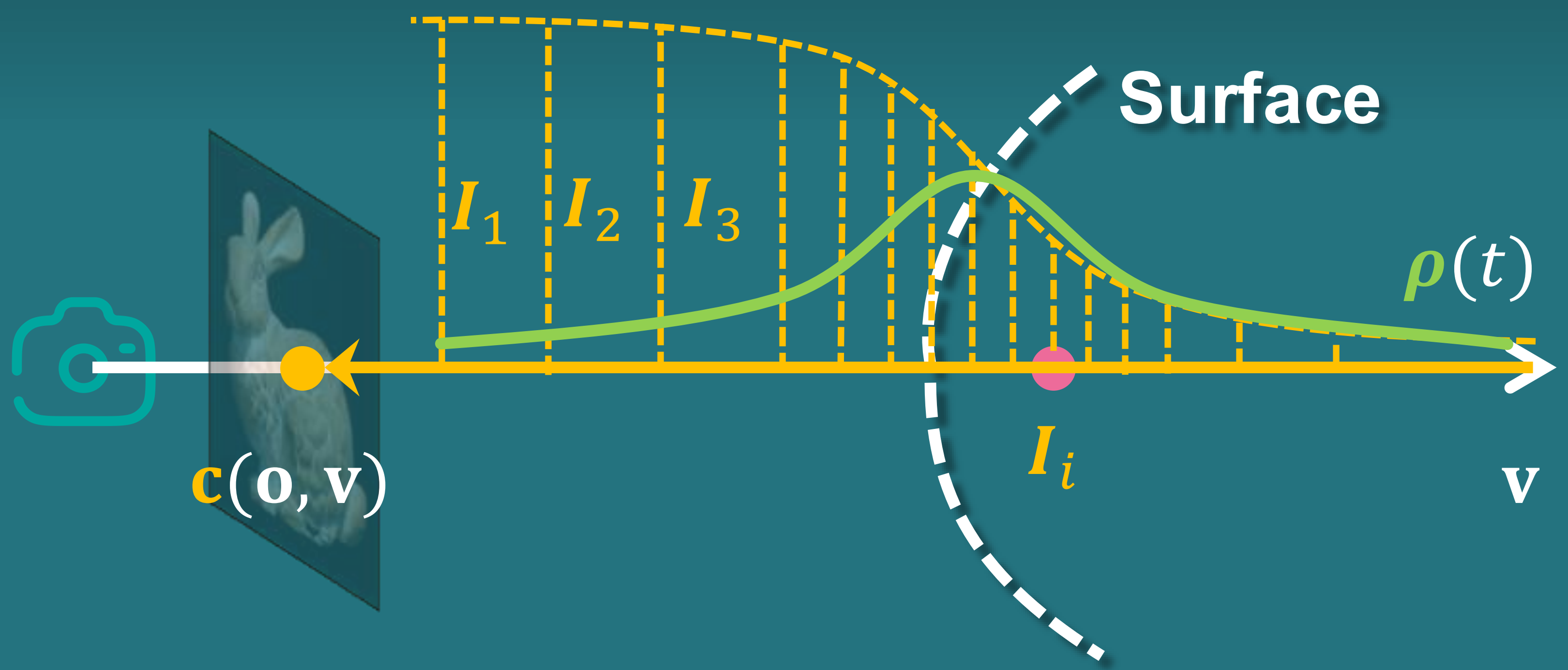
$$\mathbf{c}(\mathbf{o}, \mathbf{v}) = \sum_{i=1}^{n} w_i \mathbf{c}_i$$
$$= \sum_{i=1}^{n} T_i \alpha_i \mathbf{c}_i$$

$\mathbf{c}$: Color
w: Weight
α: Opacity
T: Accumulated Transmittance

From 3D Geometry to Light behavior



Bridge between 3D geometry and light



Beer–Lambert Law

$$\frac{d\boldsymbol{I}}{dt} = -\rho(t)\boldsymbol{I}(t)$$

ρ: Density

$\boldsymbol{I}(\mathbf{c})$: Light (Color)

t: Ray Parameter

Bridge between 3D geometry and light

Beer–Lambert Law

$$\frac{d\boldsymbol{I}}{dt} = -\boldsymbol{\rho}(t)\boldsymbol{I}(t)$$

$$\frac{dI}{I} = -\rho\, dt$$

$$\int \frac{1}{I} dI = -\int \rho dt$$

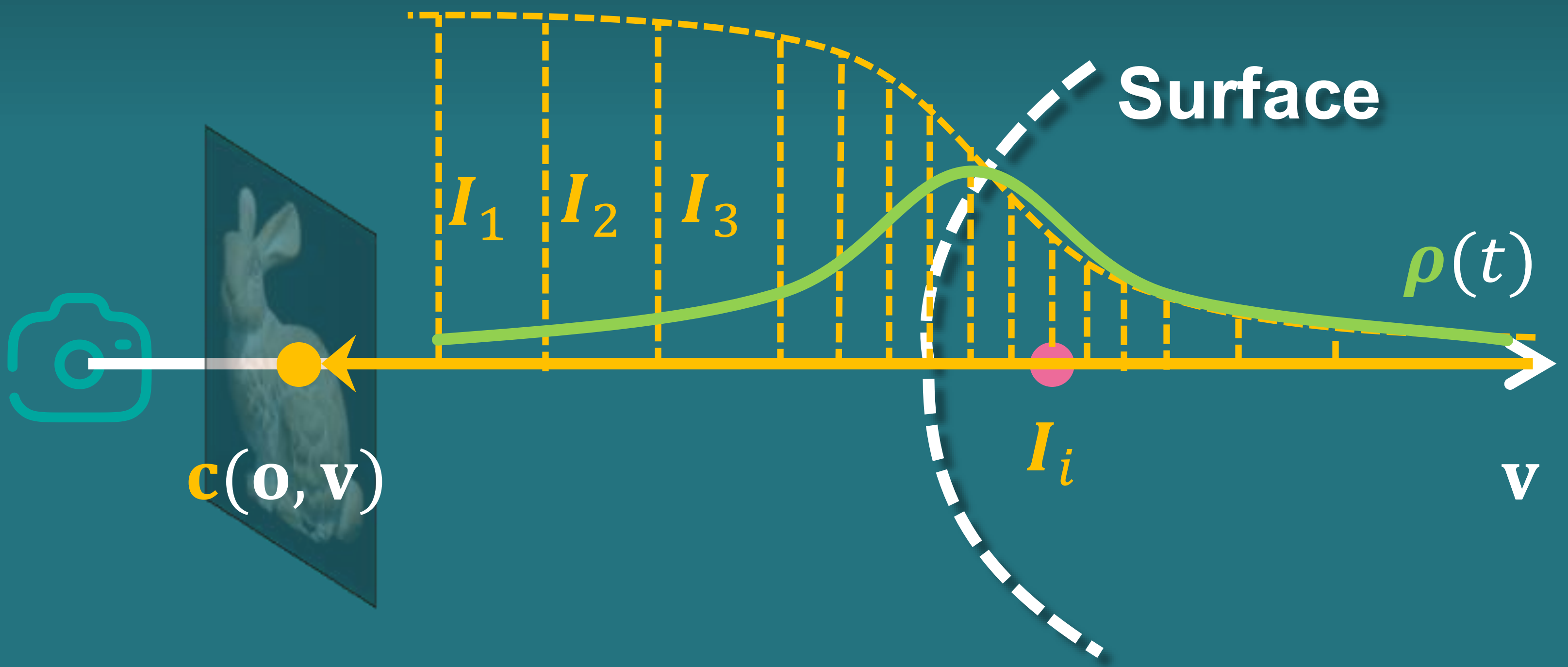
$$\ln I(t_{i+1}) - \ln I(t_i) = -\int_{t_i}^{t_{i+1}} \rho dt$$

$\boldsymbol{\rho}$: Density

$\boldsymbol{I}(\mathbf{c})$: Light (Color)

t: Ray Parameter

Beer–Lambert Law



Solve 1st ODE

$$I(t_{i+1}) = I(t_i)\exp\left(-\int_{t_i}^{t_{i+1}} \rho(t)\,\mathrm{d}t\right)$$

ΔT_i

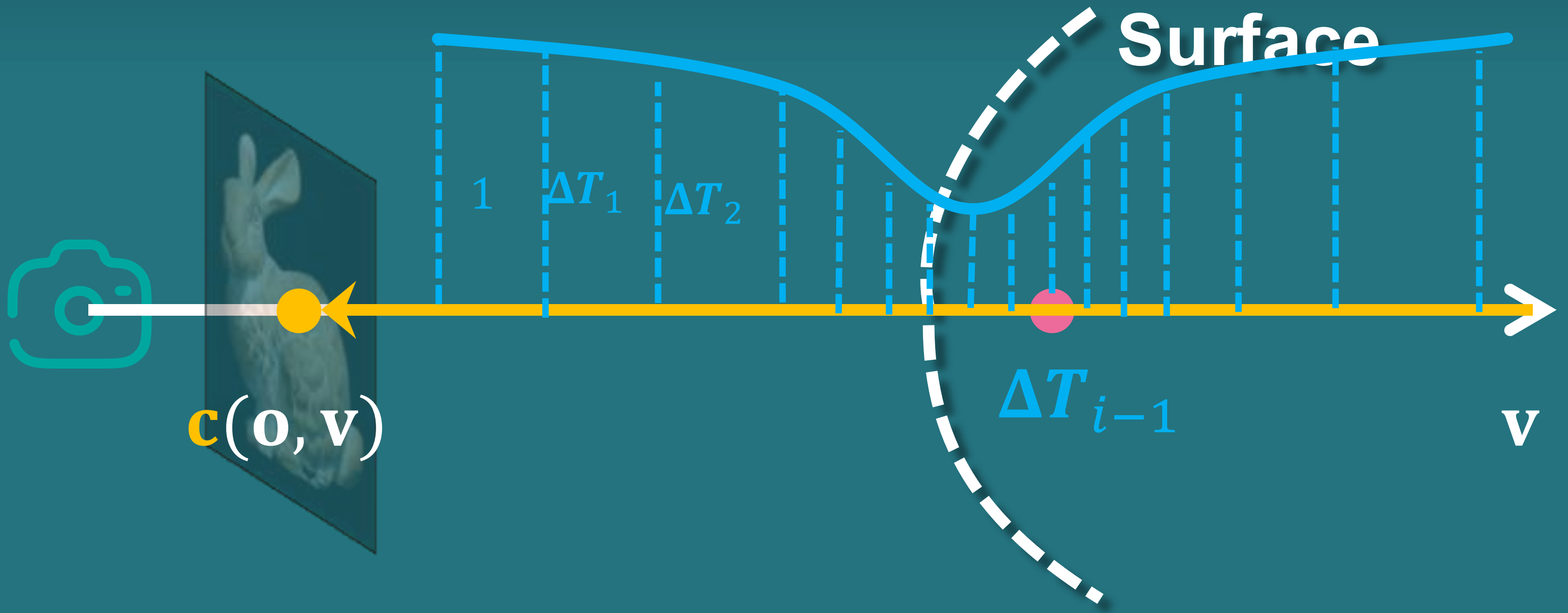
ρ: Density

$I(\mathbf{c})$: Light (Color)

t: Ray Parameter

ΔT: Transmittance

Transmittance



$$I_{i+1} = I_i \Delta T_i$$

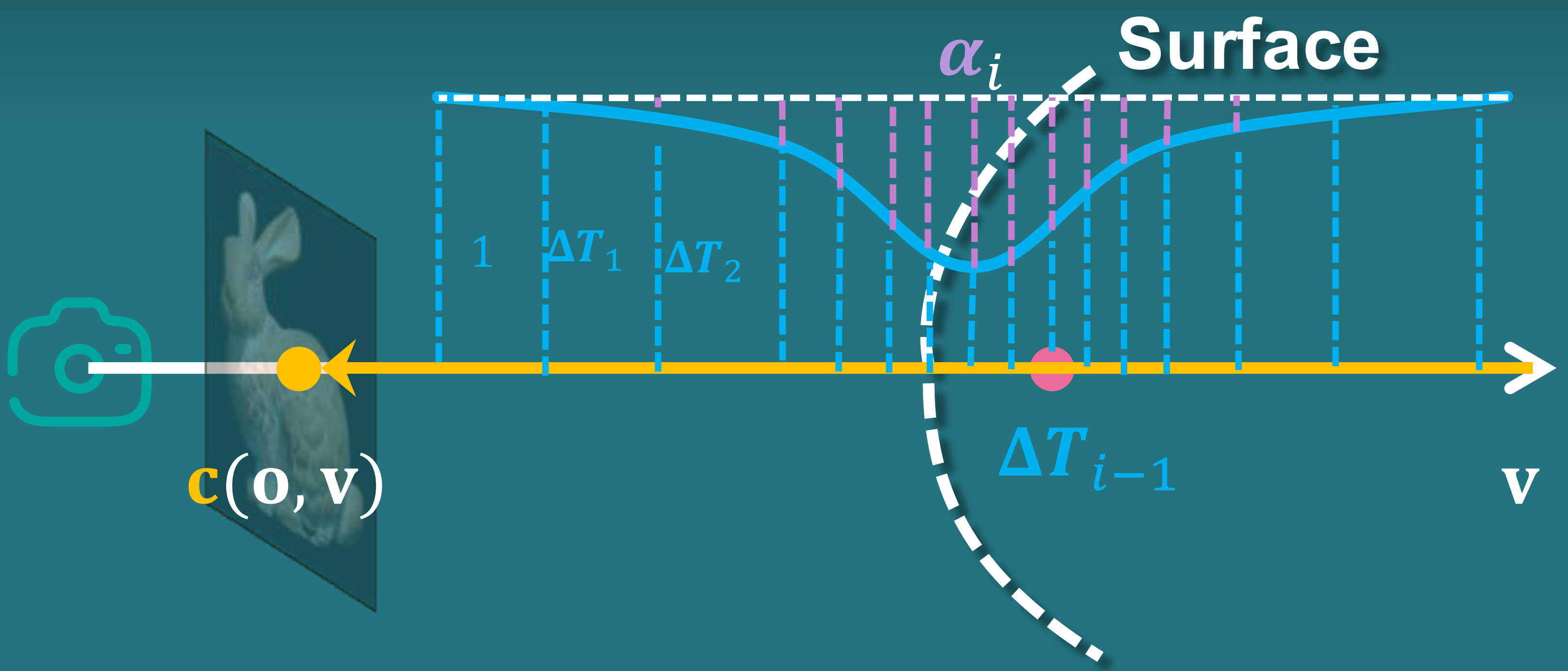
ρ: Density

$I(\mathbf{c})$: Light (Color)

t: Ray Parameter

ΔT: Transmittance

Opacity



$$I_{i+1} = I_i \Delta T_i$$

$$\alpha_i = 1 - \Delta T_i = 1 - \exp(-\rho_i \delta t_i)$$

ρ: Density

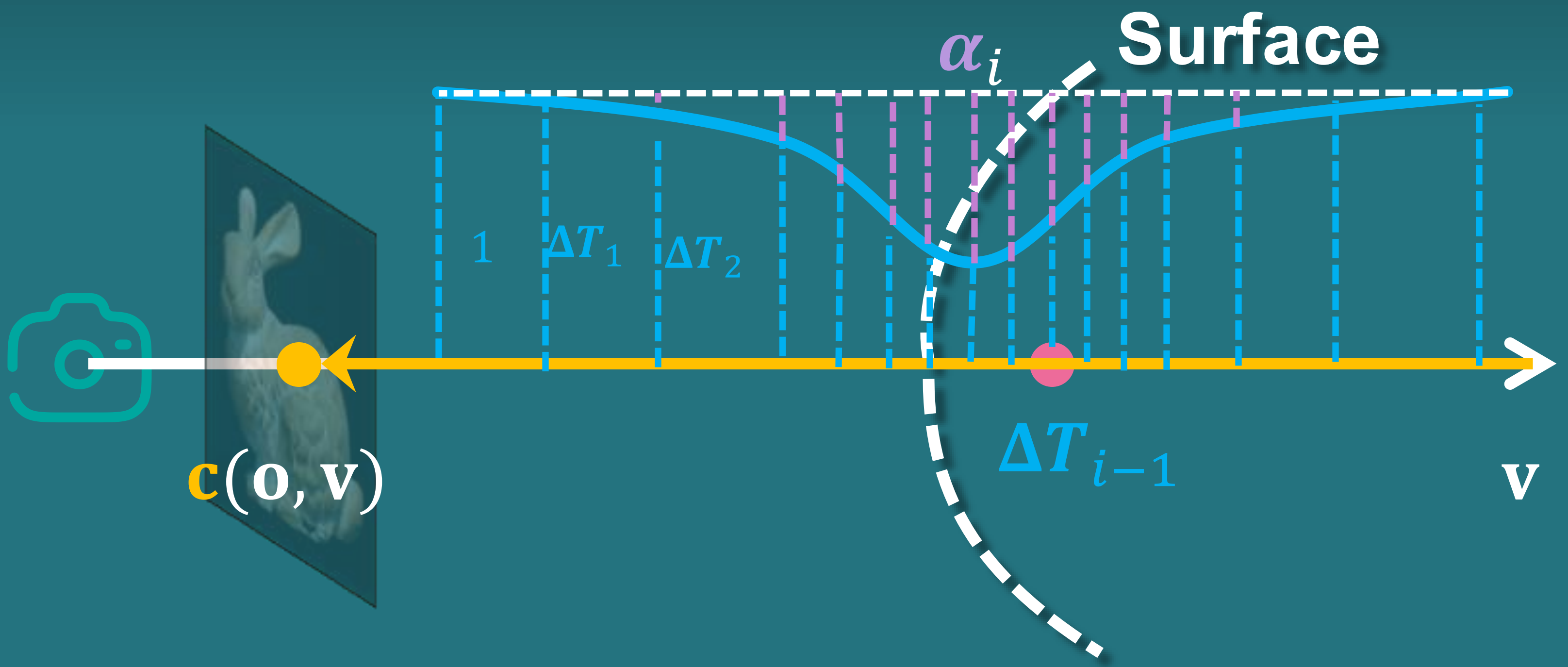
$I(\mathbf{c})$: Light (Color)

t: Ray Parameter

ΔT: Transmittance

α: Opacity

Accumulated Transmittance



$$\alpha_i = 1 - \Delta T_i = 1 - \exp(-\rho_i \delta t_i)$$

$$I_i = I_0 \Delta T_1 \Delta T_2 \Delta T_3 \cdots = I_0 T_{i-1}$$

ρ: Density

$I(\mathbf{c})$: Light (Color)

t: Ray Parameter

ΔT: Transmittance

T: Accumulated Transmittance

α: Opacity

Volumetric Rendering

$$\mathbf{c}(\mathbf{o},\mathbf{v}) = \sum_{i=1}^{n} w_i \mathbf{c}_i$$

$$= \sum_{i=1}^{n} T_i \alpha_i \mathbf{c}_i$$

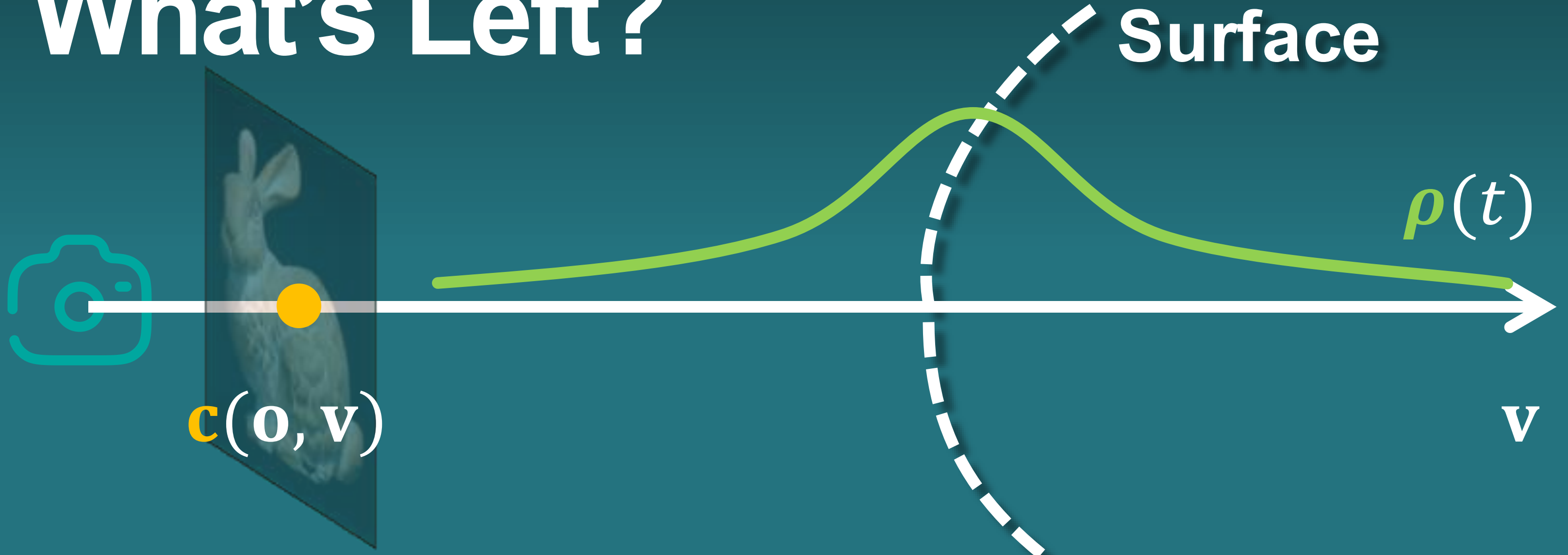
$\mathbf{c}$: Color

w: Weight

α: Opacity

T: Accumulated Transmittance

What's Left?



