NeRF & Plenoxels

MEGVII 旷视

Yi Yang

2022年5月

目录

Contents

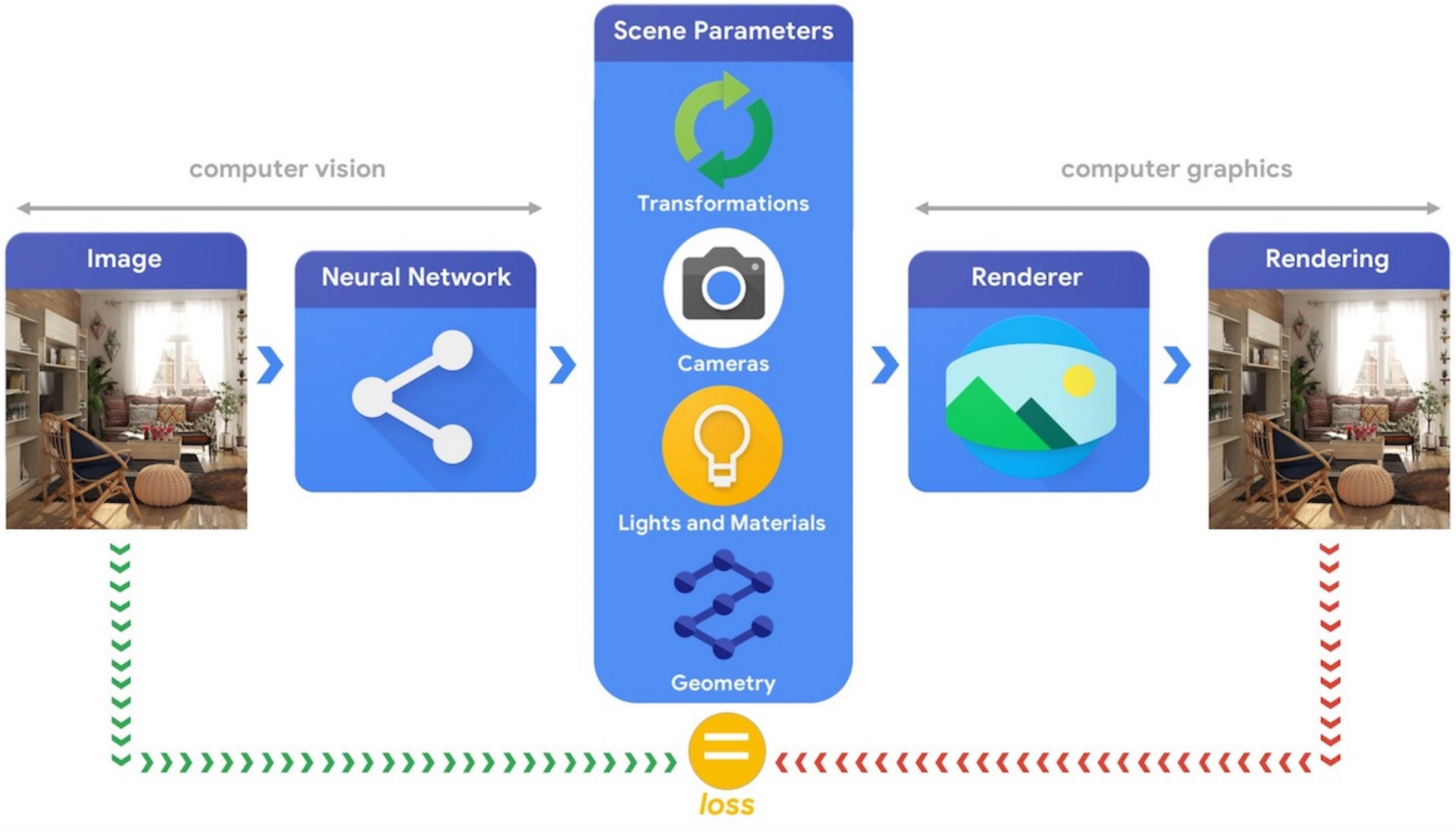
MEGVII 旷视

MEGVII 旷视

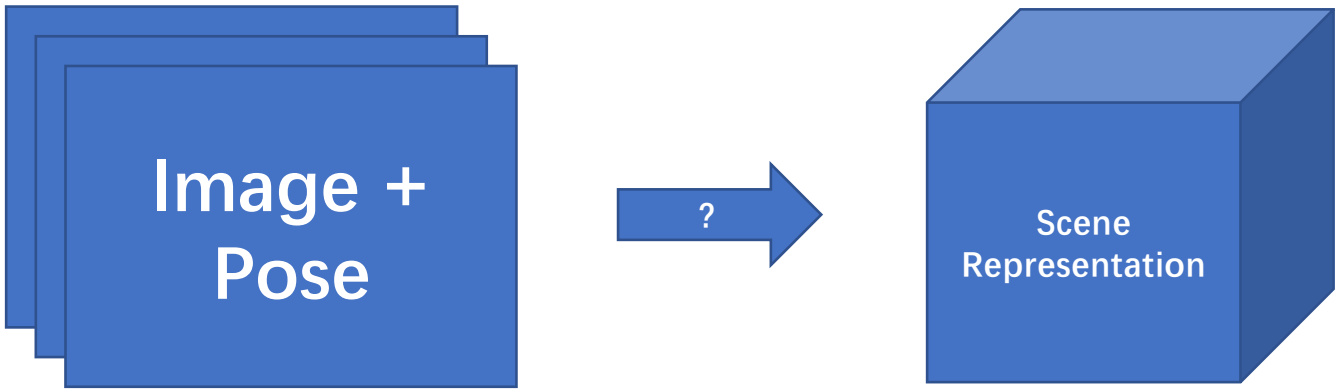
/ 00

Overview

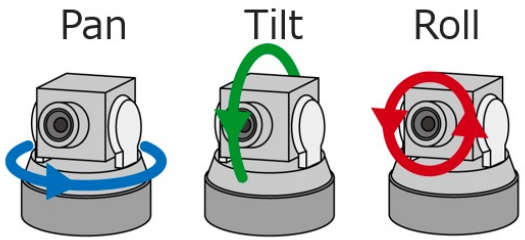
Computer Graphics meets Computer Vision



Differentiable Rendering

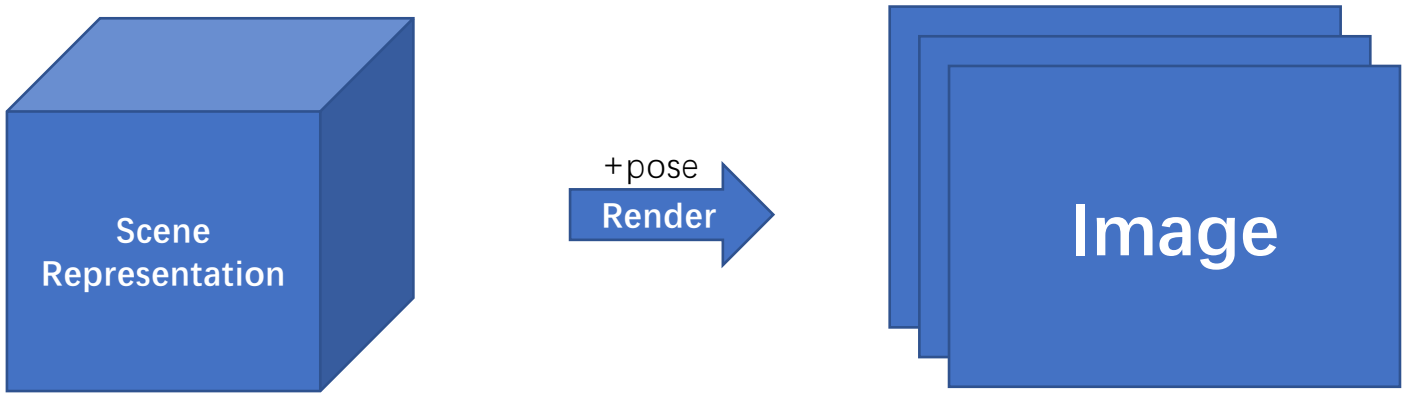


6 DoF Camera Pose

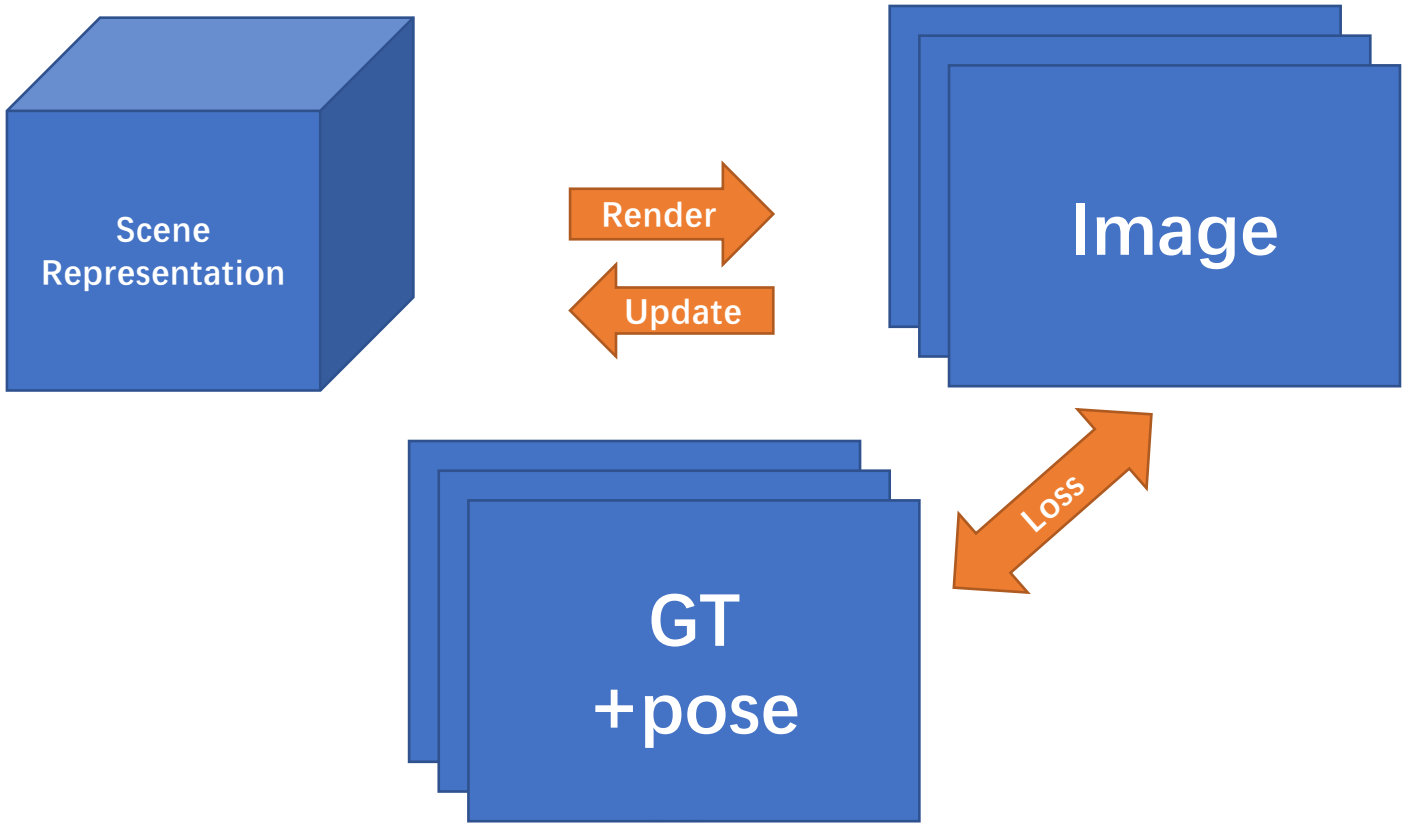


+ (x, y, z)