$$\alpha_i = 1 - \exp(-\rho_i \delta t_i)$$

$$T_i = (1-\alpha_1)(1-\alpha_2)\cdots(1-\alpha_{i-1})$$

$$\mathbf{c}(\mathbf{o},\mathbf{v}) = \sum_{i=1}^{n} w_i \mathbf{c}_i = \sum_{i=1}^{n} T_i \alpha_i \mathbf{c}_i$$

ρ: Density

$I(\mathbf{c})$: Light (Color)

w: Weight

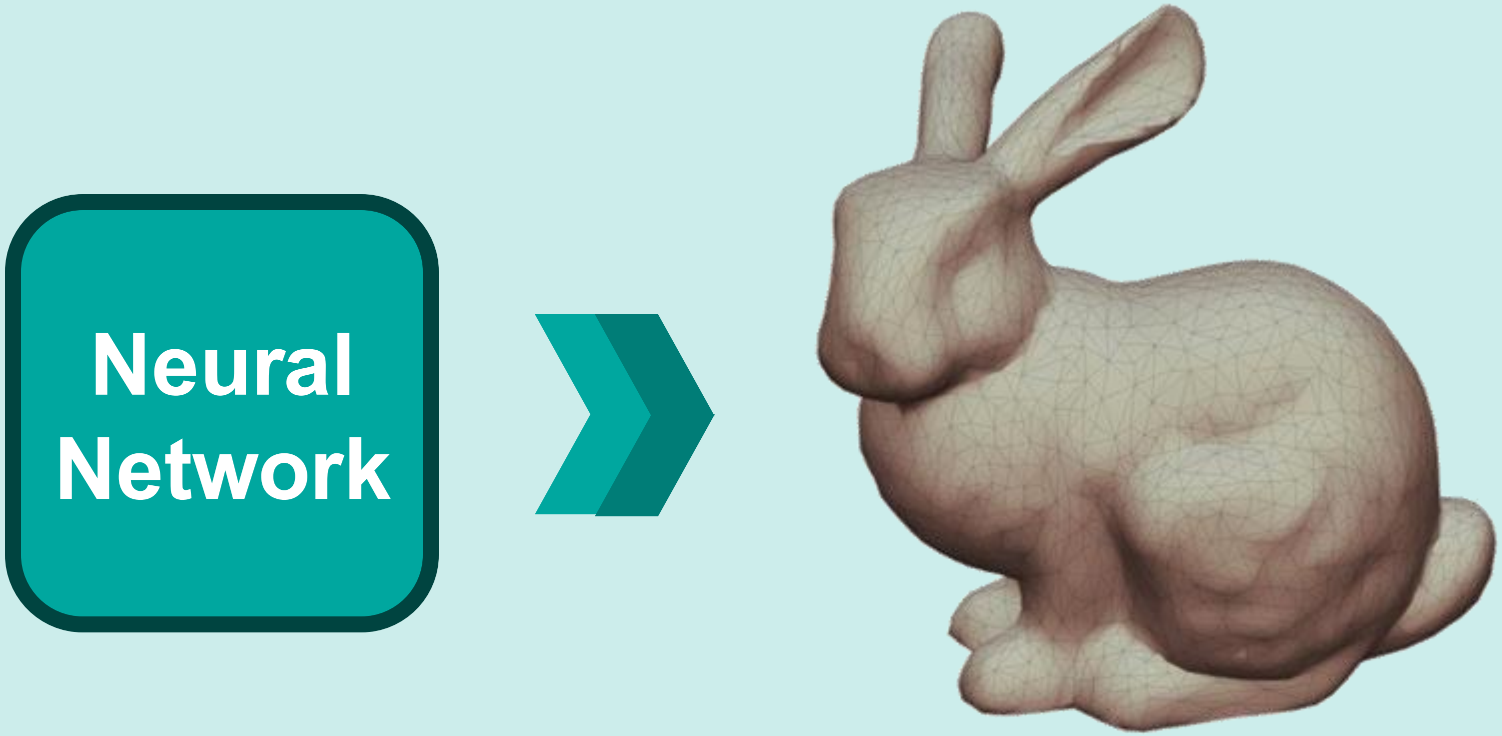
t: Ray Parameter

ΔT: Transmittance

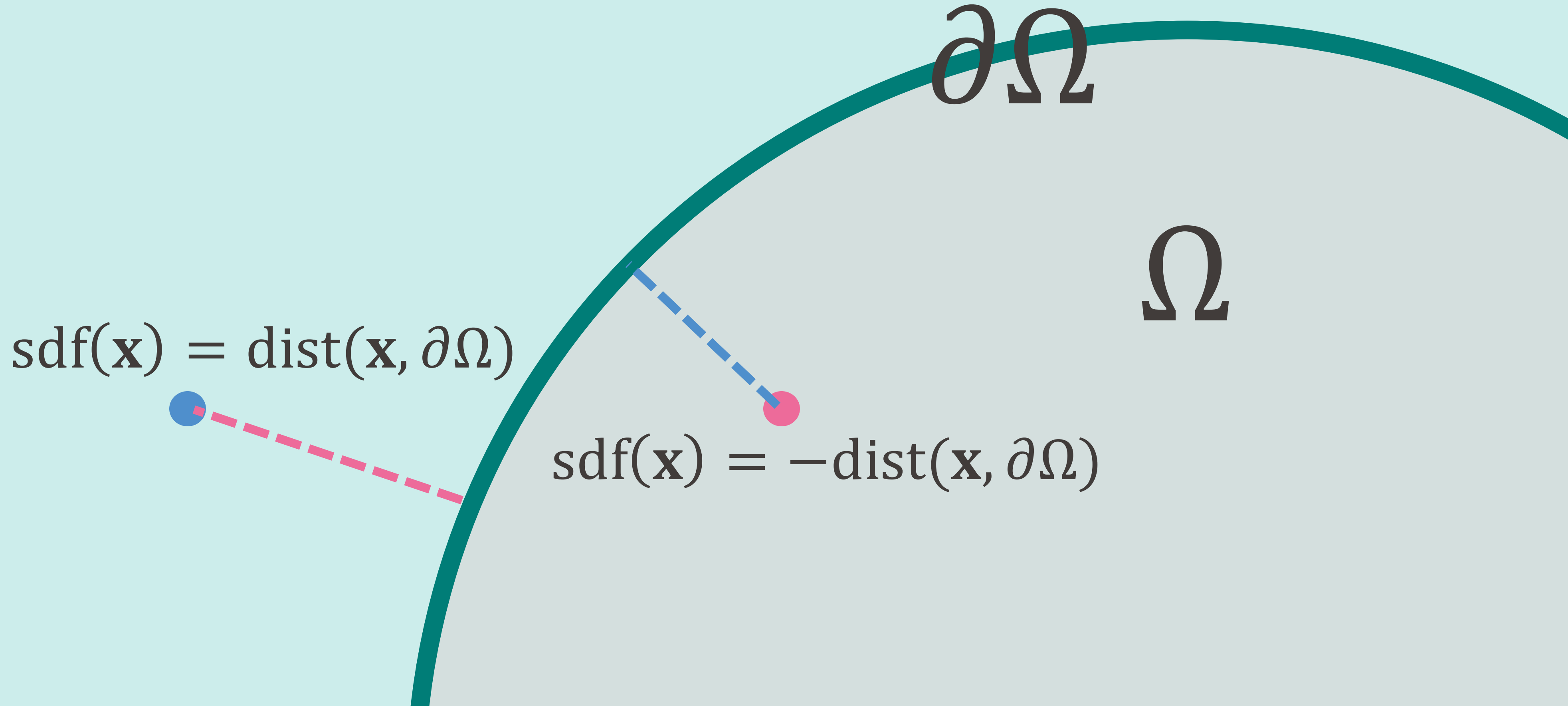
T: Accumulated Transmittance

α: Opacity

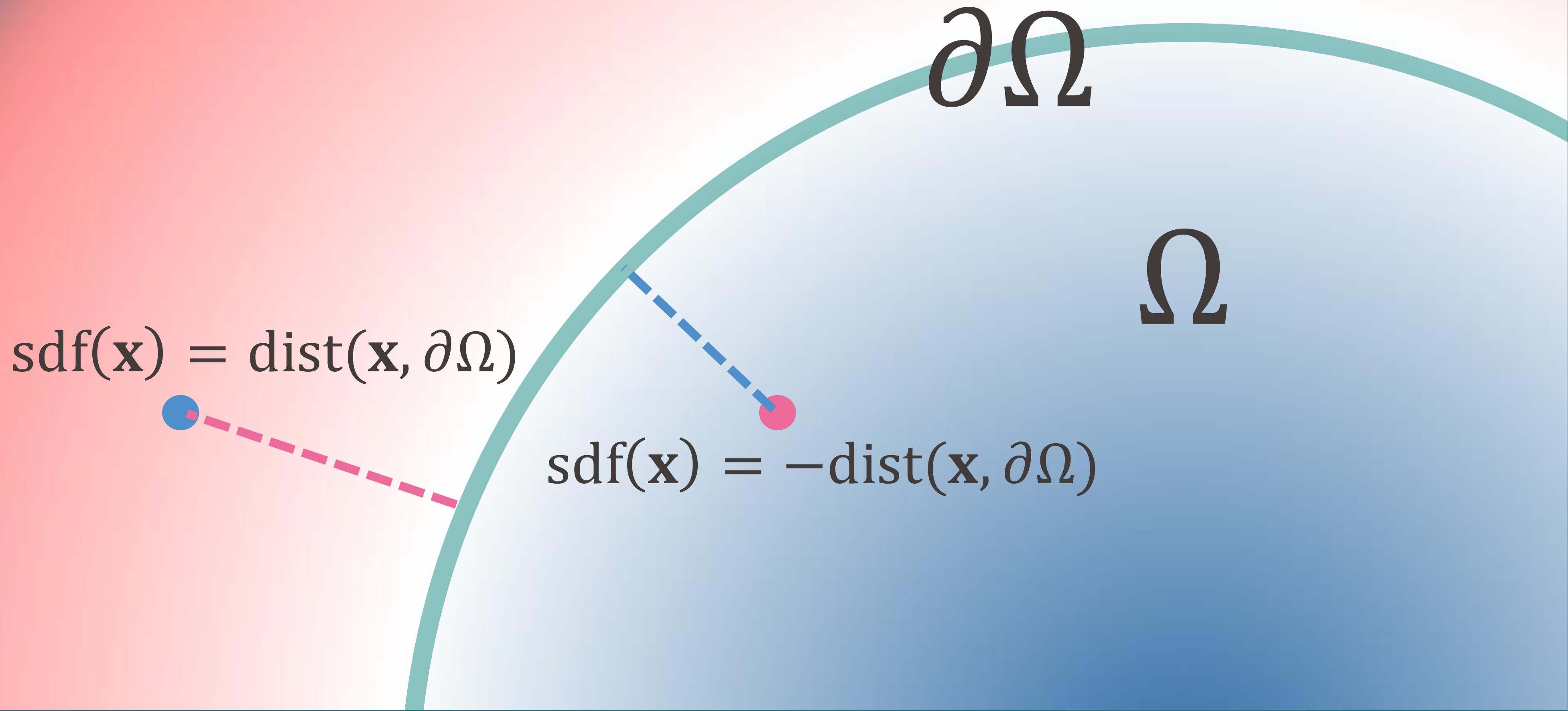
Use Neural Network to Represent the Surface



Signed Distance Function (SDF)



Signed Distance Function (SDF)



Surface Representation

Neural Network

$\mathrm{sdf}(\mathbf{x})$

$\mathrm{sdf}(\mathbf{x}) = 0$
Surface

3D Position

Signed Distance Function (SDF)