Differentiable Rendering

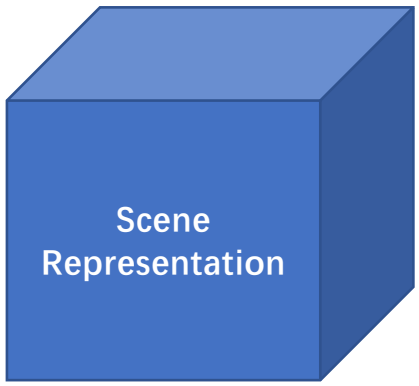


Differentiable Rendering

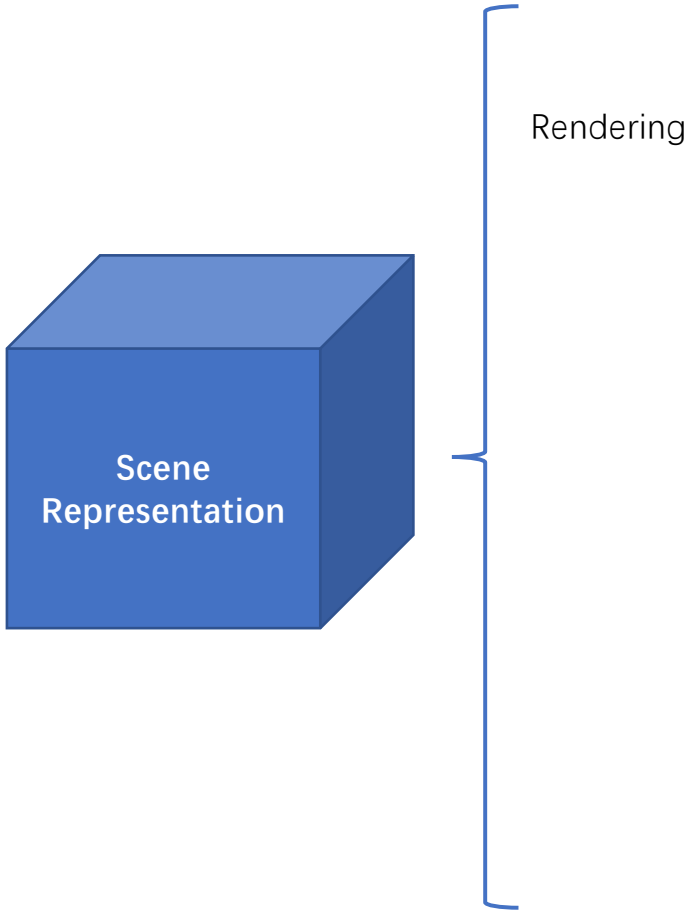


Applications

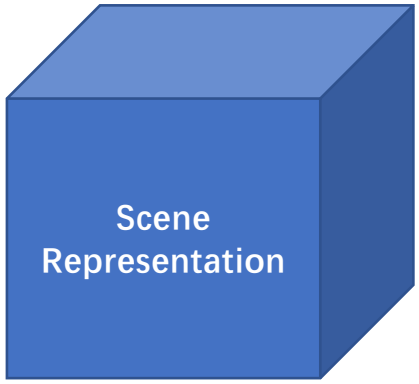
MEGVII 旷视



Applications

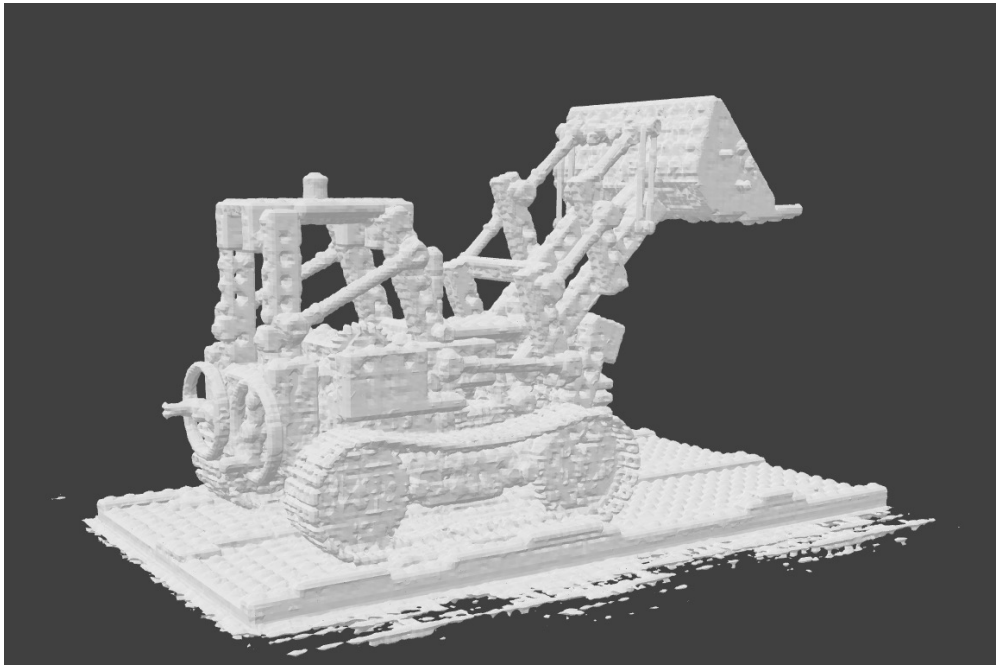


Applications



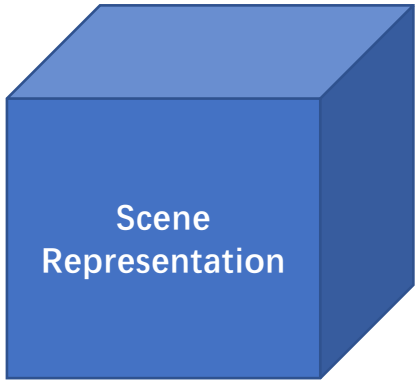
Rendering

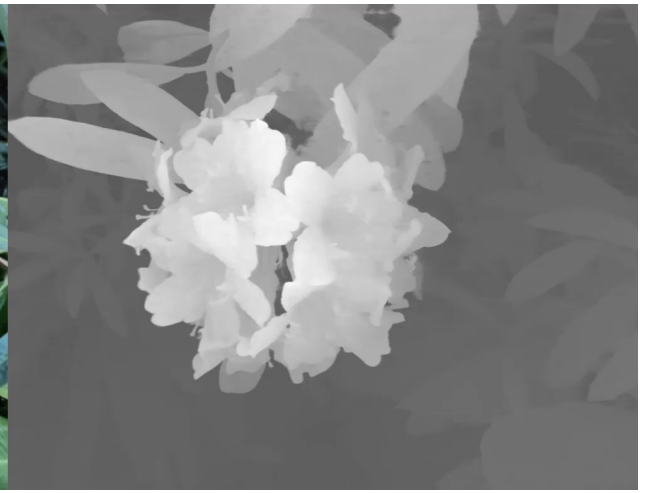
Export mesh



Applications

MEGVII 旷视





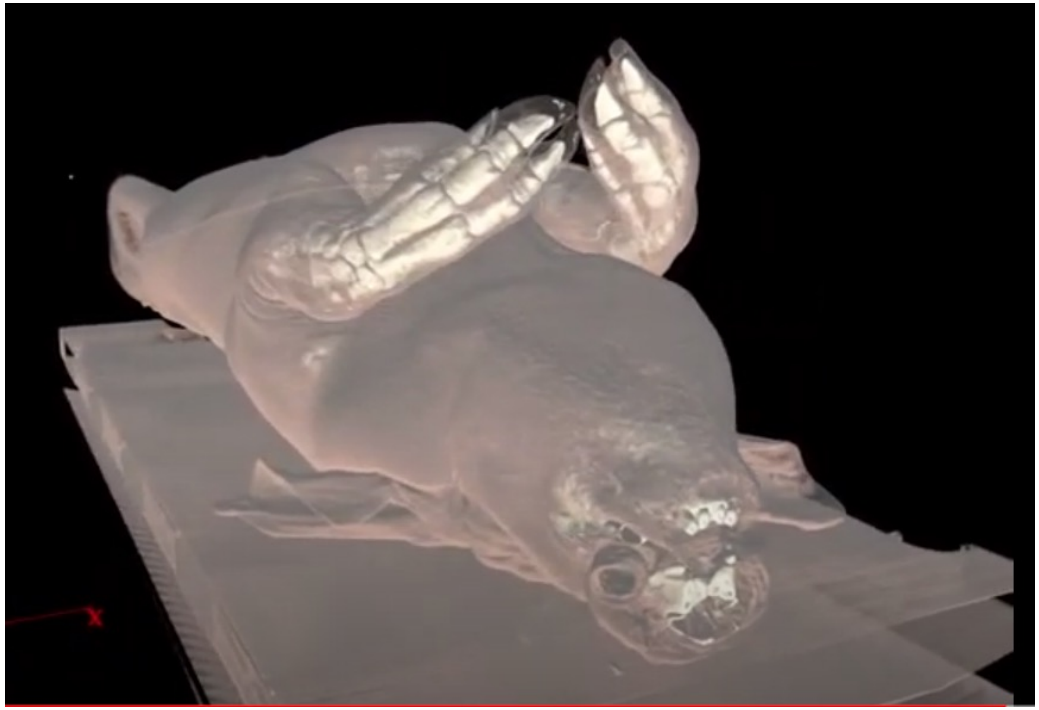
Computer Vision tasks (e.g. Depth Estimation, iNeRF)

MEGVII 旷视

/ 01

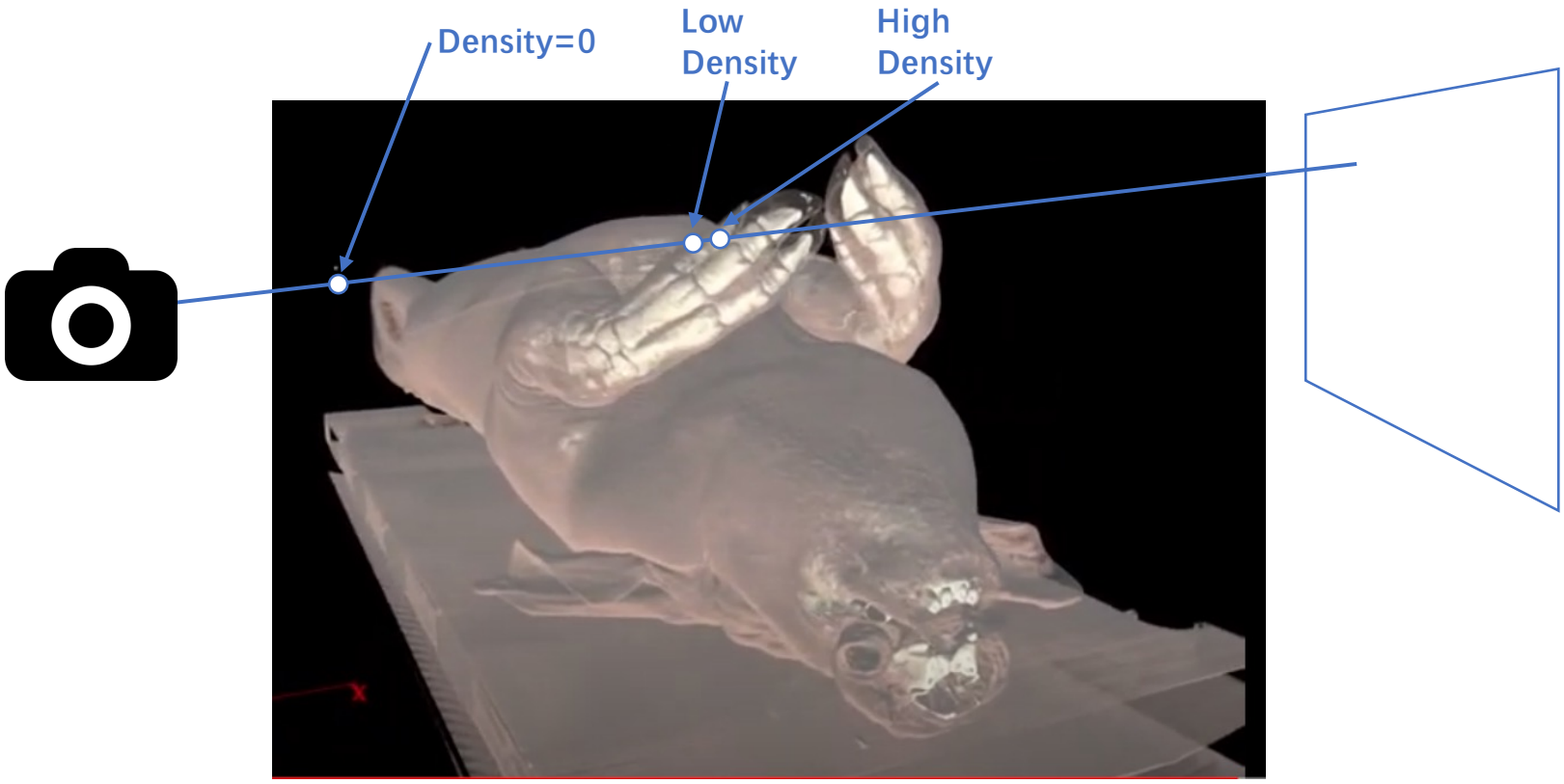
Volume Rendering

Volume Rendering



source: https://www.youtube.com/watch?v=b0uH-hqUOrk
CT Scan of a Pig, CC-BY 3.0

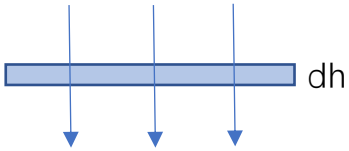
Volume Rendering



source: https://www.youtube.com/watch?v=b0uH-hqUOrk
CT Scan of a Pig, CC-BY 3.0

Volume Density

Opacity = 0.6

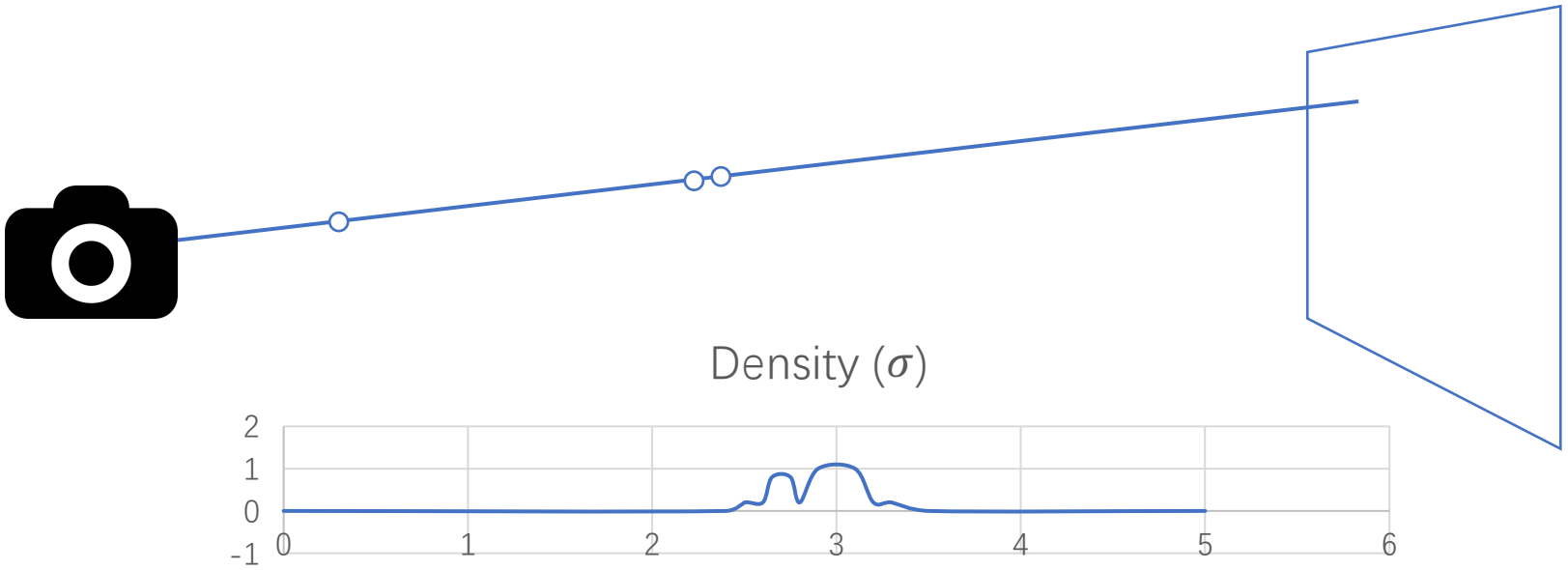


$$\sigma = \frac{d(opacity)}{dh}$$

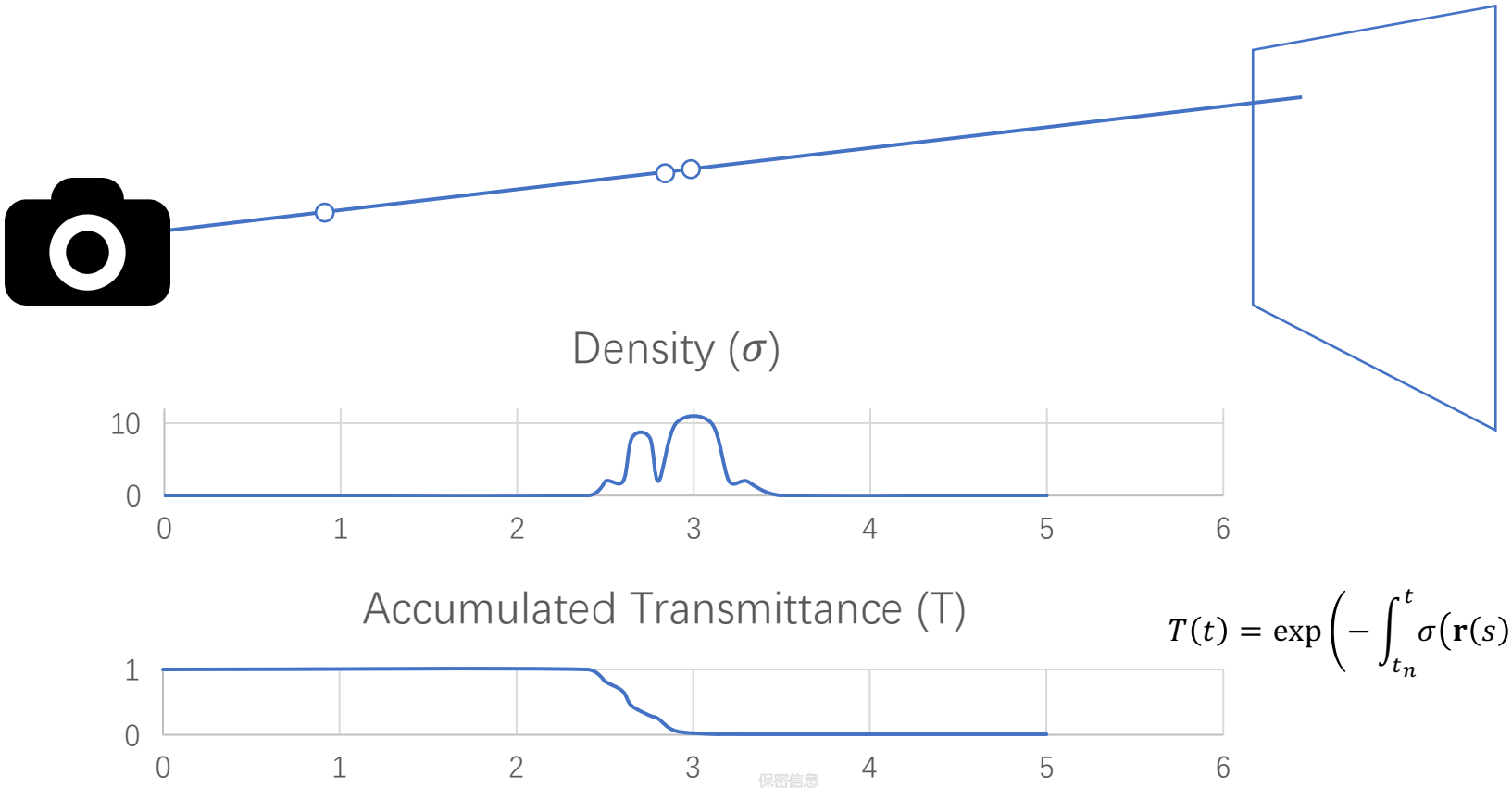
$$\log T = \int_0^{h_t} \log(1 - opacity)$$