$\mathrm{sdf}(\mathbf{x}) > 0$

Outside

$\mathrm{sdf}(\mathbf{x}) = 0$

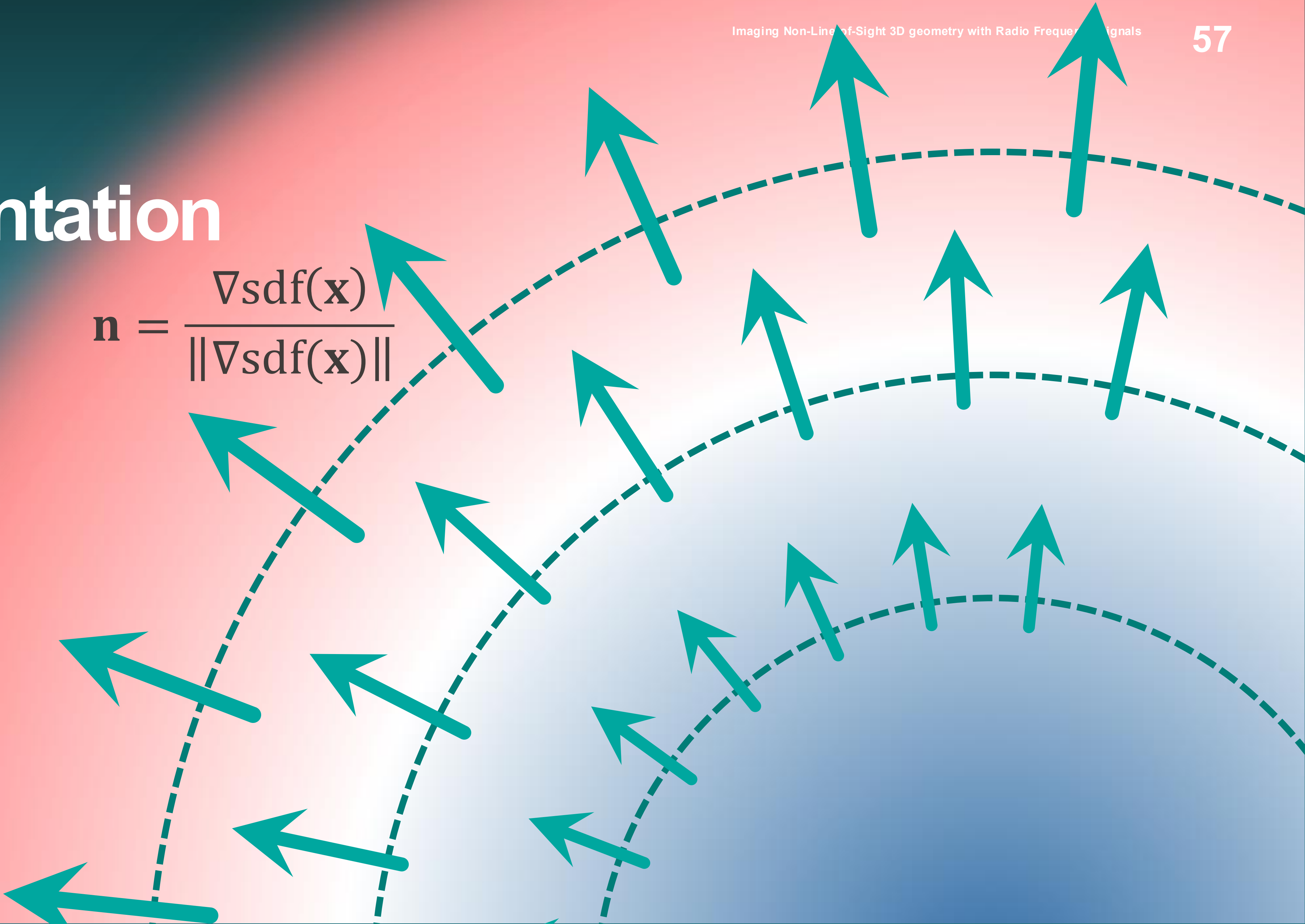
Boundary

$\mathrm{sdf}(\mathbf{x}) < 0$

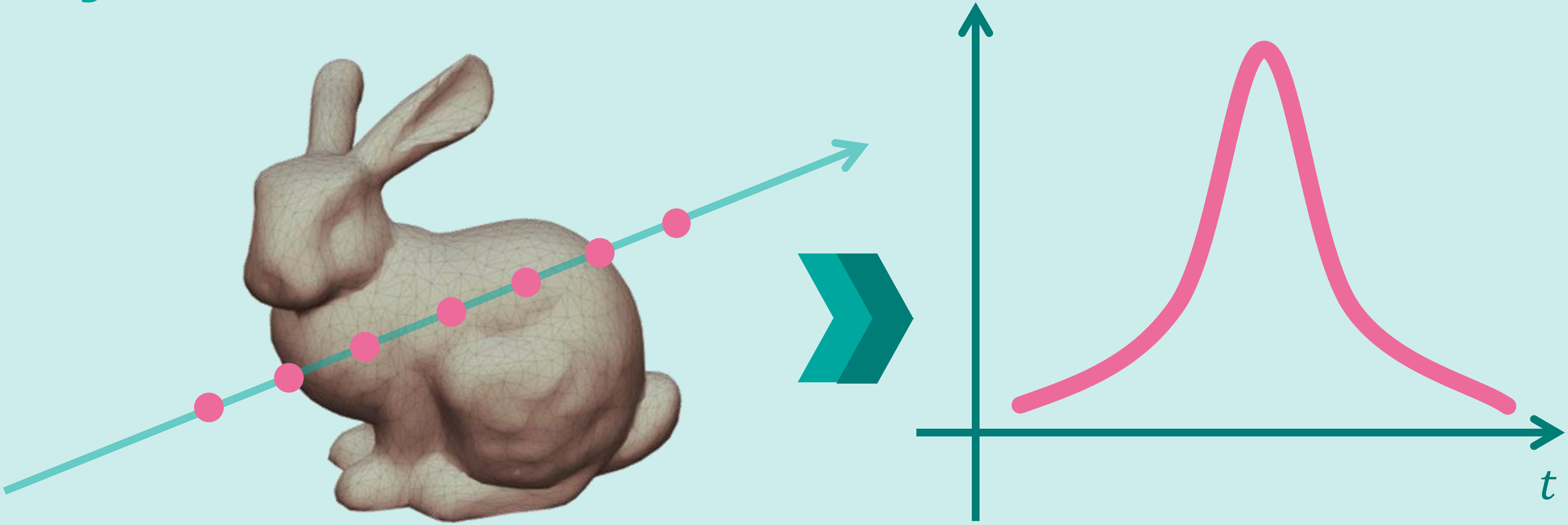
Inside

Normal Representation

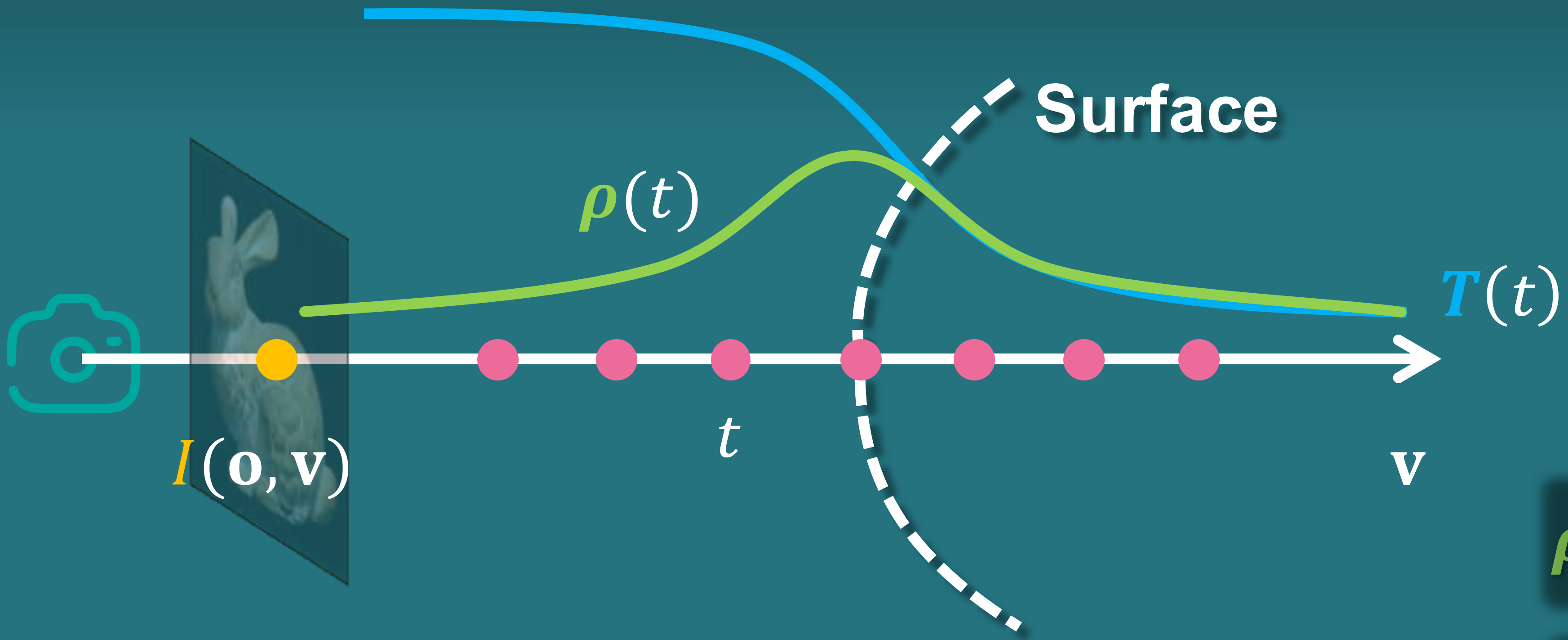
$$\mathbf{n} = \frac{\nabla \mathrm{sdf}(\mathbf{x})}{\|\nabla \mathrm{sdf}(\mathbf{x})\|}$$



How to Represent Density Distribution on the Ray from the Surface?



Continuous Perspective



Beer–Lambert Law

$$\frac{d\boldsymbol{I}}{dt} = -\boldsymbol{\rho}(t)\boldsymbol{I}(t)$$

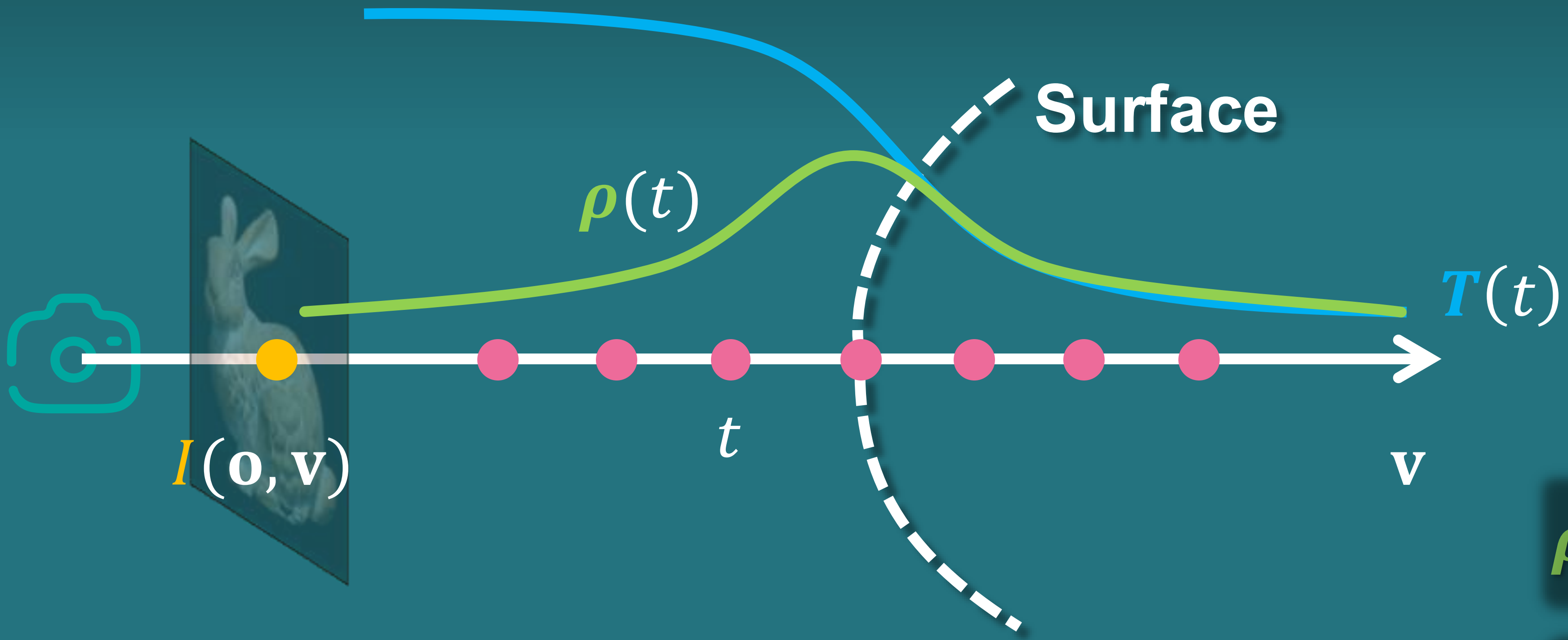
ρ: Density

$I(\mathbf{c})$: Light (Color)

t: Ray Parameter

T: Accumulated Transmittance

Continuous Perspective



Solve ODE

$$I(t) = I(0) \exp\left(-\int_0^t \rho(u)\, du\right)$$

$T(t)$

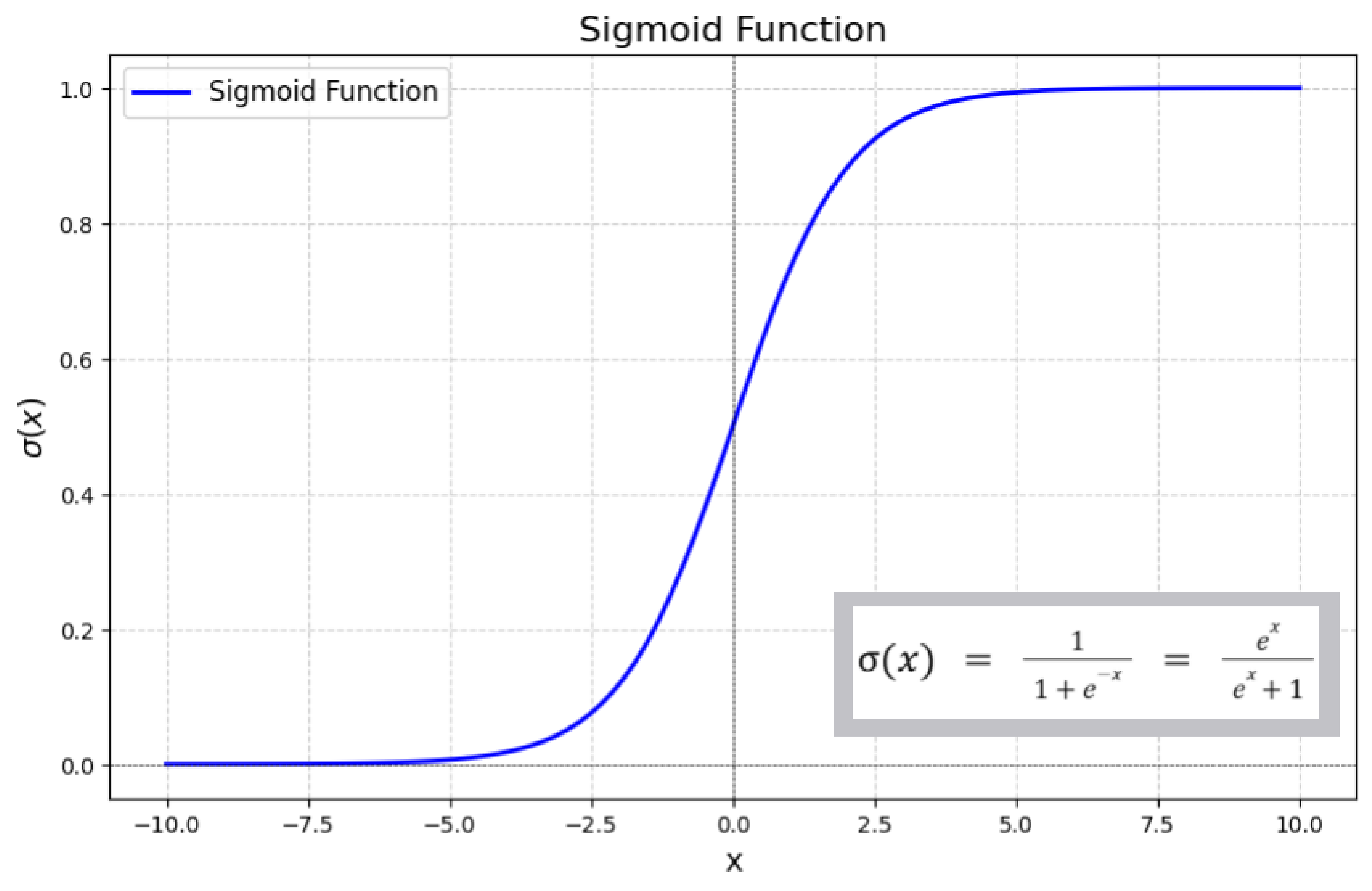
ρ: Density

$I(\mathbf{c})$: Light (Color)

t: Ray Parameter

T: Accumulated Transmittance

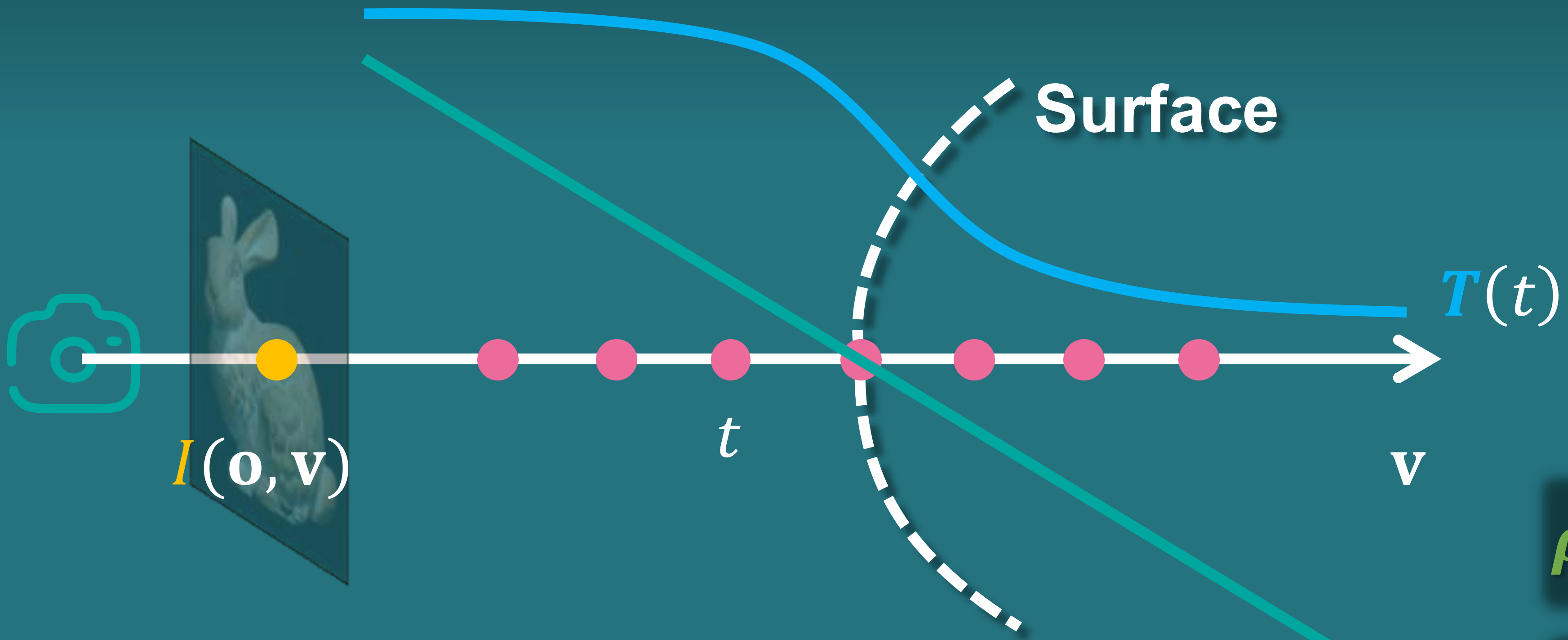
A



“the s

Wang, Peng, et
information proc

Sigmoid Function



Neus Assumption:

$$T(t) = \Phi\left(\mathrm{sdf}\left(\mathbf{x}(t, \mathrm{v})\right)\right)$$

ρ: Density

$I(\mathbf{c})$: Light (Color)

t: Ray Parameter

T: Accumulated Transmittance

Neus Volumetric Rendering

$$T_i = (1-\alpha_1)(1-\alpha_2)\cdots(1-\alpha_{i-1})$$

Two functions

$$T(t) = \mathbf{\Phi}\left(\mathrm{sdf}\big(\mathbf{x}(t,\mathbf{v})\big)\right)$$

$$\alpha_i = 1 - \frac{T_{i+1}}{T_i}$$

$$\alpha_i = 1 - \frac{\mathbf{\Phi}\left(\mathrm{sdf}\big(\mathbf{x}(t_{i+1},\mathbf{v})\big)\right)}{\mathbf{\Phi}\left(\mathrm{sdf}\big(\mathbf{x}(t_i,\mathbf{v})\big)\right)}$$

ρ: Density

$I(\mathbf{c})$: Light (Color)

t: Ray Parameter

T: Accumulated Transmittance

α: Opacity

Summary

$$\alpha_i = 1 - \frac{\Phi\left(\mathrm{sdf}\left(\mathbf{x}(t_{i+1}, \mathbf{v})\right)\right)}{\Phi\left(\mathrm{sdf}\left(\mathbf{x}(t_i, \mathbf{v})\right)\right)}$$

$$T_i = (1 - \alpha_1)(1 - \alpha_2)\cdots(1 - \alpha_{i-1})$$

$$\mathbf{c}(\mathbf{o}, \mathbf{v}) = \sum_{i=1}^{n} w_i \mathbf{c}_i = \sum_{i=1}^{n} T_i \alpha_i \mathbf{c}_i$$