$$T = \exp\left(-\int_0^{h_t} \sigma(h)dh\right)$$

Volume Rendering

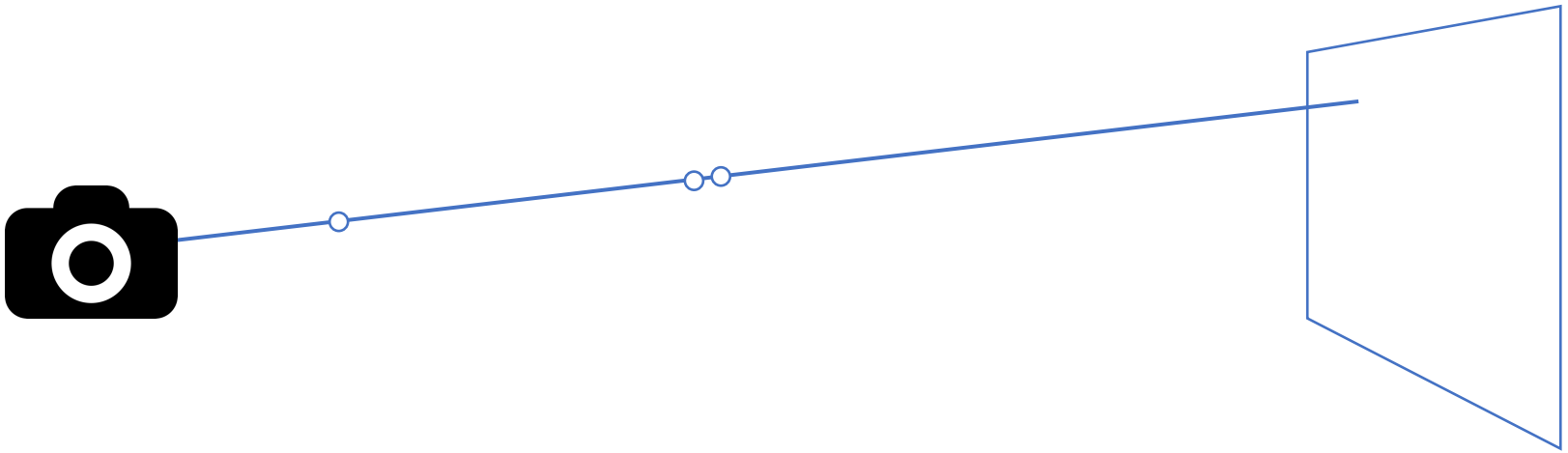


Volume Rendering



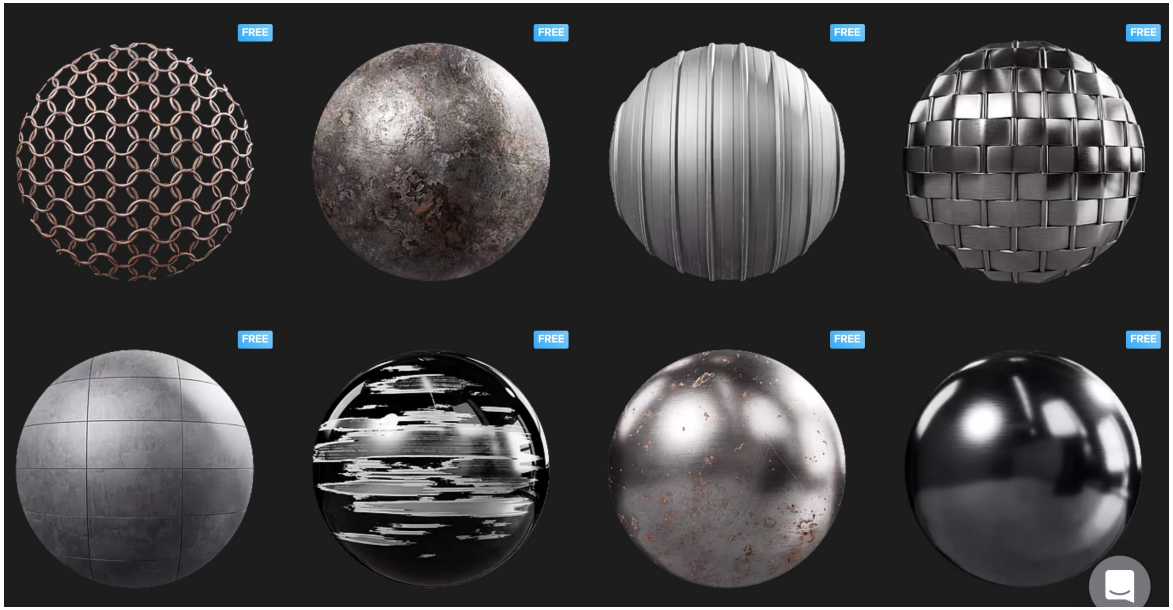
$$T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$$

Volume Rendering



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

Volume Rendering

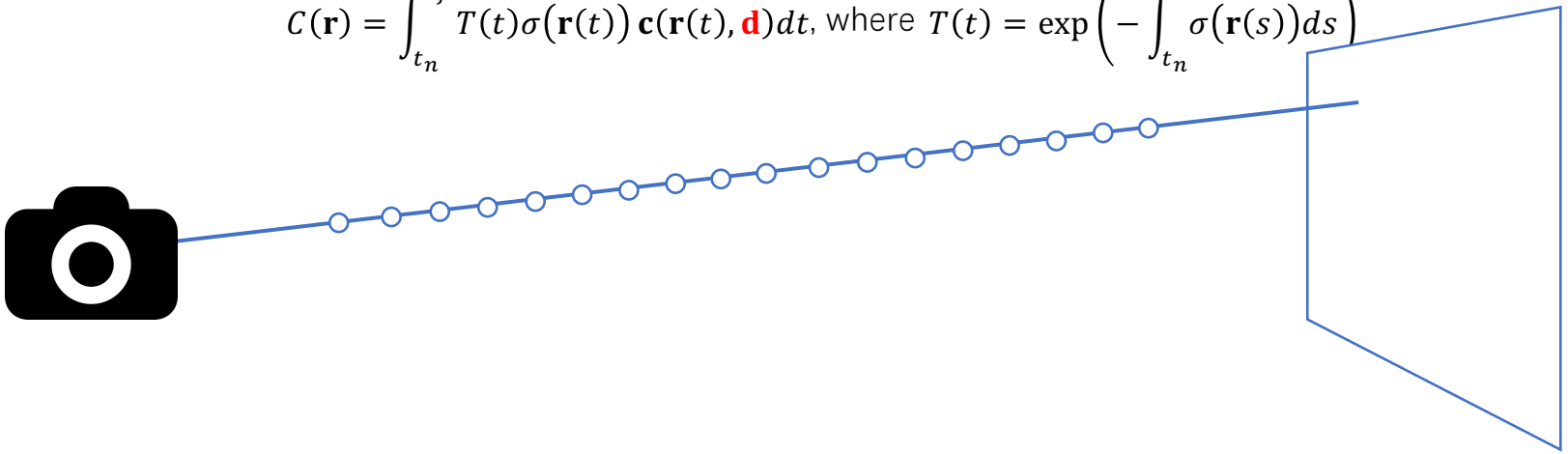


$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma\big(\mathbf{r}(t)\big)\,\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt, \text{ where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma\big(\mathbf{r}(s)\big)ds\right)$$

https://www.poliigon.com/textures/metal?credit=0

Discretization

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

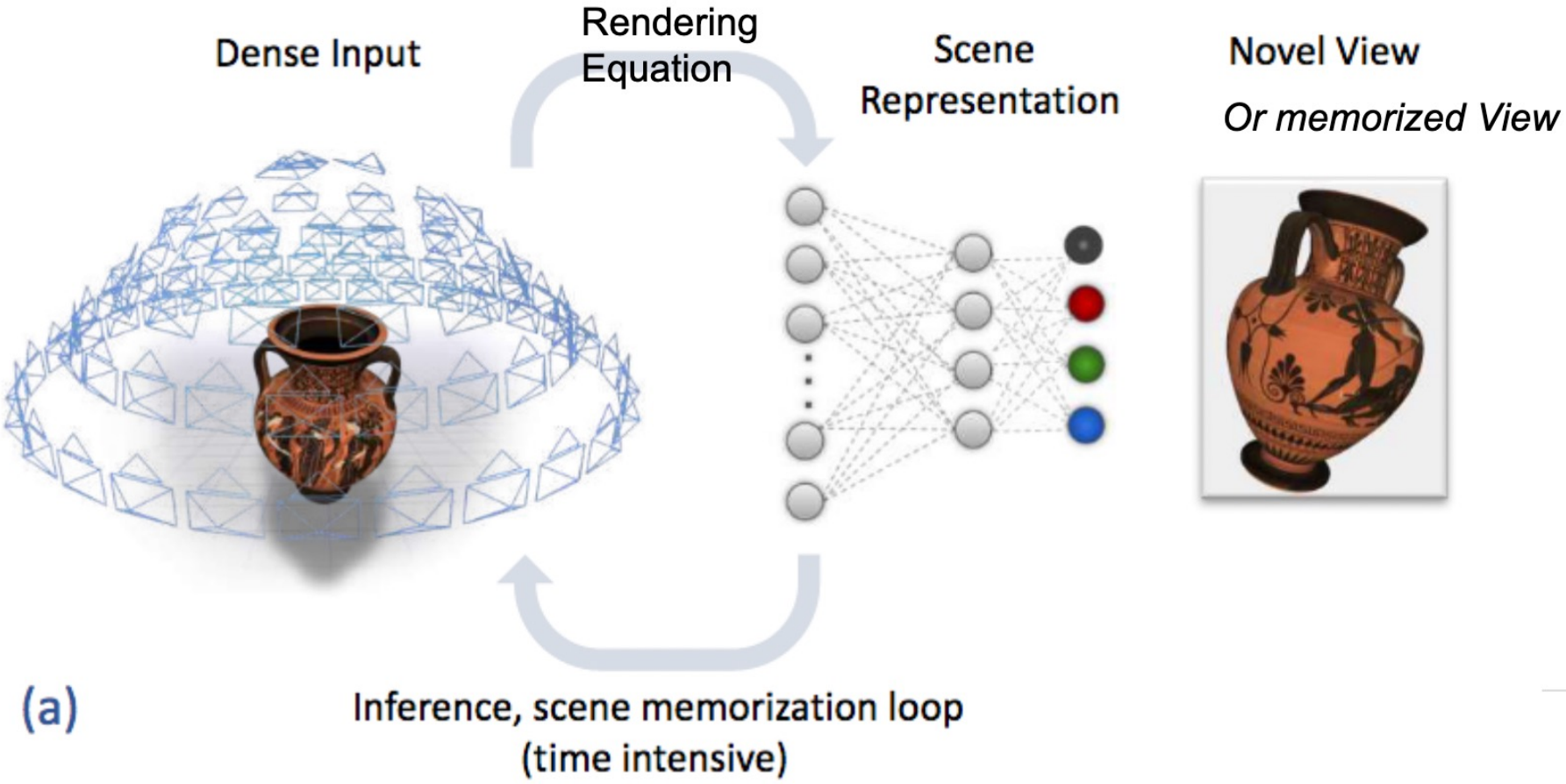


$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1 - \exp(-\sigma_i\delta_i))\mathbf{c}_i \text{ , where } T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j\delta_j\right)$$