ρ: Density

$I(\mathbf{c})$: Light (Color)

w: Weight

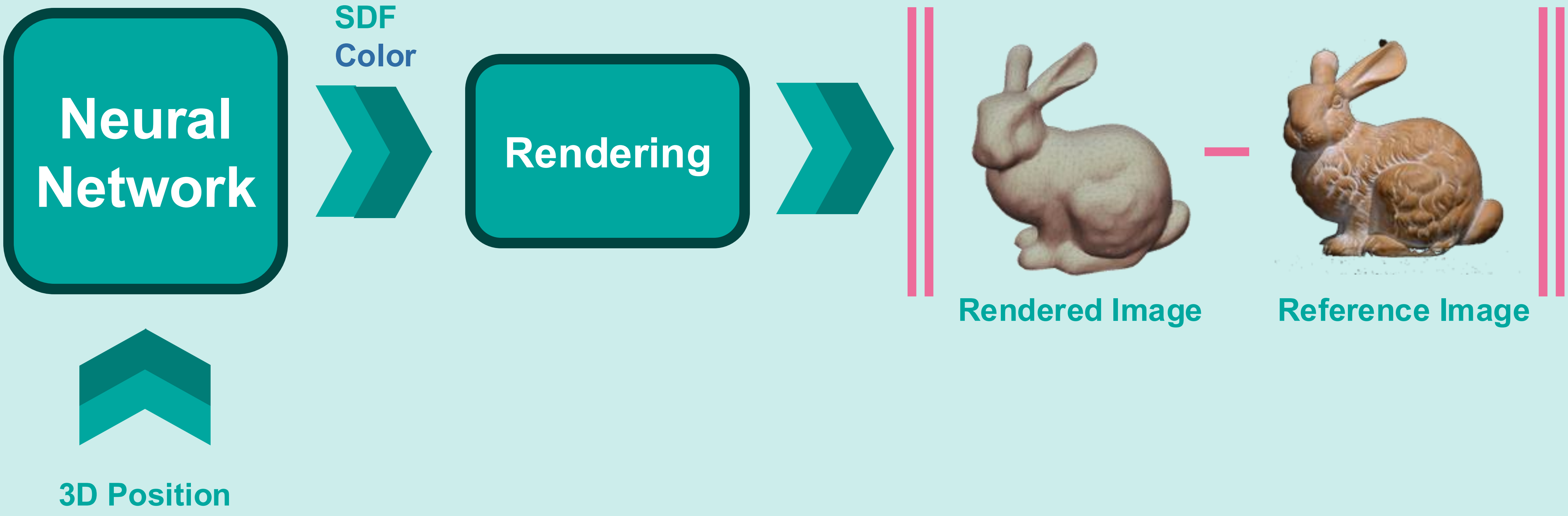
t: Ray Parameter

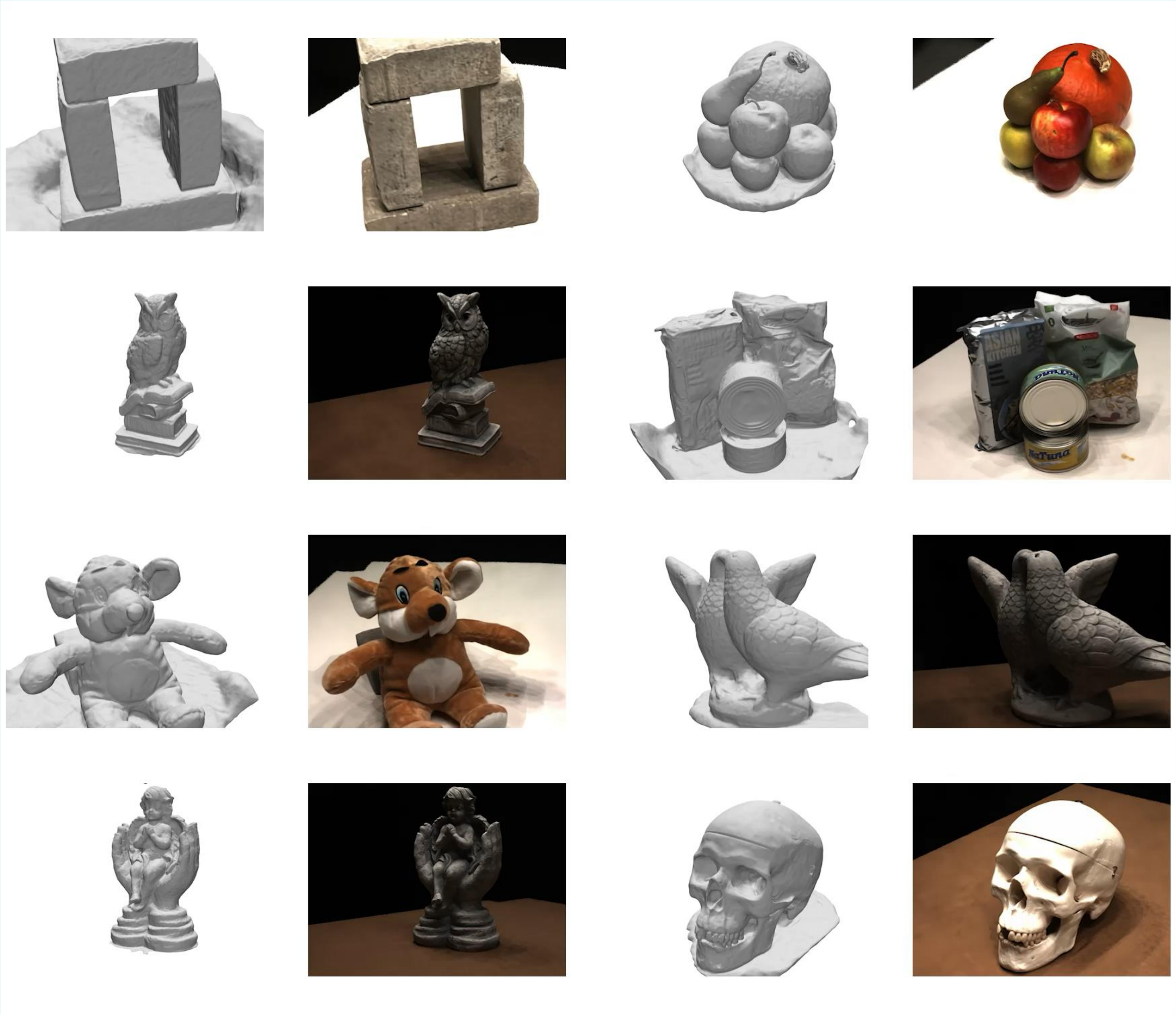
ΔT: Transmittance

T: Accumulated Transmittance

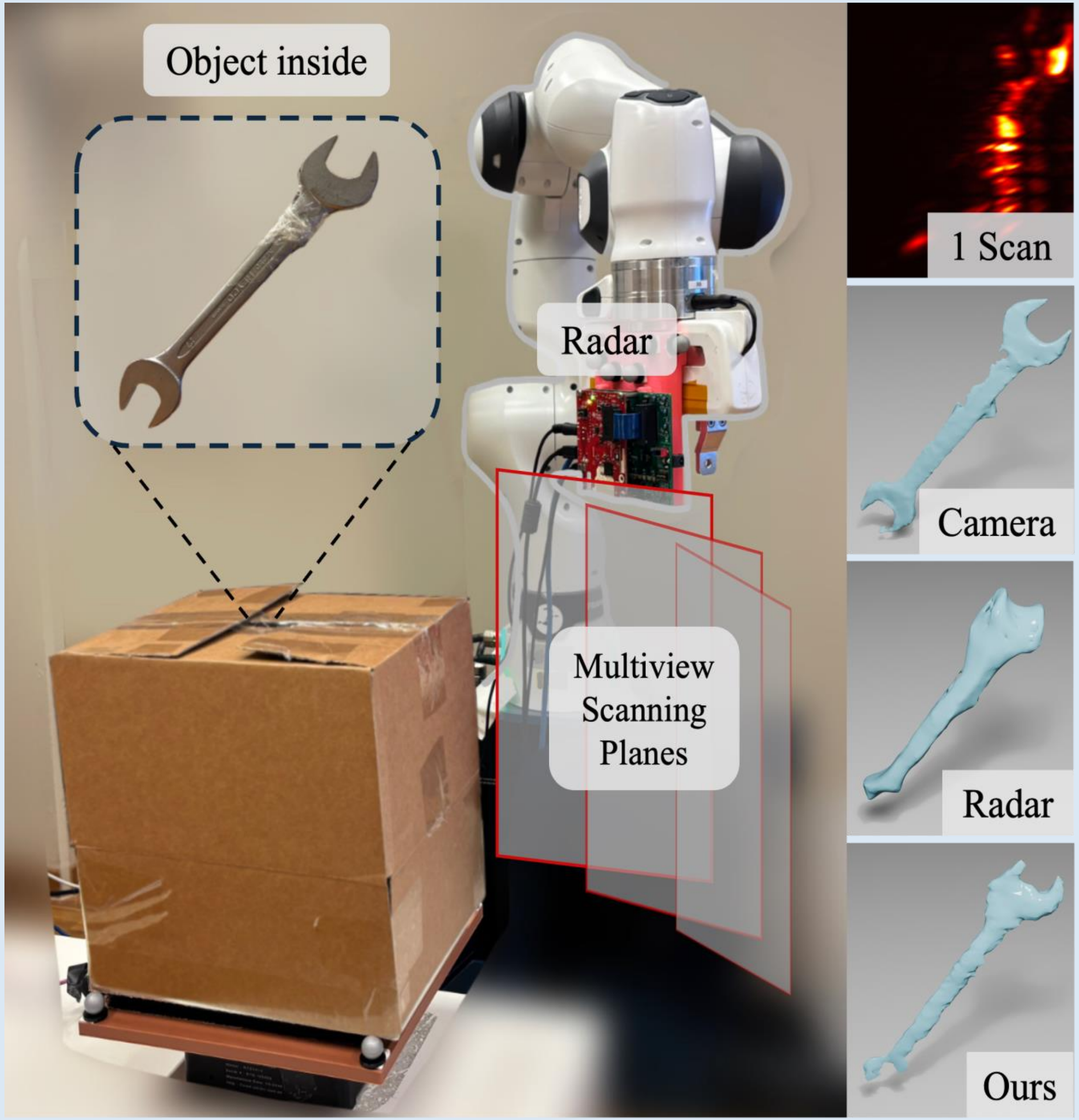
α: Opacity

Summary


Neural Network
SDF
Color
Rendering
Rendered Image
Reference Image
3D Position



How to apply Neural Radiance Field to Radio Frequency?



High-Fidelity Reconstruction from Visible Light

3D reconstruction from photographs

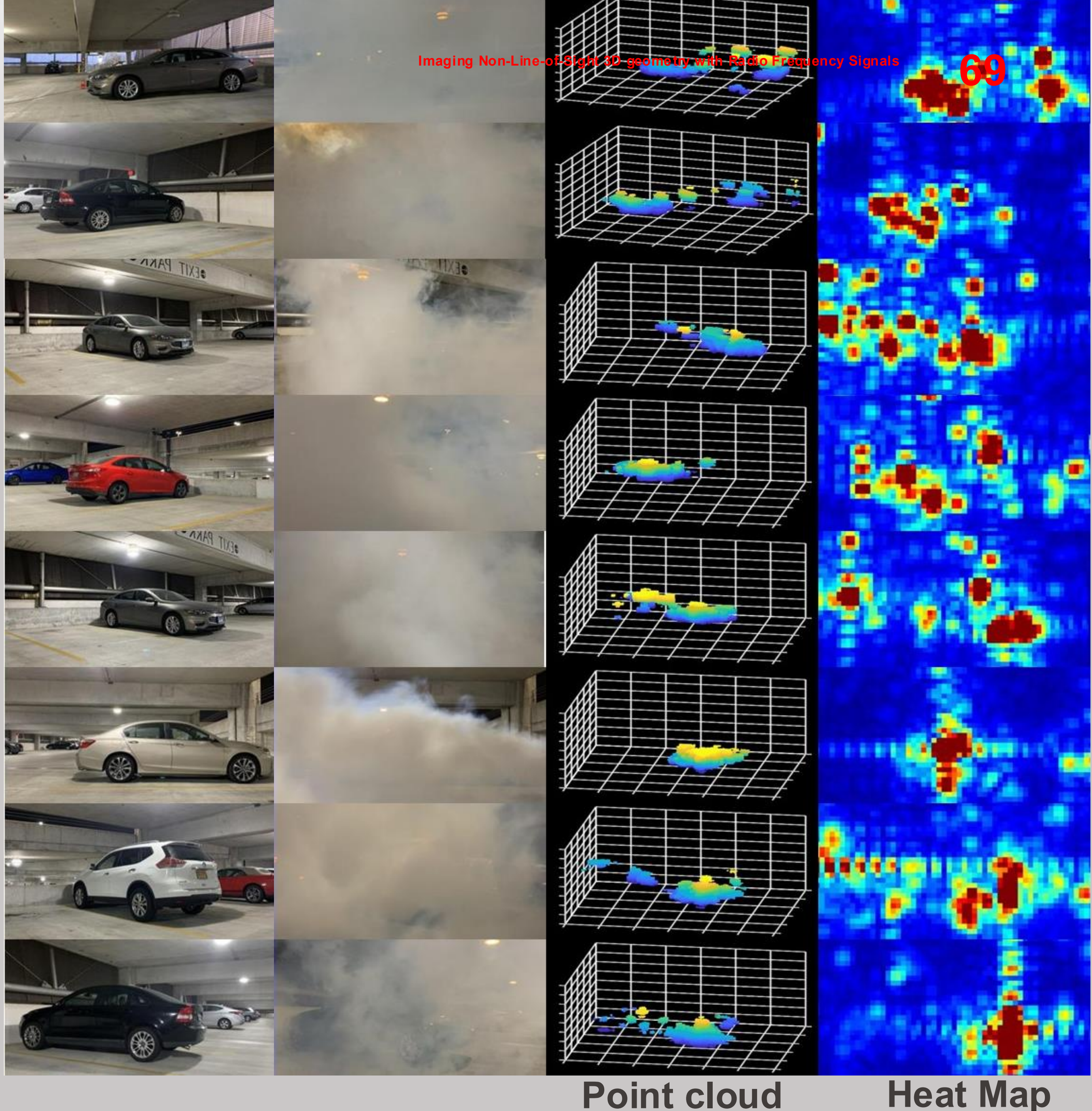
Using the Gold cape dataset from NeRD, we extract a triangle mesh, materials and environment lighting.

- 3D Geometry
- Surface Materials
- Lighting

Munkberg, Jacob, et al. "Extracting triangular 3d models, materials, and lighting from images." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

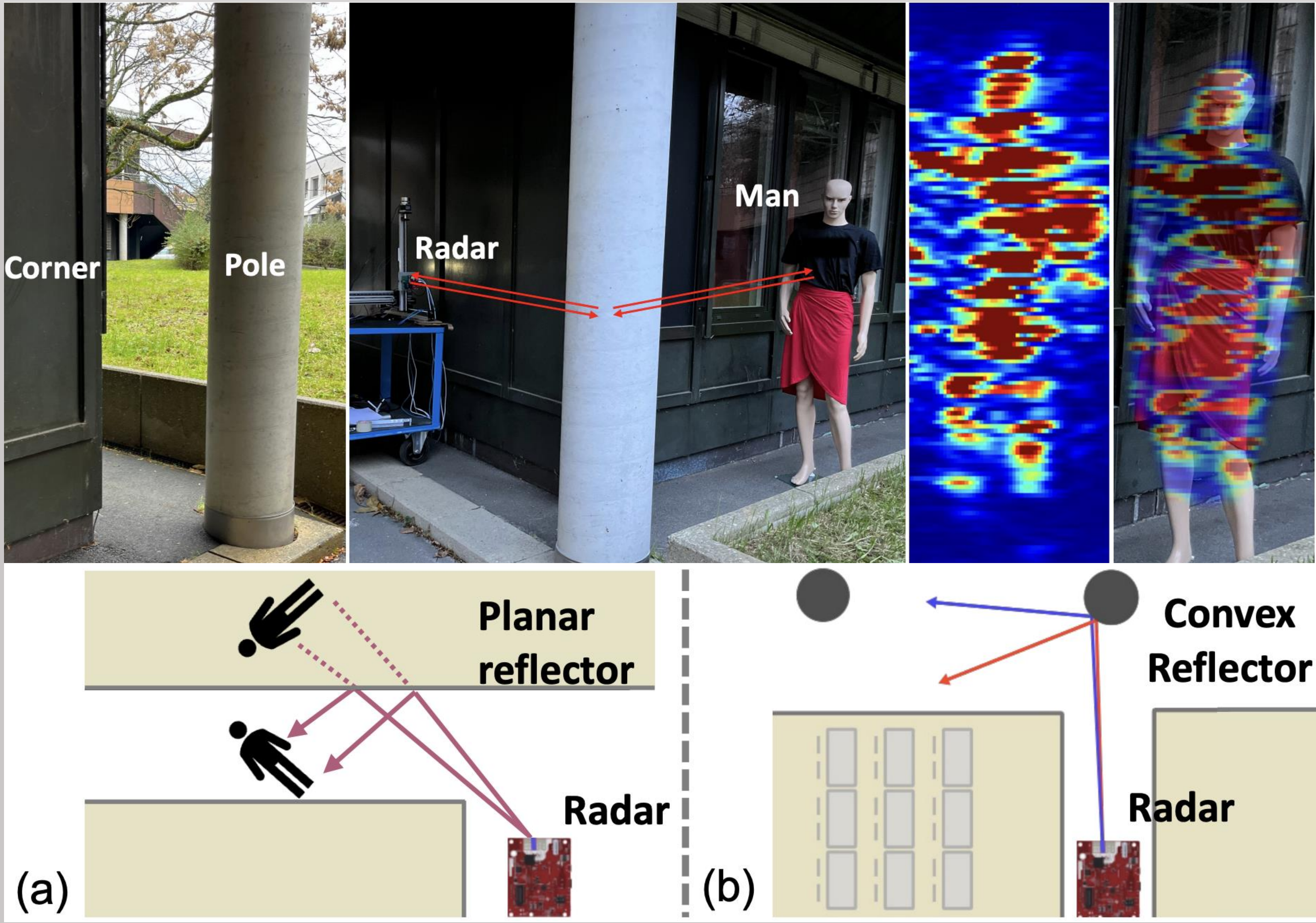
Challenges of Radio Frequency Imaging:

Lower Resolution
&
Higher Noise



Guan, Junfeng, et al. "Through fog high-resolution imaging using millimeter wave radar." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

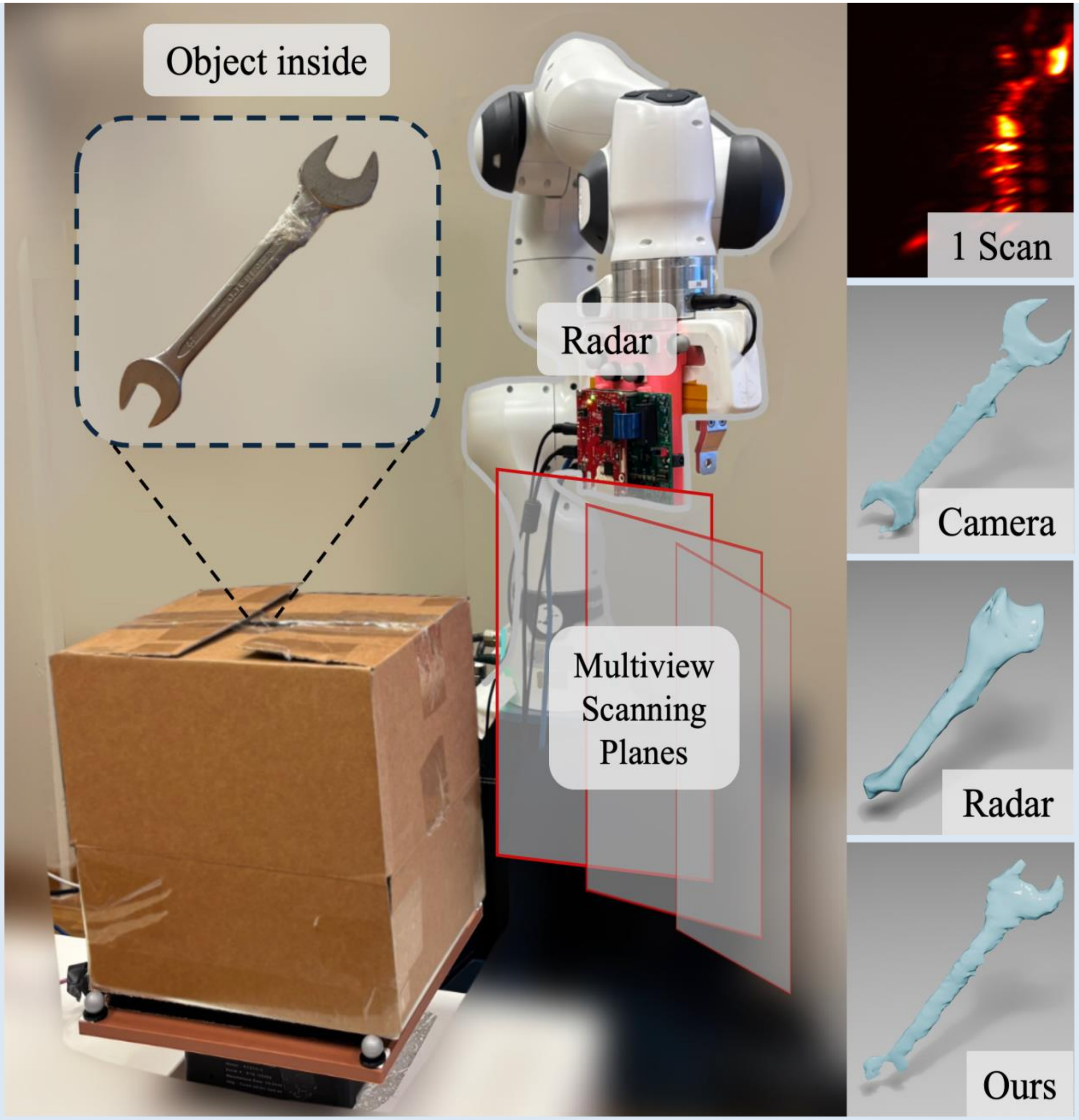
Challenges of Radio Frequency Imaging:
dominated by specular reflection

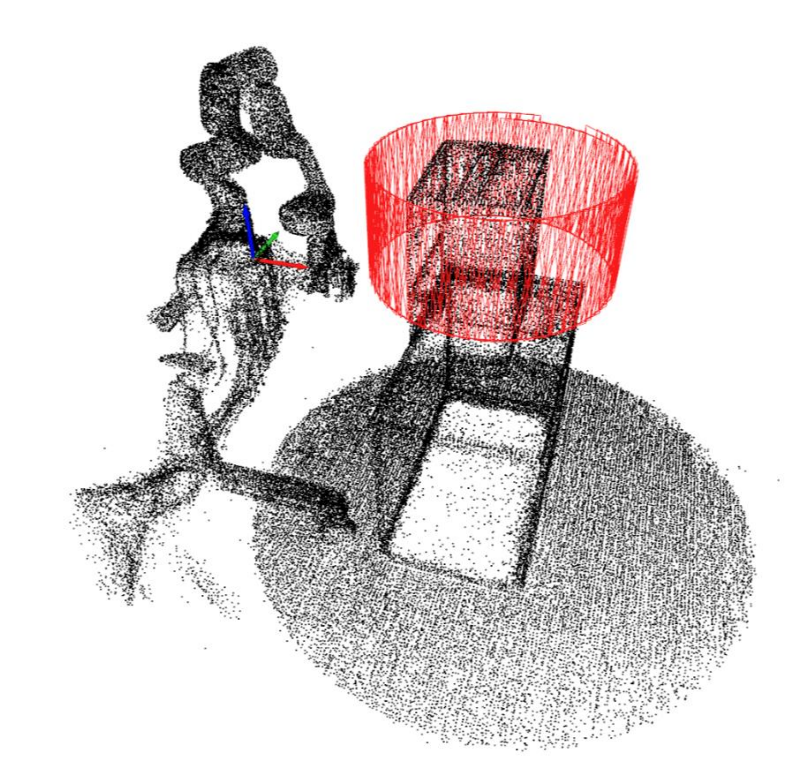


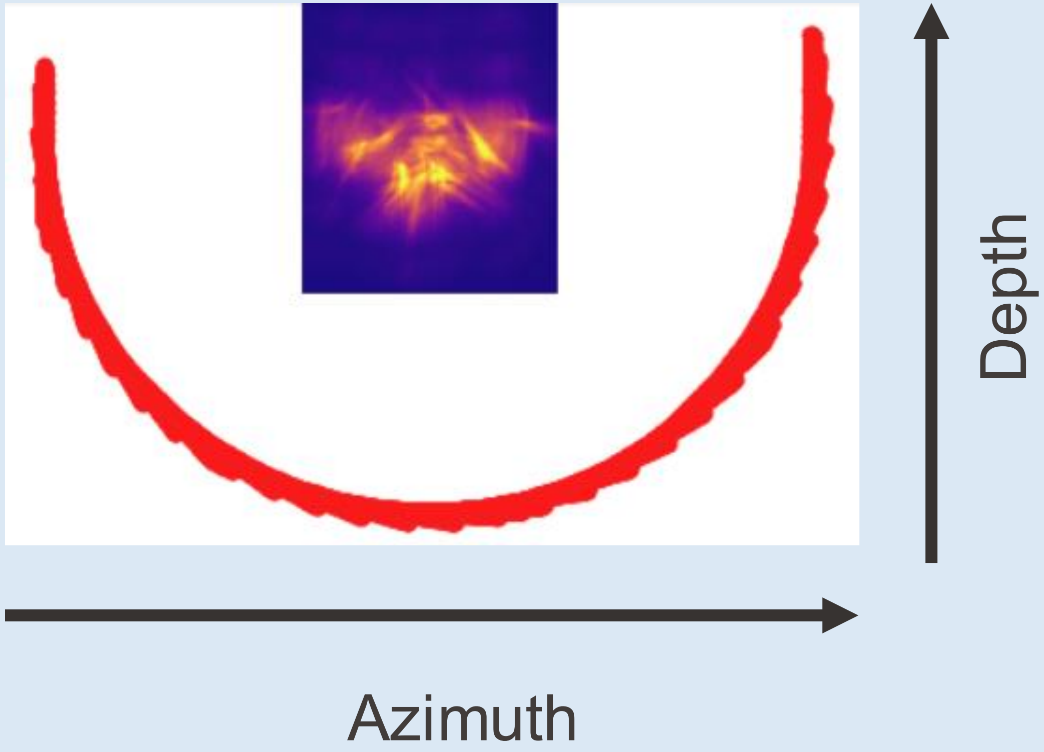
Dodds, Laura, et al. "Around the corner mmWave imaging in practical environments." *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*. 2024.

Challenges of Radio Frequency Imaging:
Data Driven

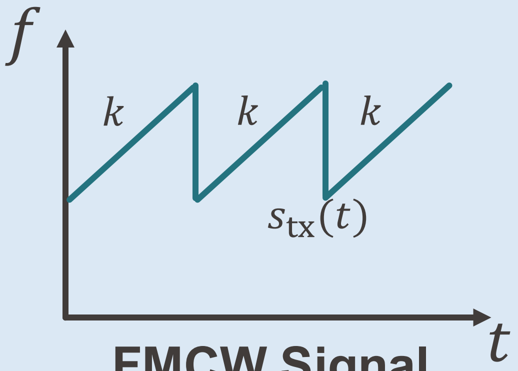
Zhao, Mingmin, et al. "Through-wall human pose estimation using radio signals." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.


Object inside
Radar
Multiview Scanning Planes
1 Scan
Camera
Radar
Ours

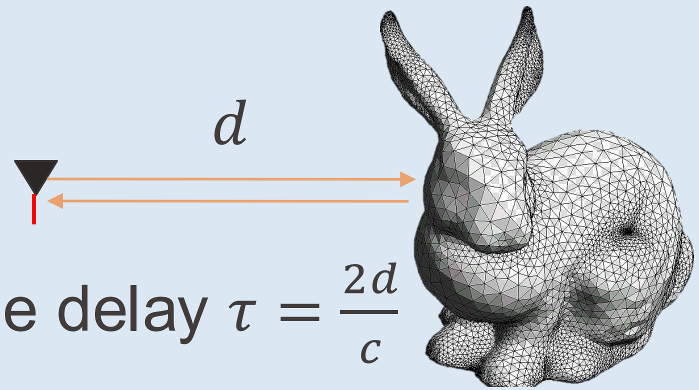



Depth
Azimuth

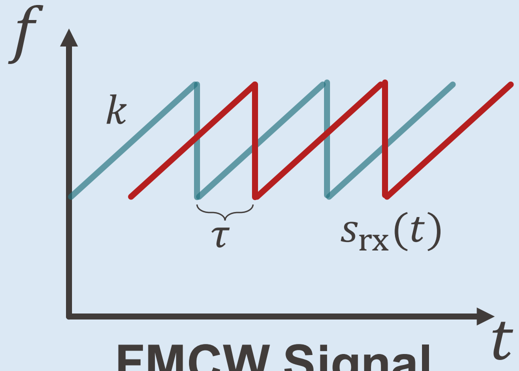
Matched Filter



FMCW Signal

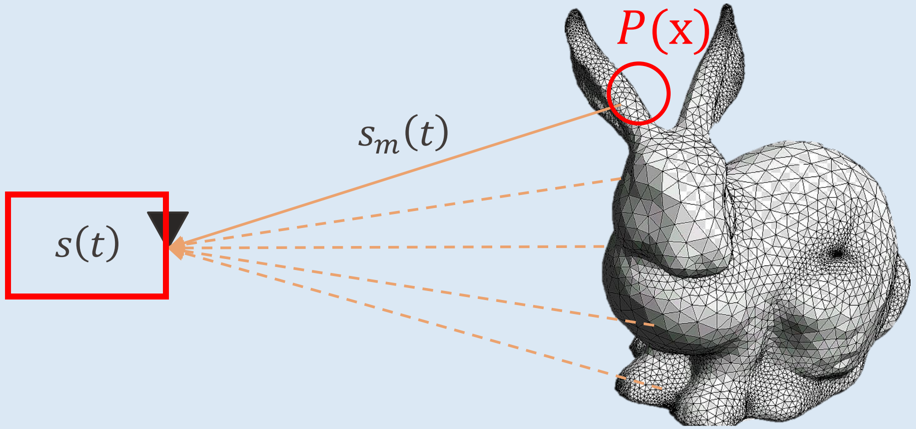


Time delay $\tau = \frac{2d}{c}$



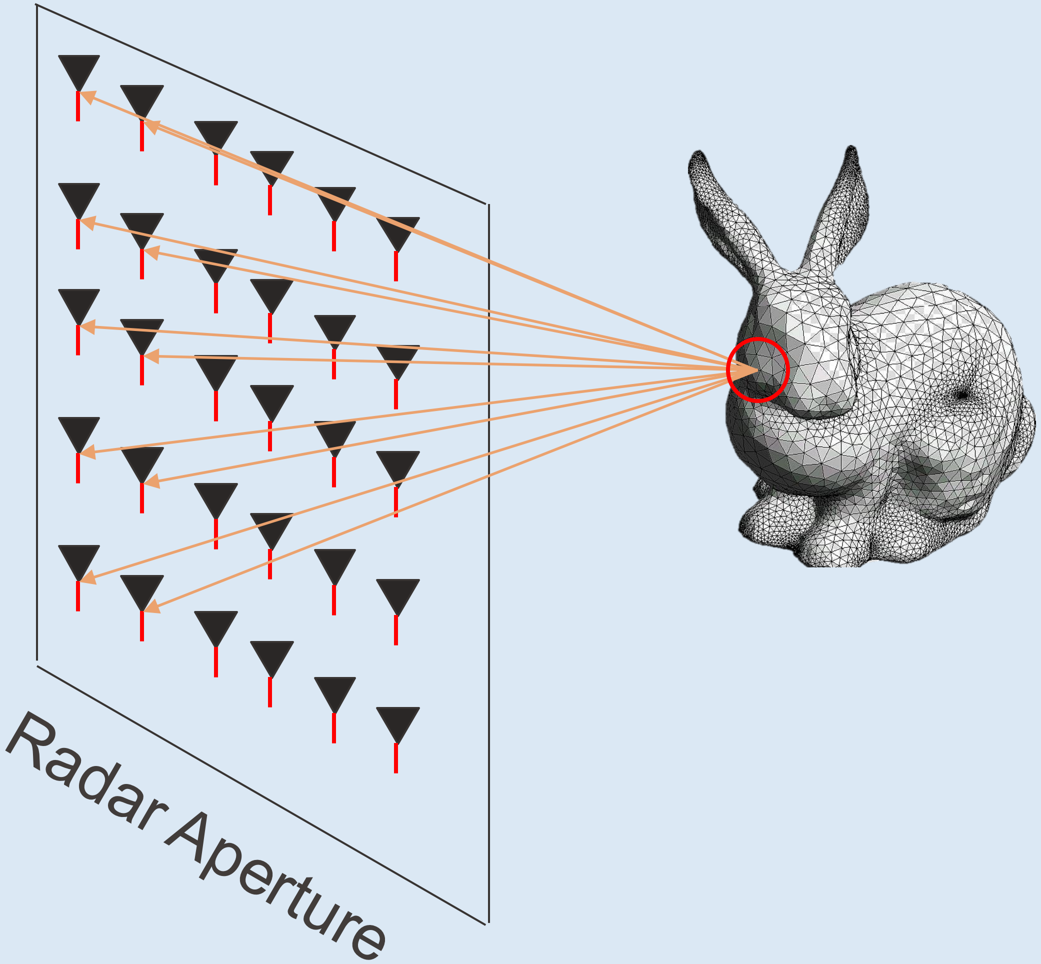
FMCW Signal

We mix the transmitted signal and received signal: $s_m(t) = s_{tx}(t) \times s_{rx}^*(t) = Ae^{-j2\pi(f+kt)\tau} = Ae^{-j2\pi \cdot k\tau \cdot t} e^{-j2\pi \cdot f\tau}$



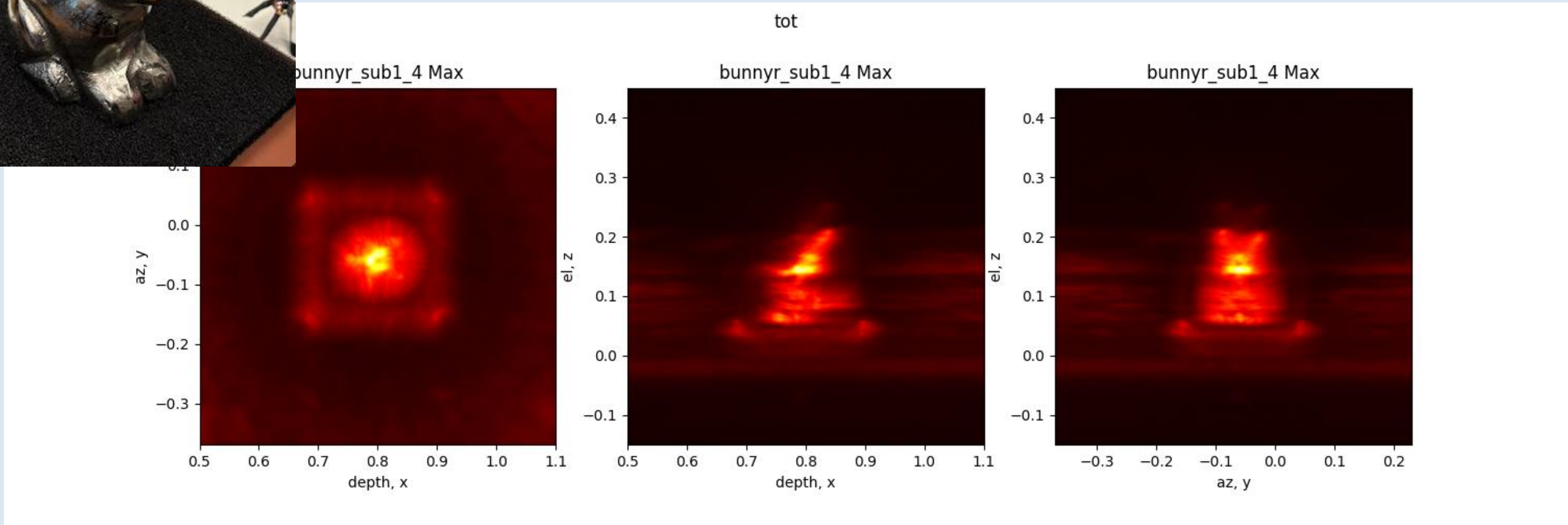
$$P(\mathbf{x}) = \left\| \sum_t s(t) \times s_m^*(t) \right\|$$

Matched Filter



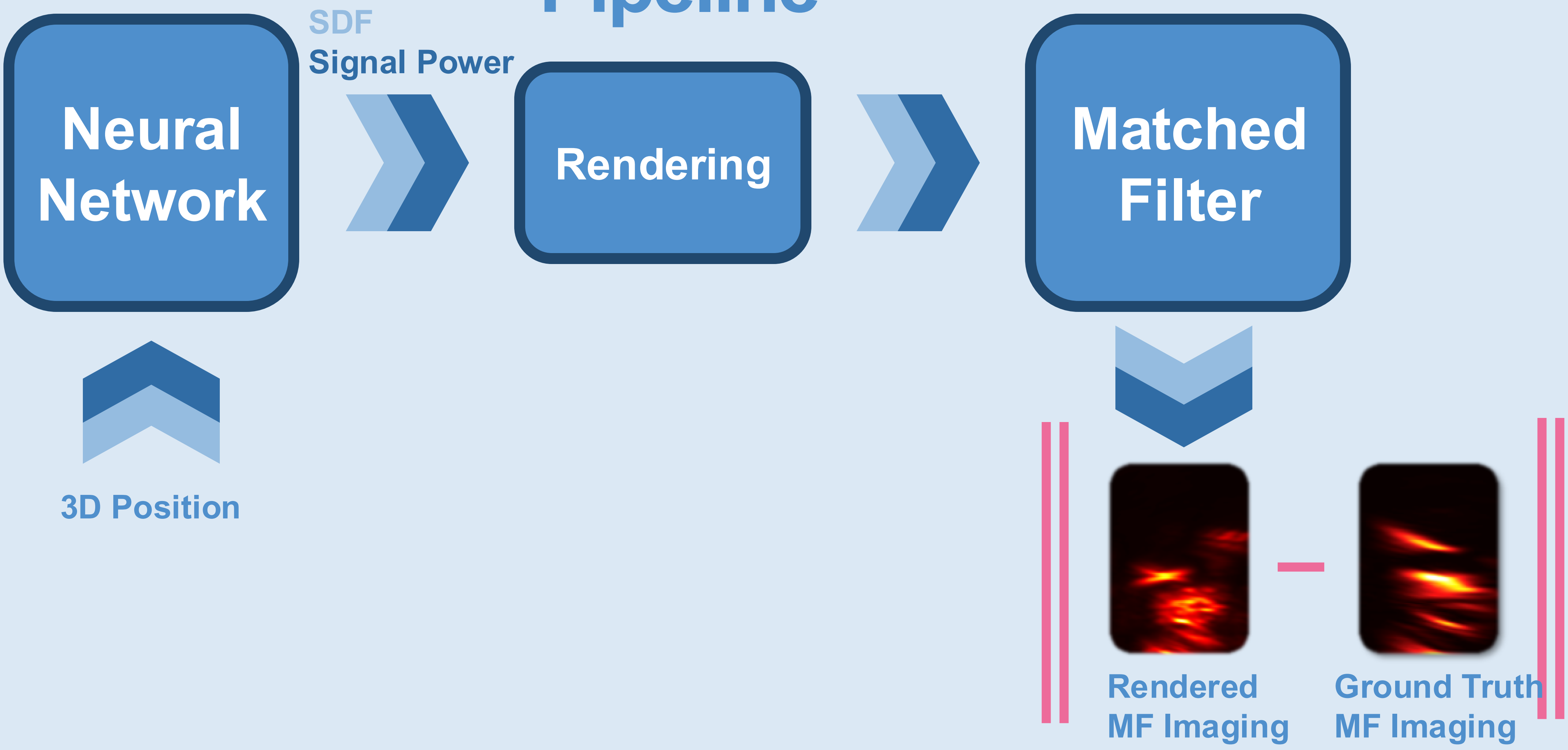
$$P(\mathbf{x}) = \left\| \sum_{i=1}^{N_{\text{ant}}} \sum_{t} s(i,t) \times s_m^*(i,t) \right\|$$