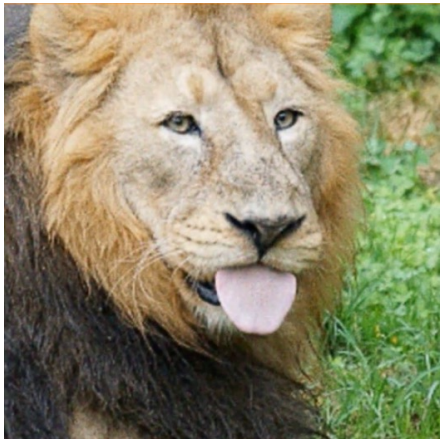
/ 02

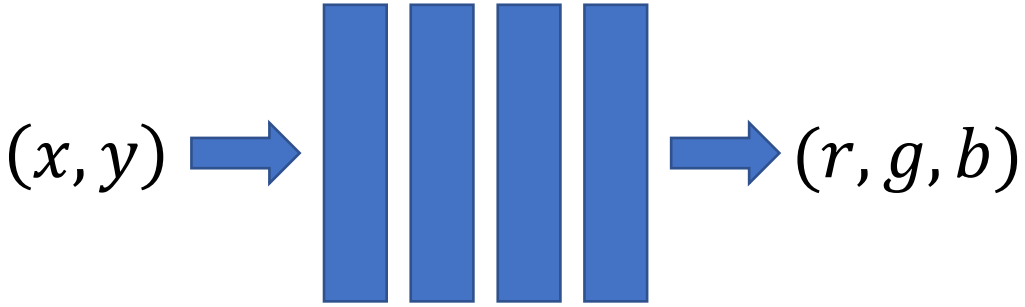
Neural Radiance Field

Neural Radiance Field



Coordinate MLP (2D)

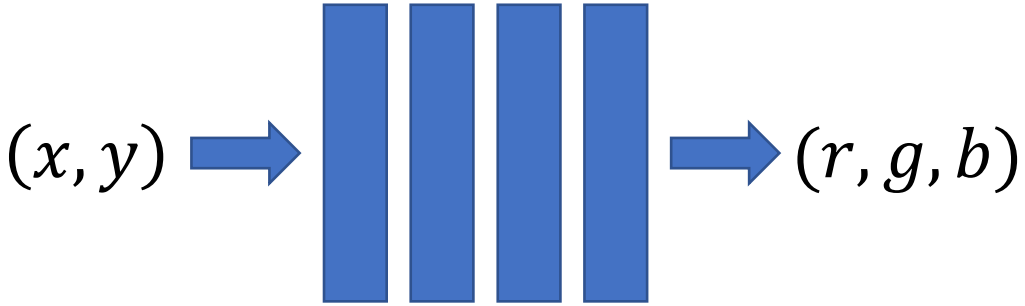




source: arXiv:2006.10739

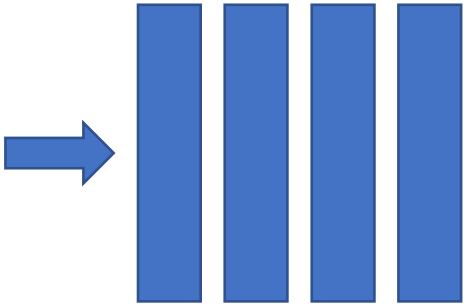
Coordinate MLP (2D)

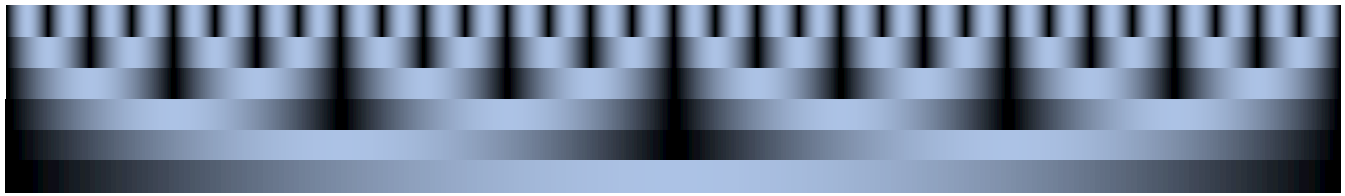
source: arXiv:2006.10739



Positional Encoding

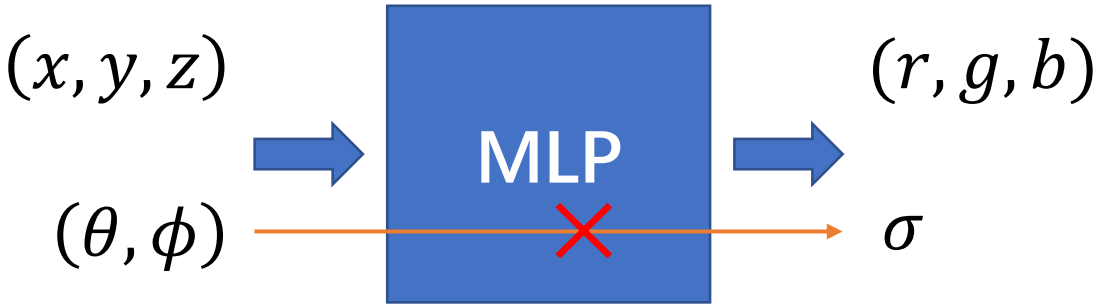
MEGVII 旷视

(x, y) → $\sin(2^0\pi x)$, $\cos(2^0\pi x)$, $\sin(2^1\pi x)$, $\cos(2^1\pi x)$, ..., $\sin(2^{L-1}\pi x)$, $\cos(2^{L-1}\pi x)$, $\sin(2^0\pi y)$, $\cos(2^0\pi y)$, $\sin(2^1\pi y)$, $\cos(2^1\pi y)$, ..., $\sin(2^{L-1}\pi y)$, $\cos(2^{L-1}\pi y)$ →  → (r, g, b)