Matched Filter

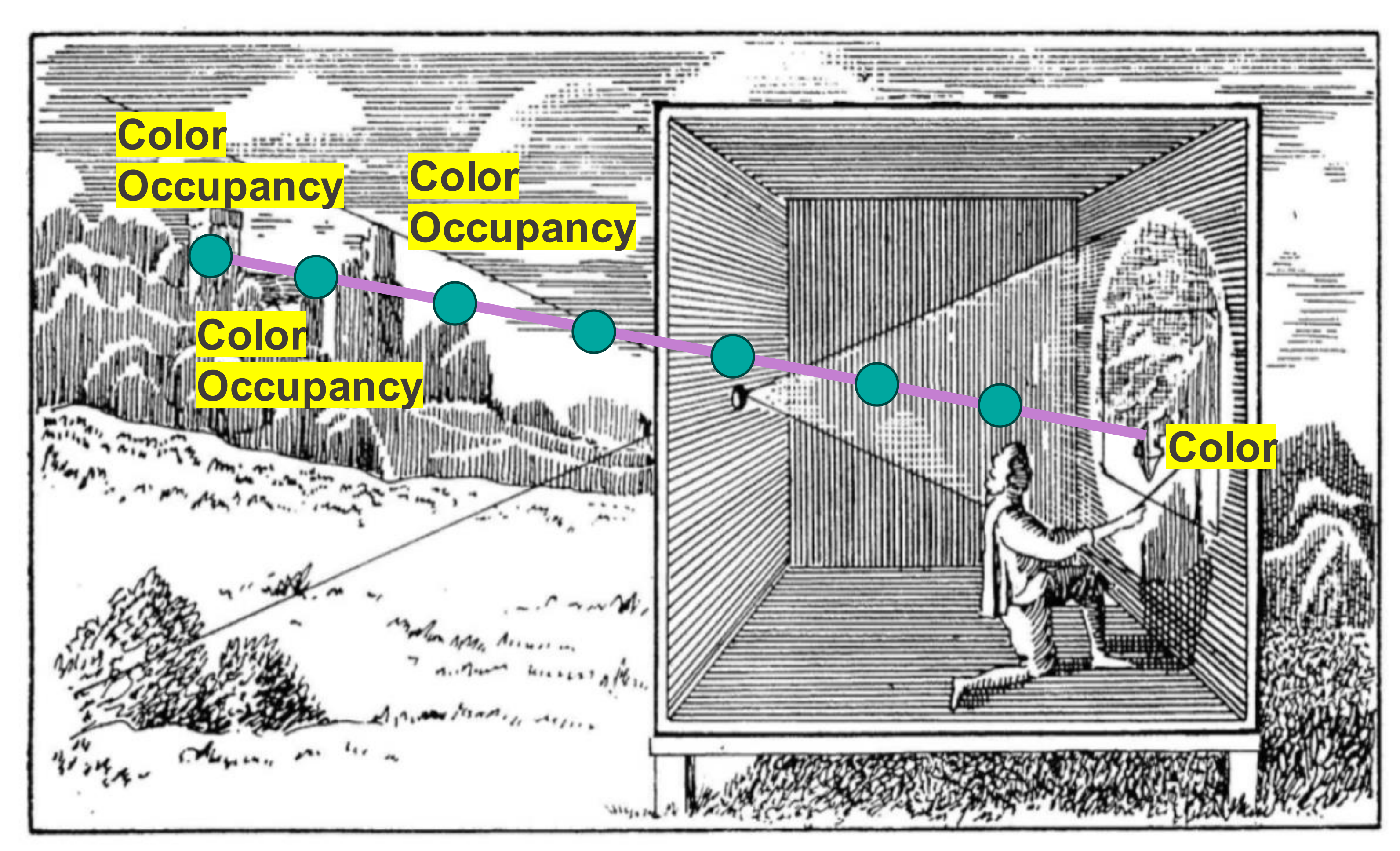


Matched Filter

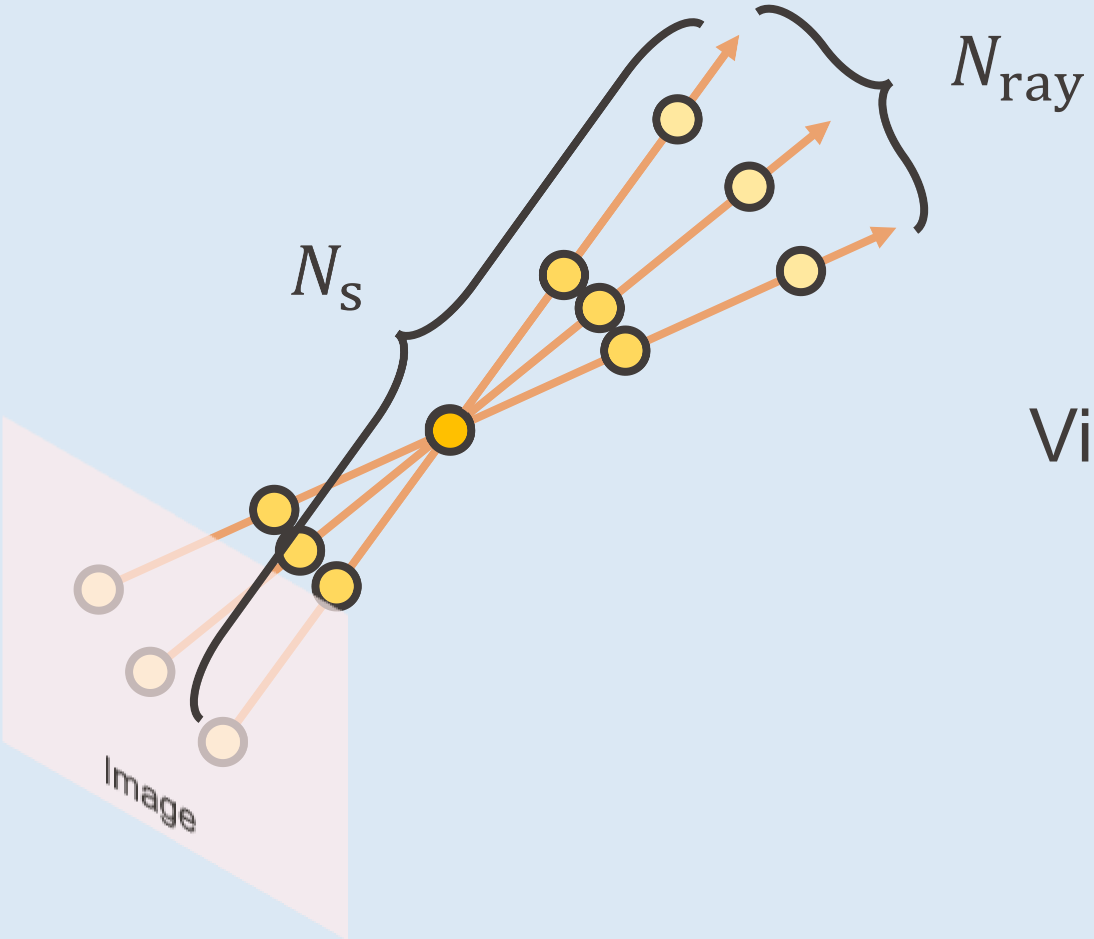
Pipeline


Neural Network
SDF
Signal Power
Rendering
Matched Filter
3D Position
Rendered MF Imaging
Ground Truth MF Imaging

Pinhole model is not applicable in RF

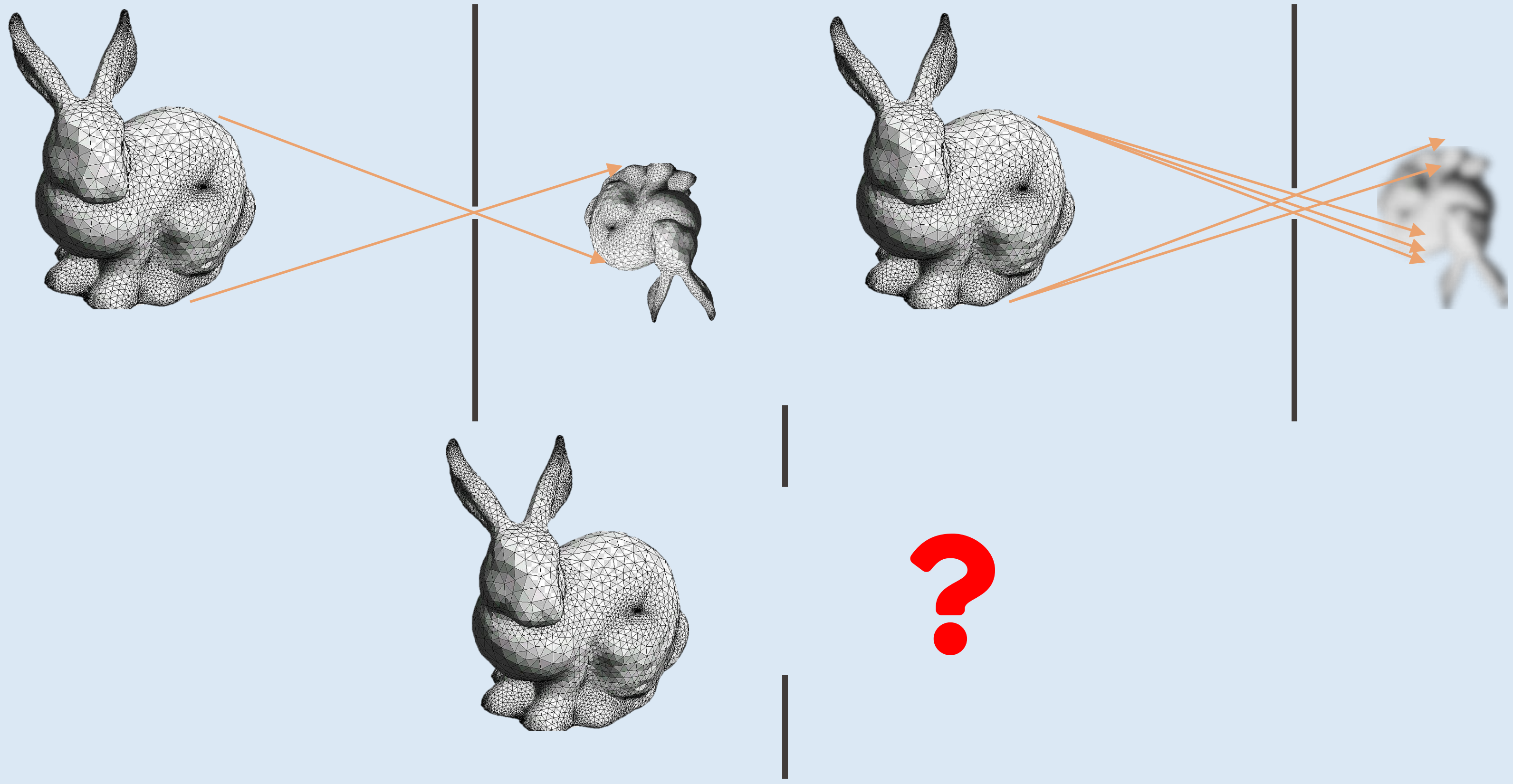


Camera model

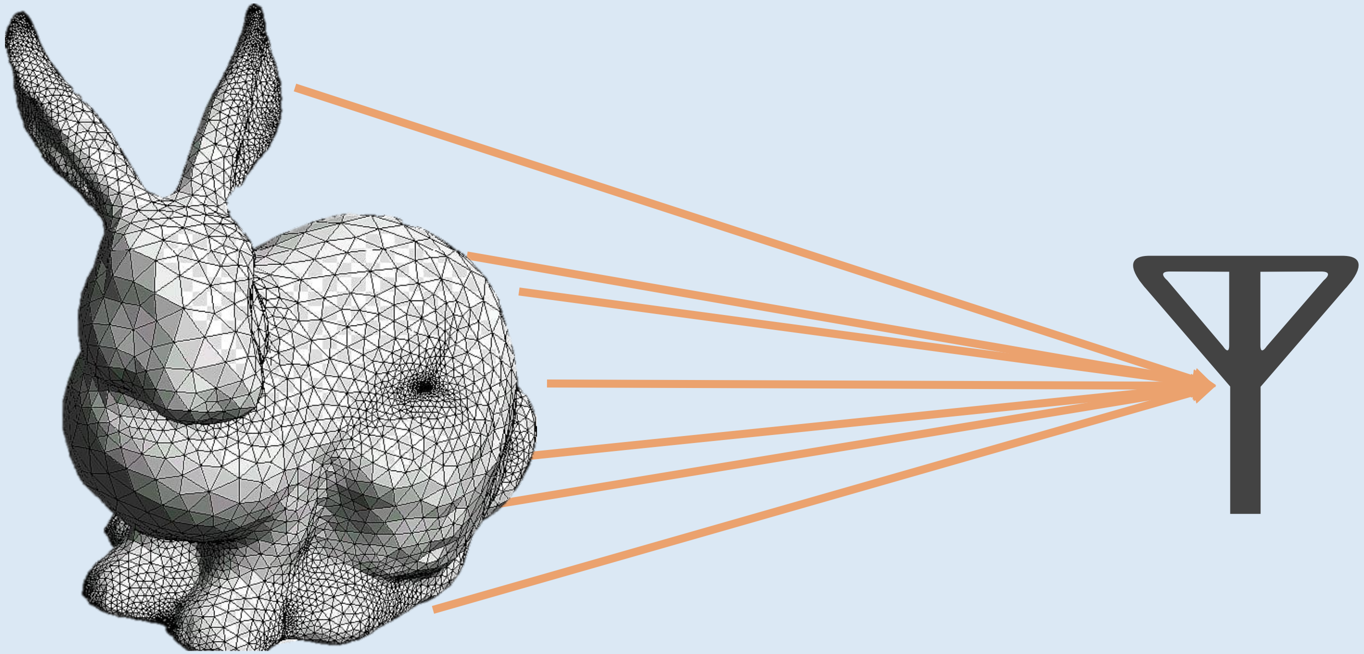


Vision computation: $\mathcal{O}(N_s N_{\mathrm{ray}} C_{\mathrm{mlp}})$

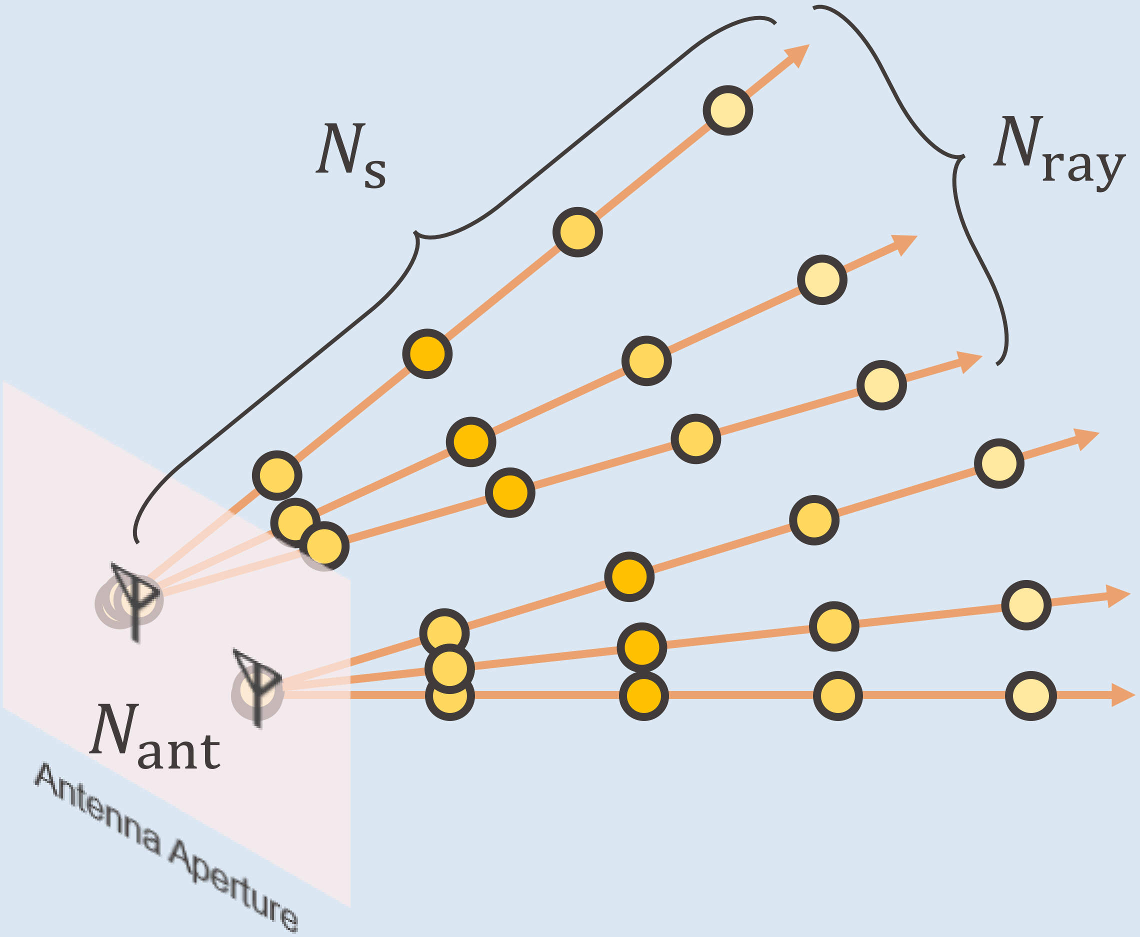
Pinhole model is not applicable in RF



Antenna model



Antenna model



RF computation: $\mathcal{O}(N_{ray} N_s N_{ant} C_{mlp})$

Vision computation: $\mathcal{O}(N_s N_{ray} C_{mlp})$

Revisit: Vision Rendering

$$\alpha_i = 1 - \frac{\Phi\left(\mathrm{sdf}\left(\mathbf{x}(t_{i+1}, \mathbf{v})\right)\right)}{\Phi\left(\mathrm{sdf}\left(\mathbf{x}(t_i, \mathbf{v})\right)\right)}$$

$$T_i = (1 - \alpha_1)(1 - \alpha_2) \cdots (1 - \alpha_{i-1})$$

$$\mathbf{c}(\mathbf{o}, \mathbf{v}) = \sum_{i=1}^{n} w_i \mathbf{c}_i = \sum_{i=1}^{n} T_i \alpha_i \mathbf{c}_i$$

ρ: Density

$I(\mathbf{c})$: Light (Color)

w: Weight

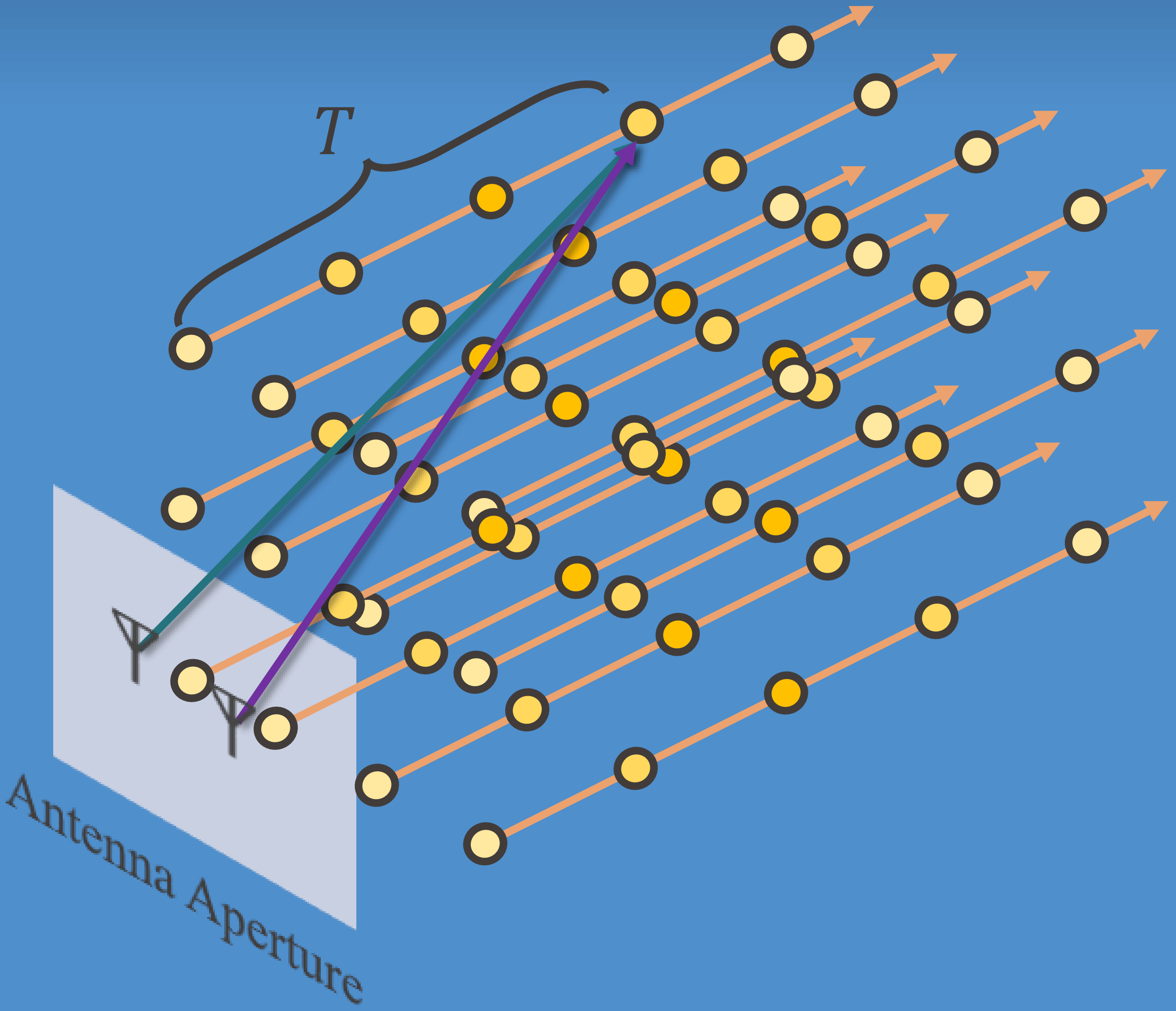
t: Ray Parameter

ΔT: Transmittance

T: Accumulated Transmittance

α: Opacity

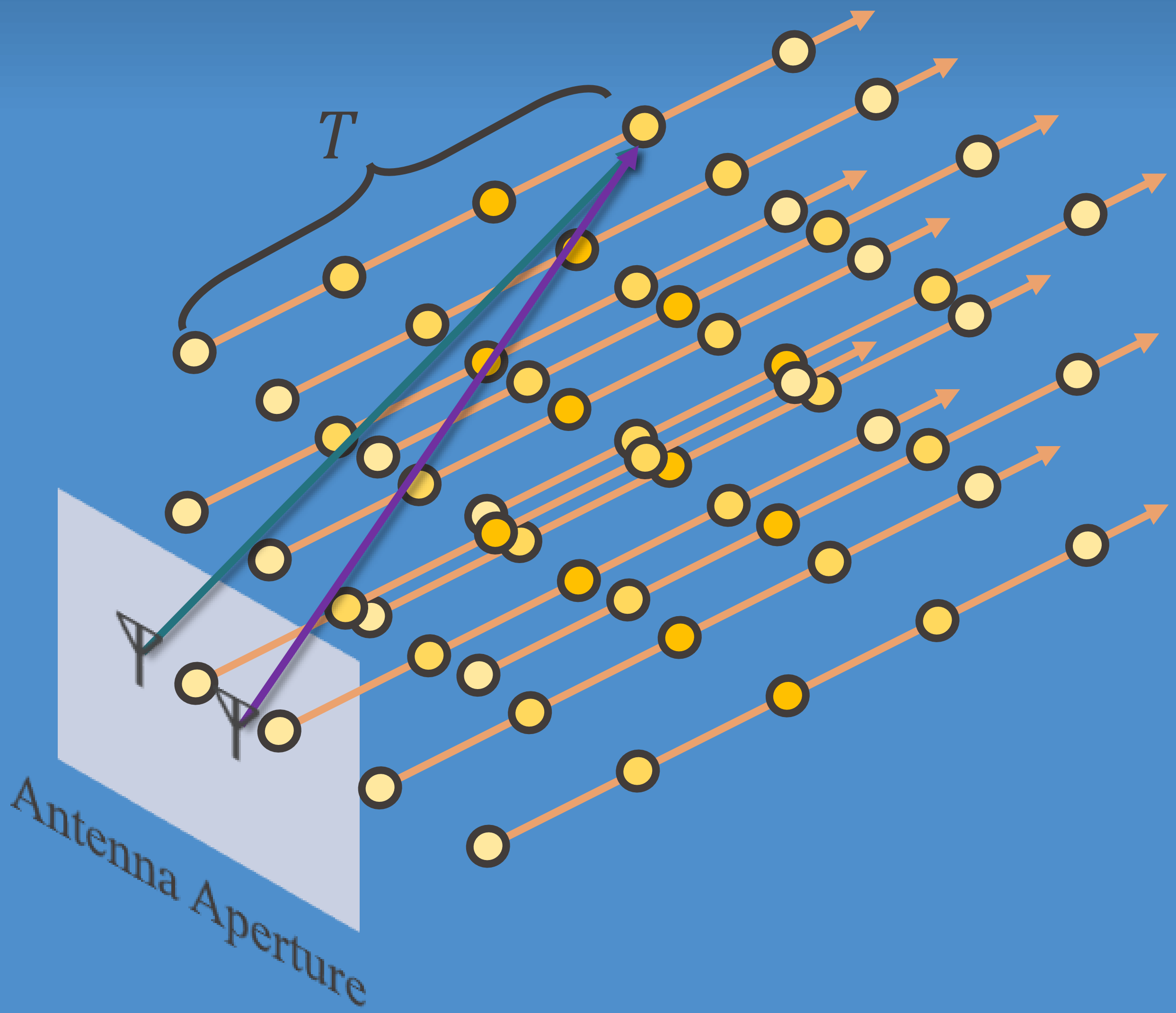
RF Rendering



1. Calculate α_i, T_i for each sample
2. Although α_i, T_i are ray dependent, we approximately using the same α_i, T_i for all antennas.

$$\mathcal{O}(N_{\text{ray}} N_s C_{\text{mlp}})$$

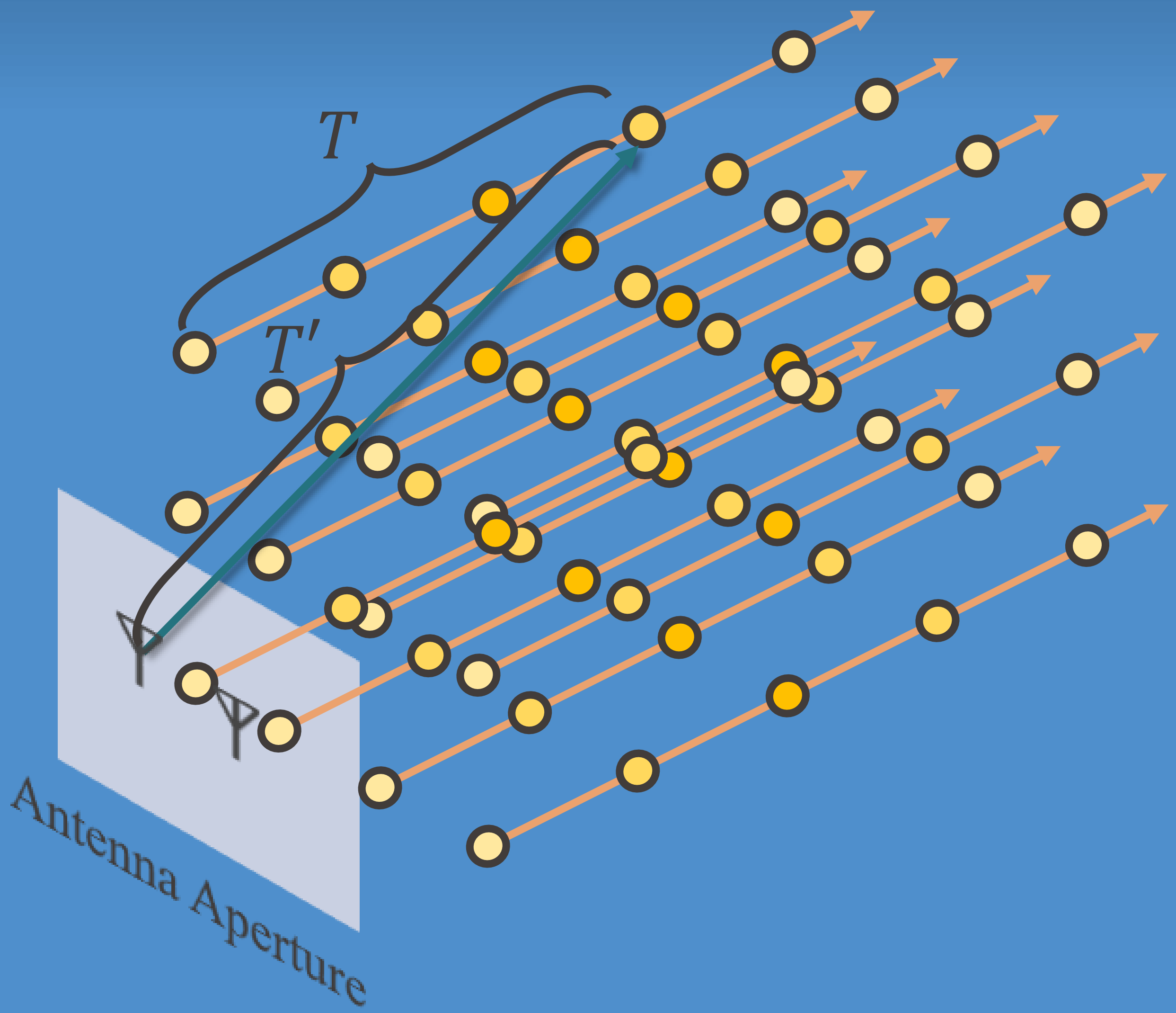
RF Rendering



1. Calculate α_i, T_i for each sample
2. Although α_i, T_i are ray dependent, we approximately using the same α_i, T_i for all antennas.

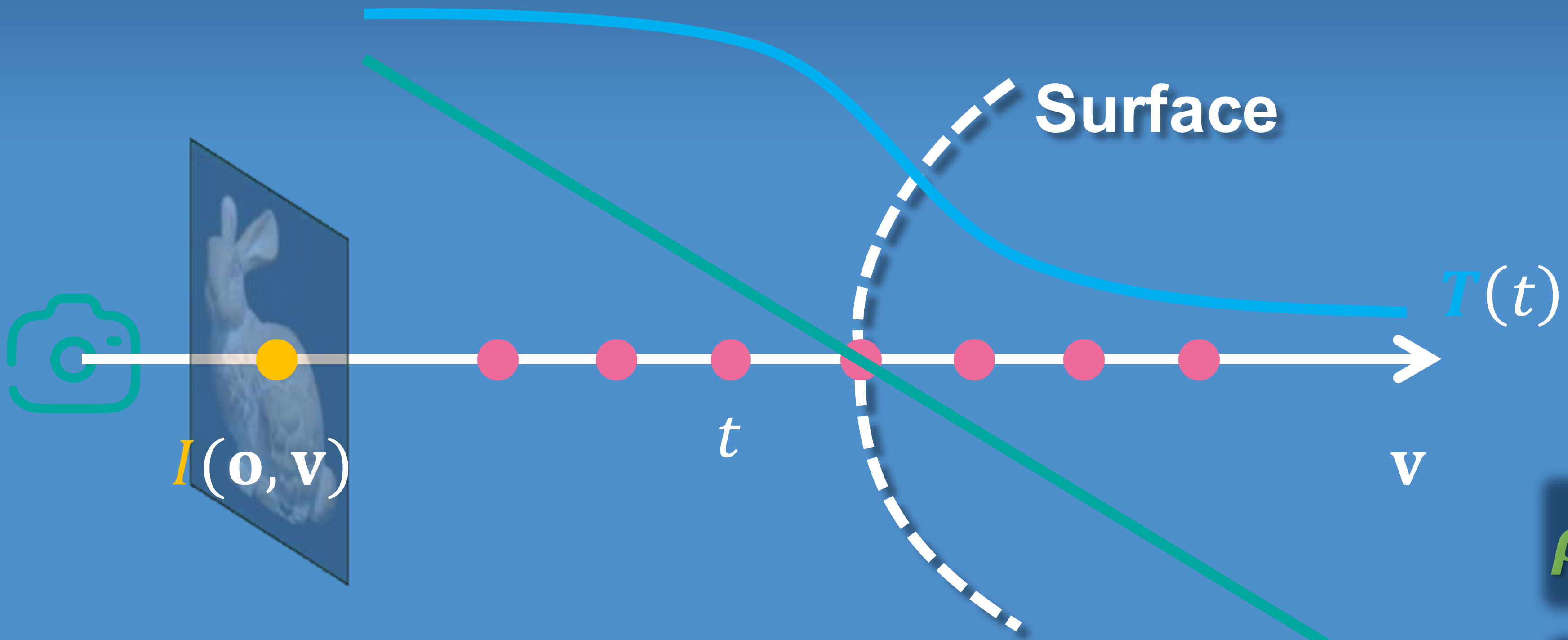
$$\mathcal{O}(N_{\text{ray}} N_s C_{\text{mlp}})$$

How to adjust Transmittance?



$$T \neq T'$$

Remember: vision assumption



Core Assumption:

$$T(t) = \mathrm{Sigmoid}\big(\mathrm{sdf}(\mathbf{x})\big)$$

"the surface is simply a plane that approaches infinitely far off the camera"

ρ: Density

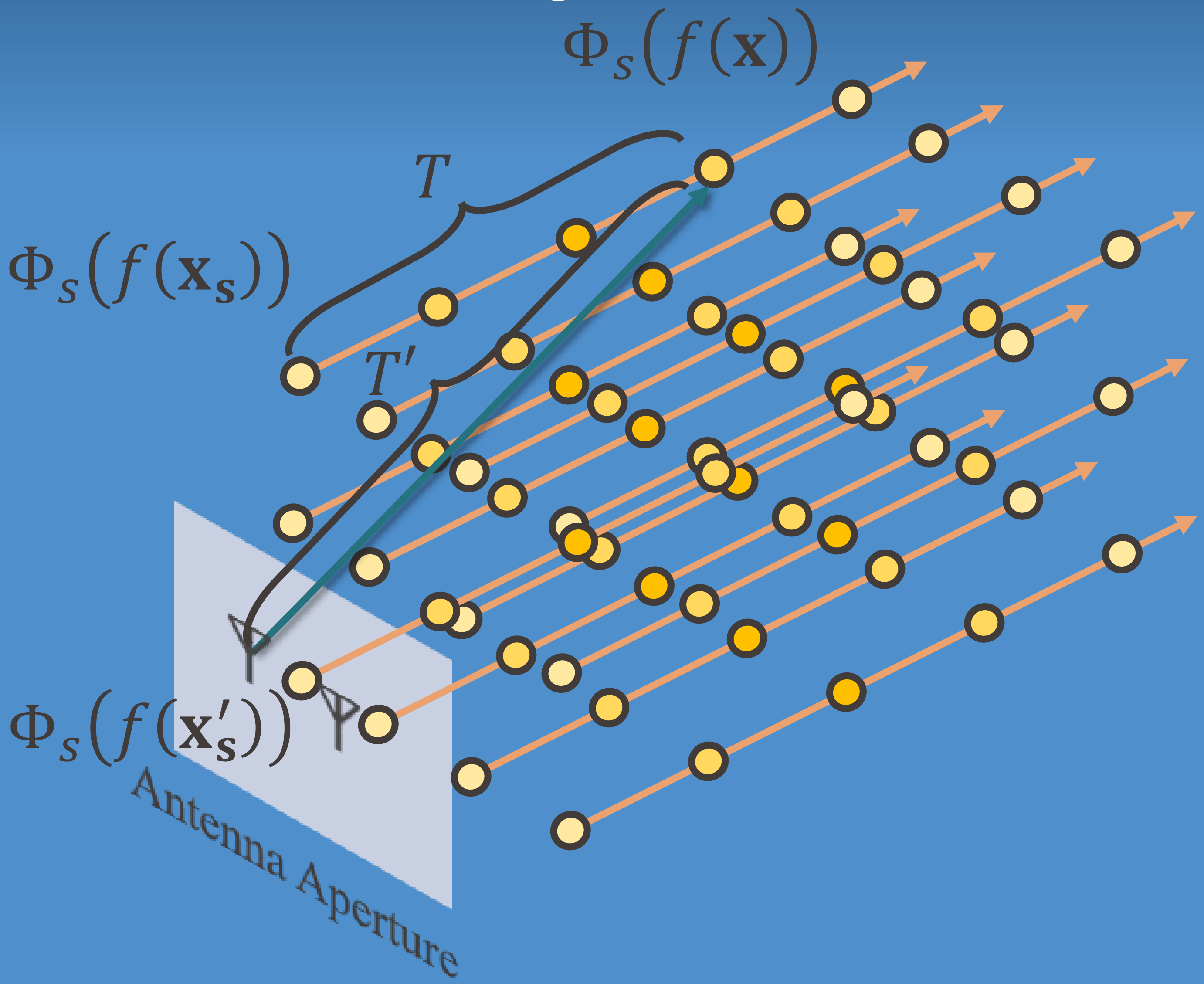
$I(\mathbf{c})$: Light (Color)

t: Ray Parameter

T: Accumulated Transmittance

Wang, Peng, et al. "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction." *Advances in neural information processing systems. 2021*

How to adjust Transmittance?