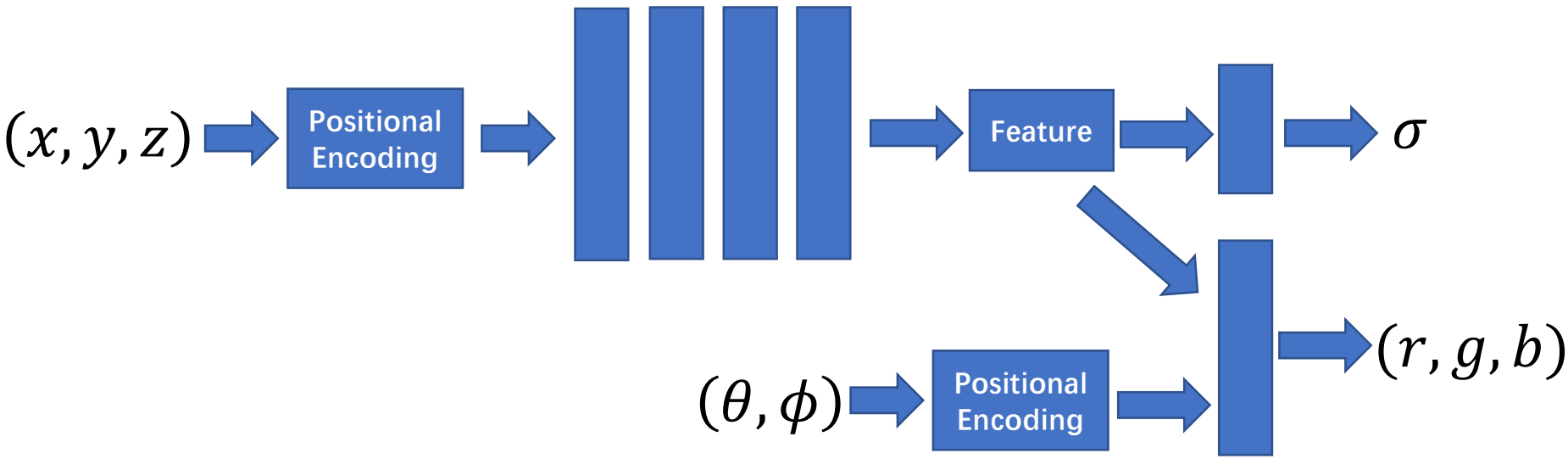


3D Coordinate MLP

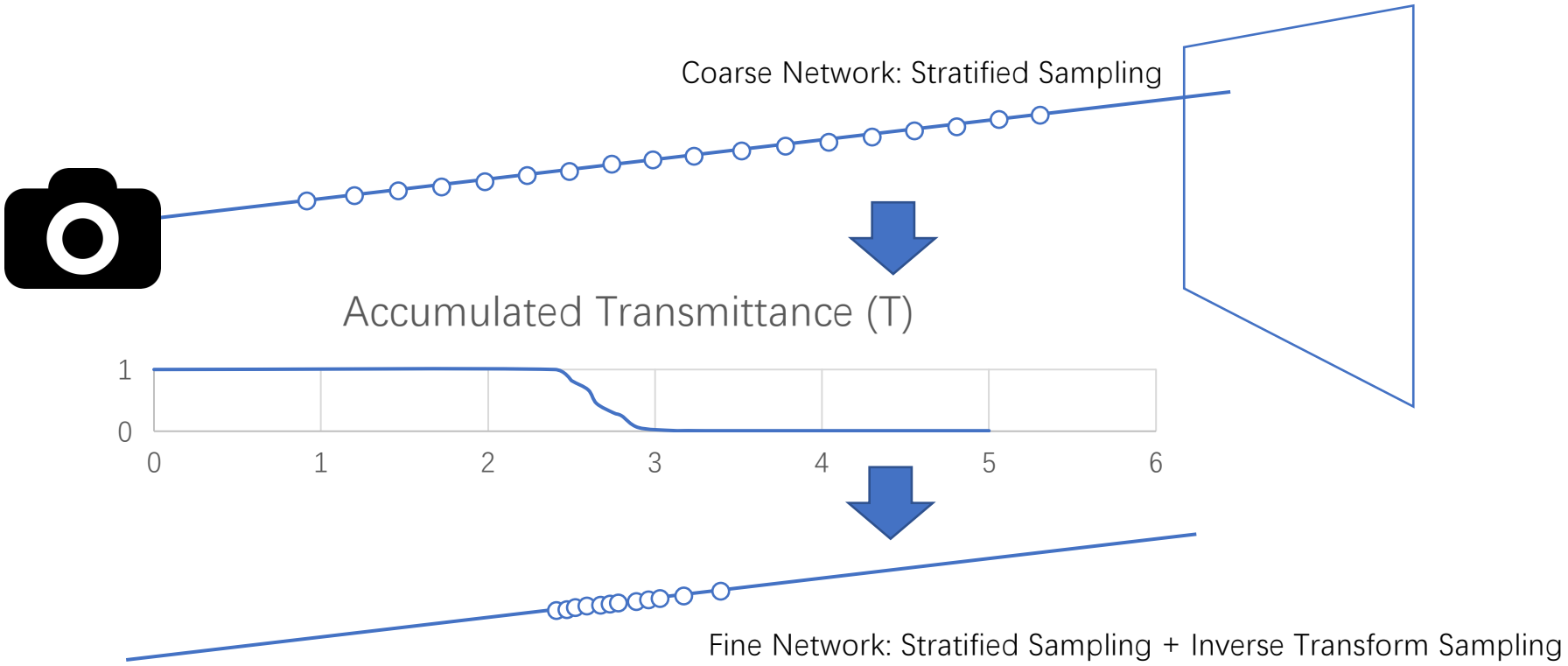
3D Coordinate MLP



The Model



Coarse to Fine



More Examples

/ 03

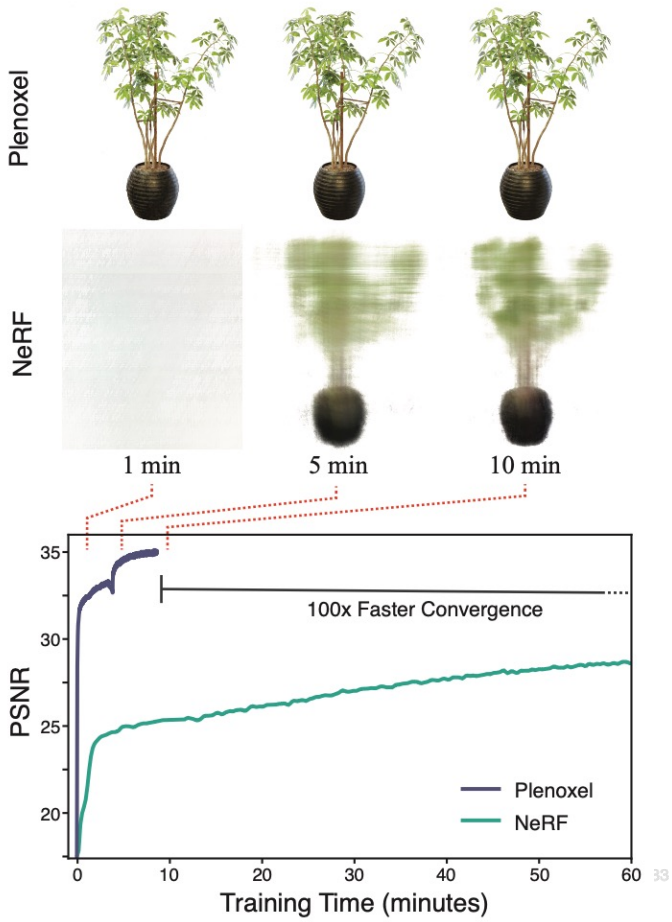
Plenoxels

Plenoxels

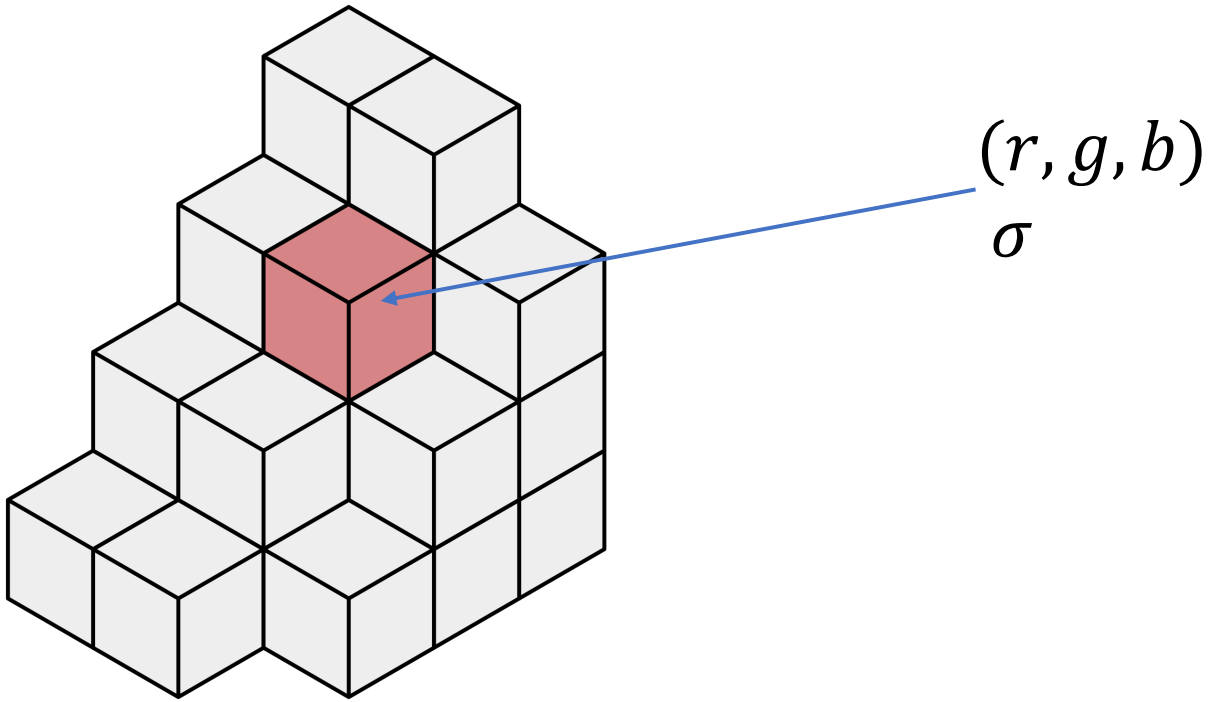