$$T = \Phi_s(f(\mathbf{x})) - \Phi_s(f(\mathbf{x_s}))$$
$$T' = \Phi_s(f(\mathbf{x})) - \Phi_s(f(\mathbf{x_s}'))$$

$$T' = T - \Phi_s(f(\mathbf{x_s})) + \Phi_s(f(\mathbf{x}'_s))$$

$$\mathcal{O}(N_{\mathrm{ray}} N_s C_{\mathrm{mlp}} + N_{\mathrm{ray}} N_{\mathrm{ant}} C_{\mathrm{mlp}})$$

Revisit: Vision Rendering

$$\alpha_i = 1 - \frac{\Phi\left(\mathrm{sdf}\left(\mathbf{x}(t_{i+1}, \mathbf{v})\right)\right)}{\Phi\left(\mathrm{sdf}\left(\mathbf{x}(t_i, \mathbf{v})\right)\right)}$$

$$T_i = (1 - \alpha_1)(1 - \alpha_2) \cdots (1 - \alpha_{i-1})$$

$$\mathbf{c}(\mathbf{o}, \mathbf{v}) = \sum_{i=1}^{n} w_i \mathbf{c}_i = \sum_{i=1}^{n} T_i \alpha_i \mathbf{c}_i$$

ρ: Density

$I(\mathbf{c})$: Light (Color)

w: Weight

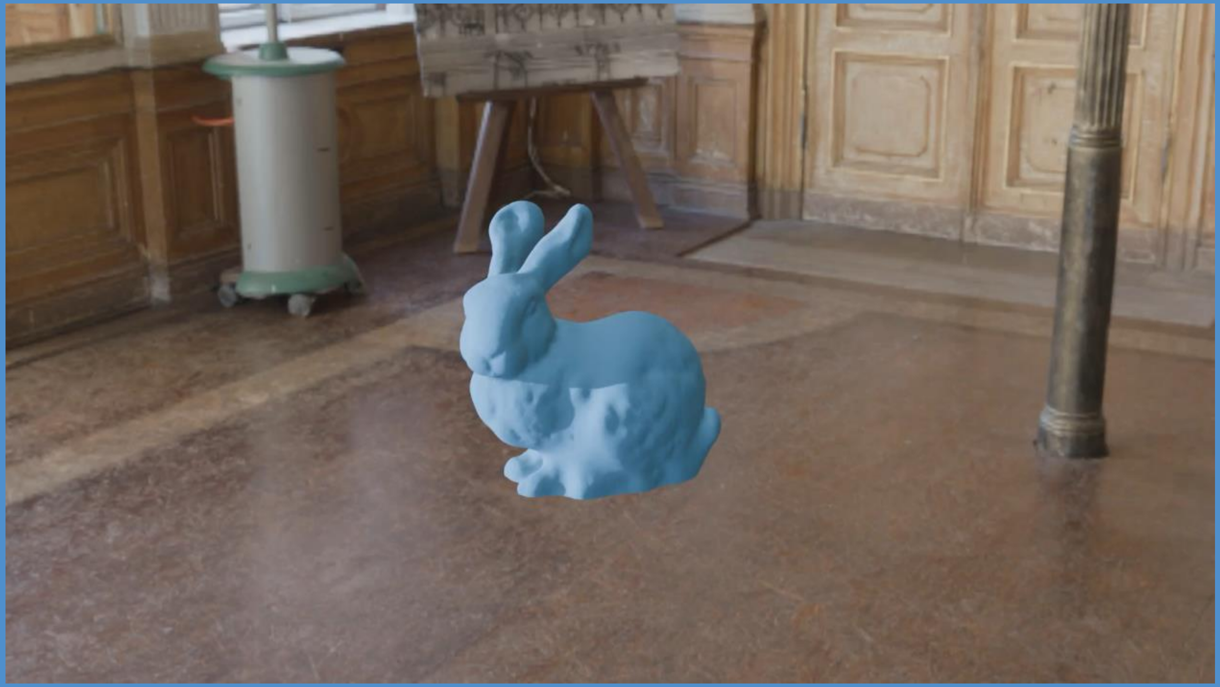
t: Ray Parameter

ΔT: Transmittance

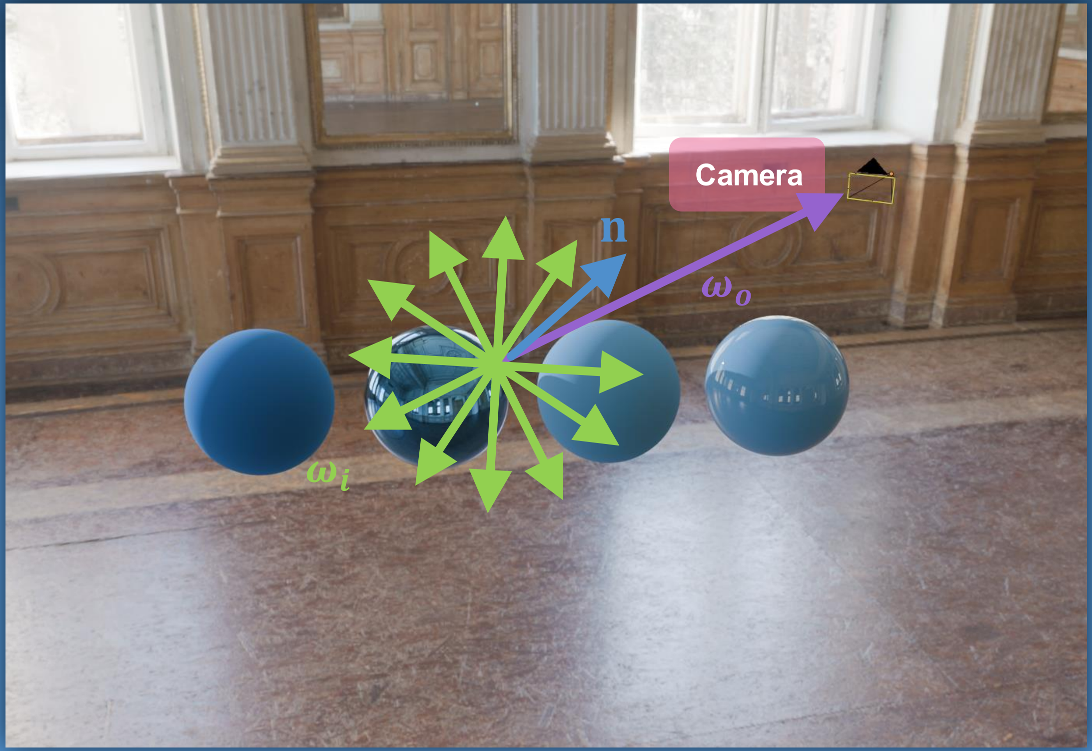
T: Accumulated Transmittance

α: Opacity

NeRF limitation: diffusion only



Reflection



$$\mathbf{c}(\omega_o) = \int_\Omega L(\omega_i) f(\omega_i, \omega_o)(\omega_i \cdot \mathbf{n})\, d\omega_i$$

L: light radiance

f: reflection model

ω_o: view direction

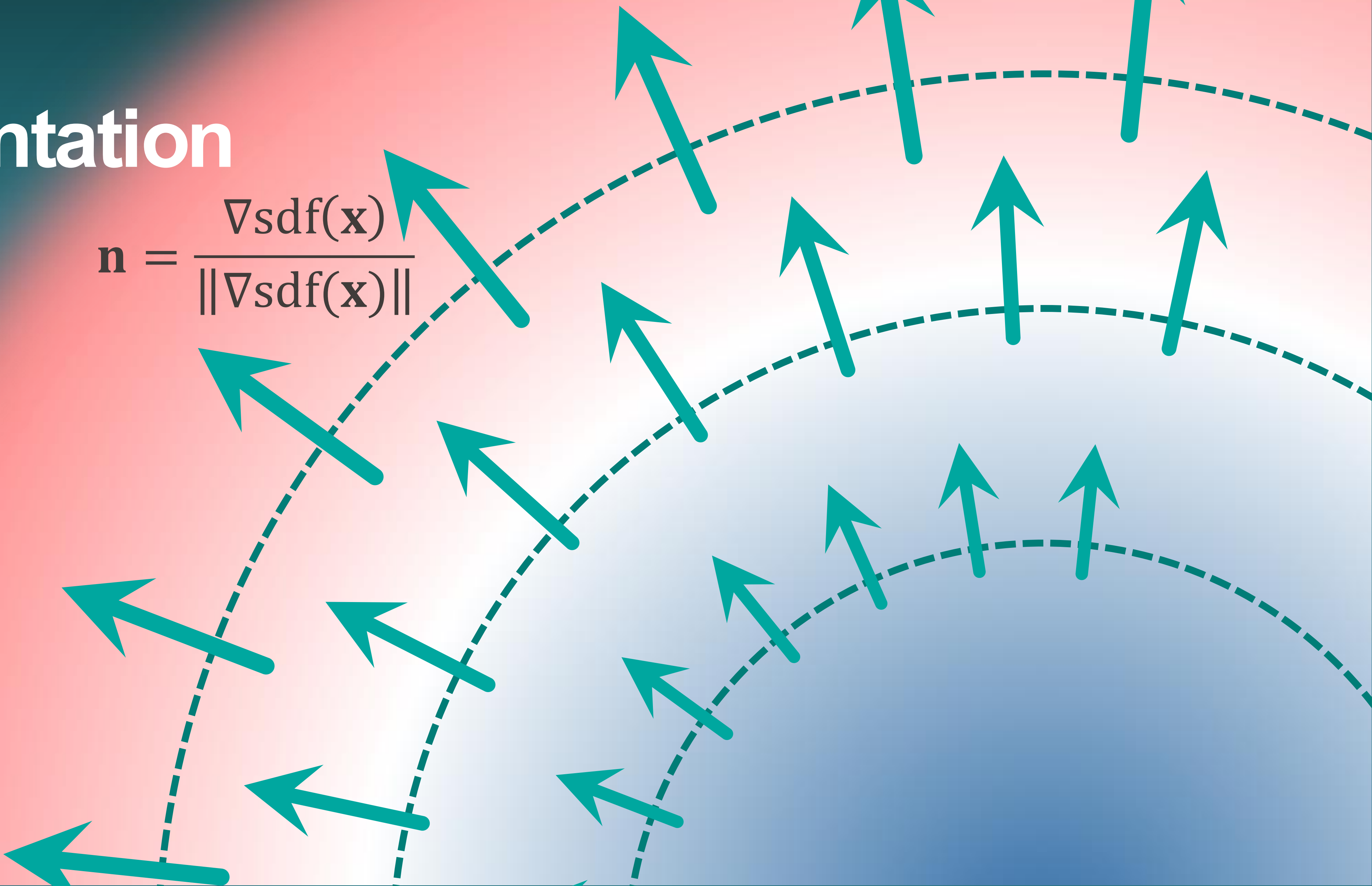
ω_i: input direction

$\mathbf{n}$: surface normal

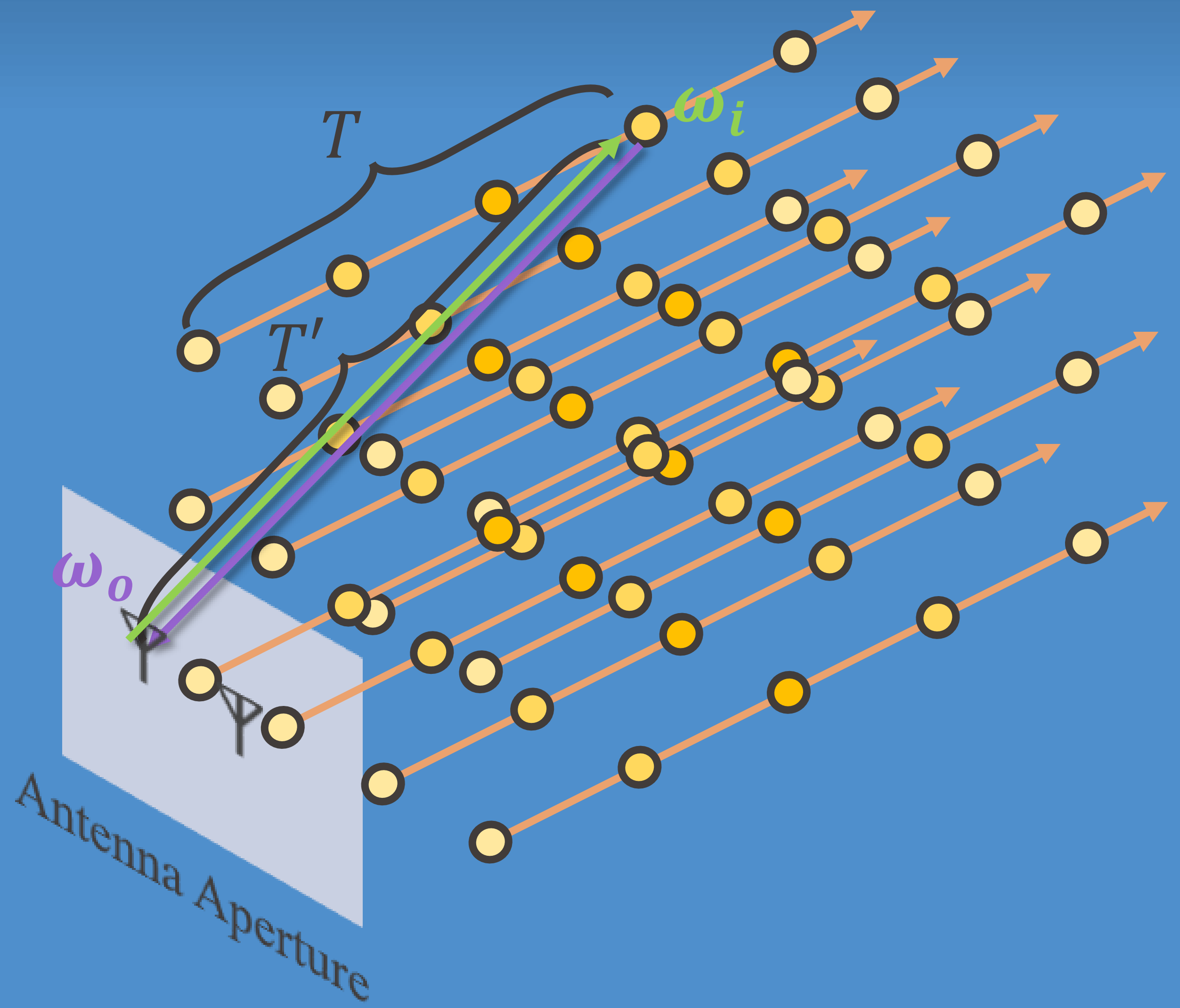
Cook, Robert L., and Kenneth E. Torrance. "A reflectance model for computer graphics." *ACM Transactions on Graphics (ToG)* 1.1 (1982): 7-24.

Normal Representation

$$\mathbf{n} = \frac{\nabla \mathrm{sdf}(\mathbf{x})}{\|\nabla \mathrm{sdf}(\mathbf{x})\|}$$

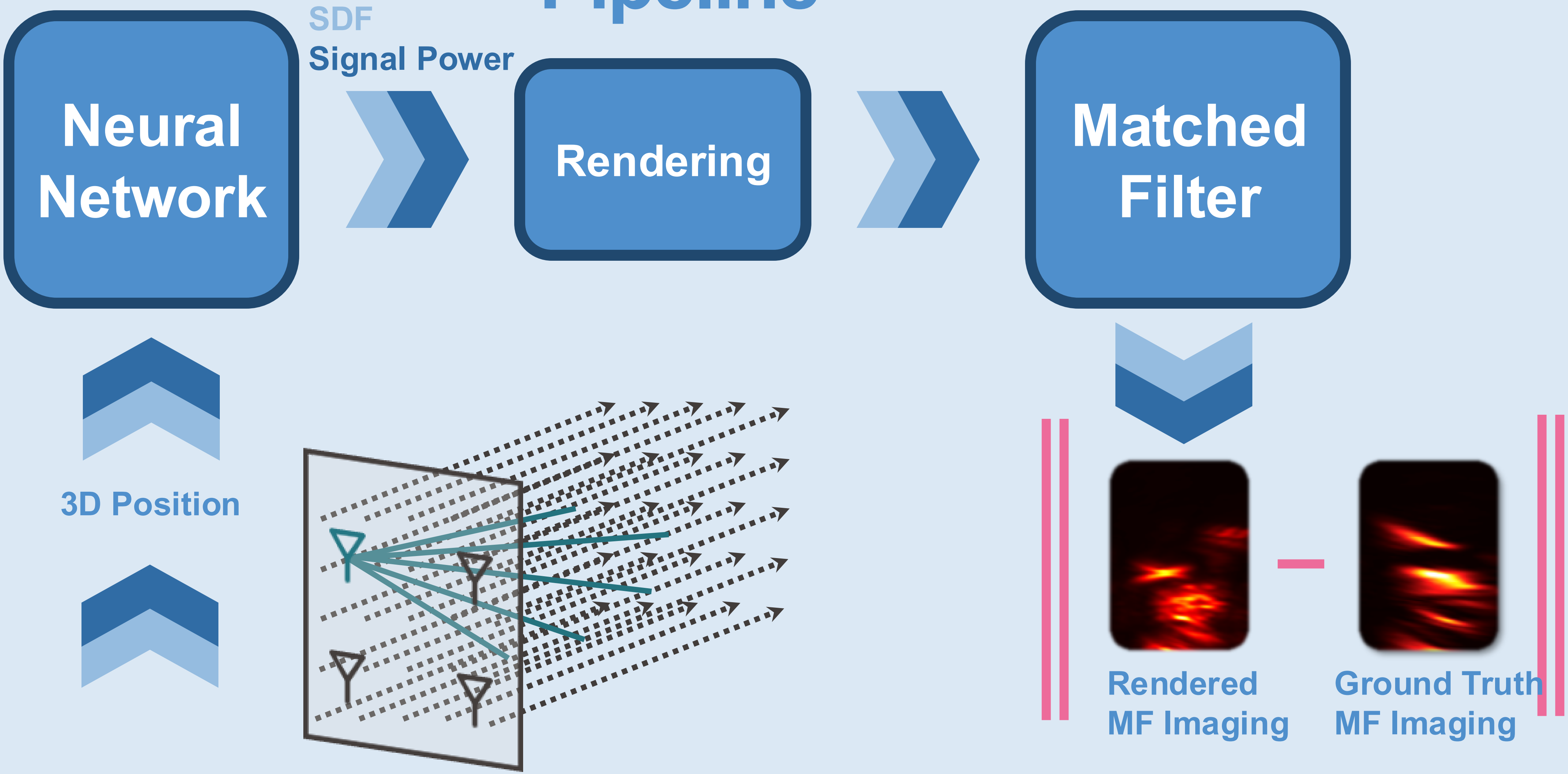


Reflection of radar



$$\mathbf{s}(\omega_o) = L(\omega_i)(\omega_i \cdot \mathbf{n})$$

Pipeline


Neural Network
SDF
Signal Power
Rendering
Matched Filter
3D Position
Rendered MF Imaging
Ground Truth MF Imaging








Thank you for your attention !

PRESEDENTED BY

Jiachen Lu
Imaging Non-Line-of-Sight 3D geometry with Radio Frequency Signals

École
polytechnique
fédérale
de Lausanne