Faster training & inference

No neural networks

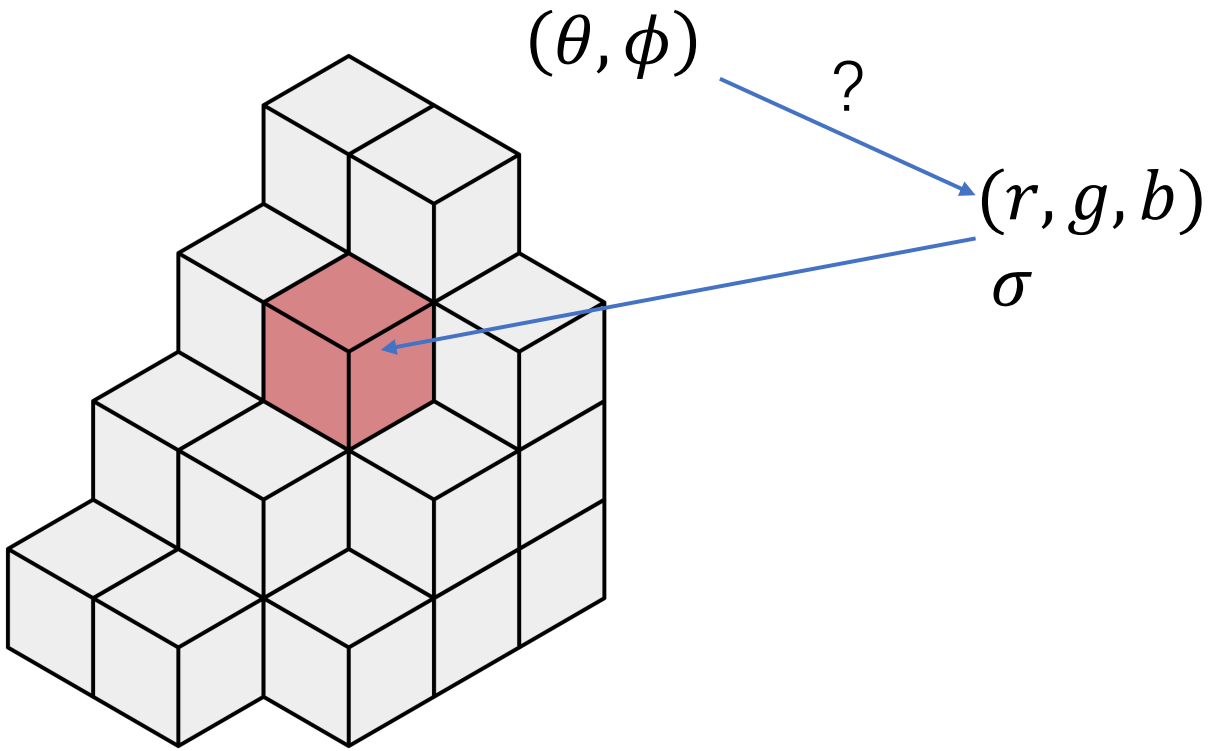
Custom CUDA implementation



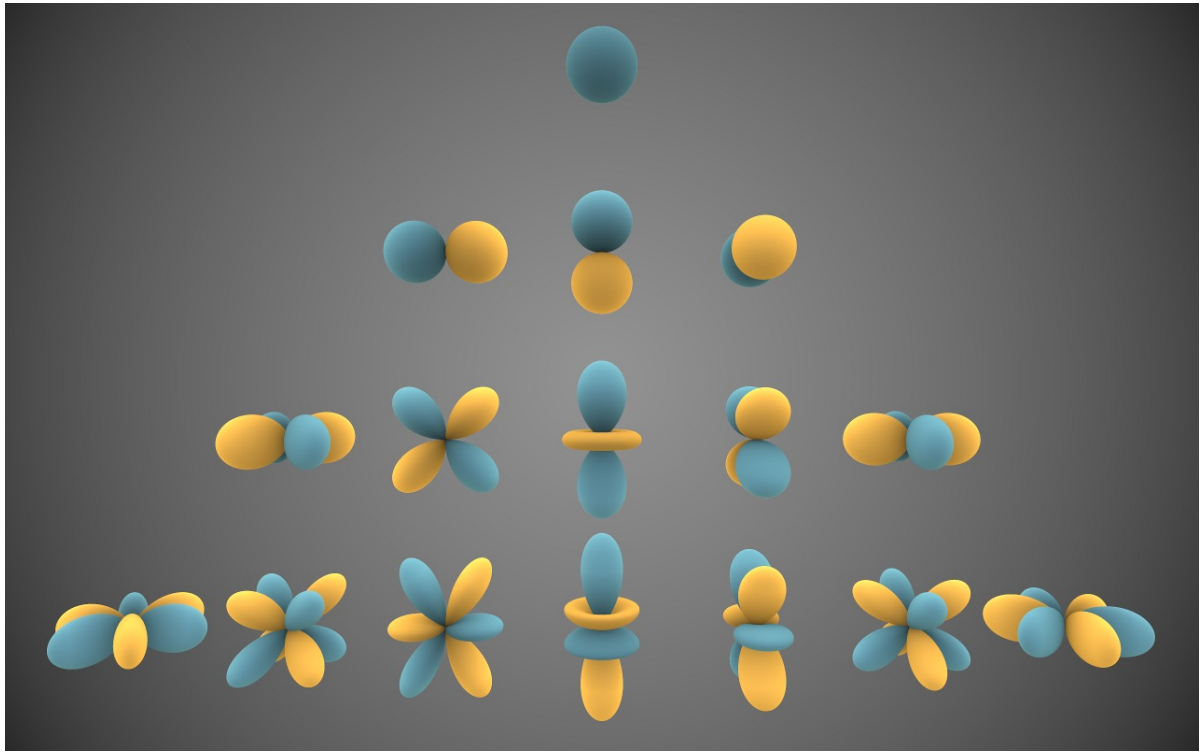
Voxels



Voxels

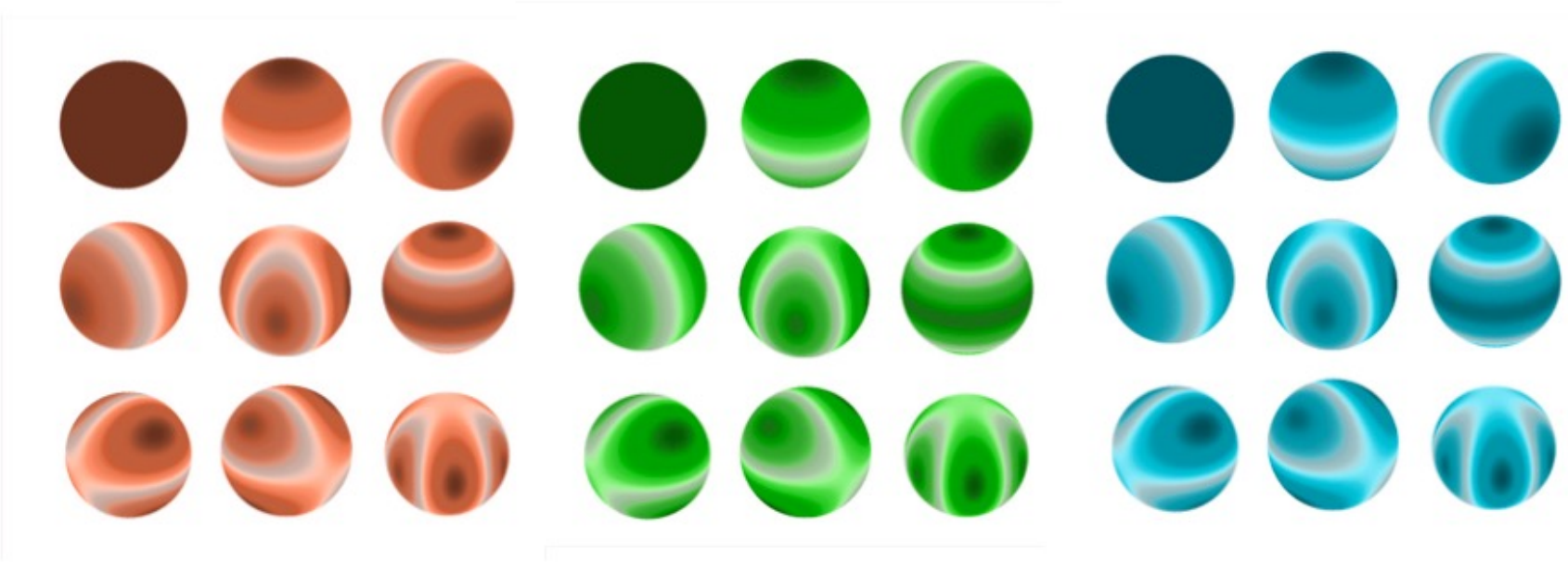


“Plenoptic”：Spherical Harmonics

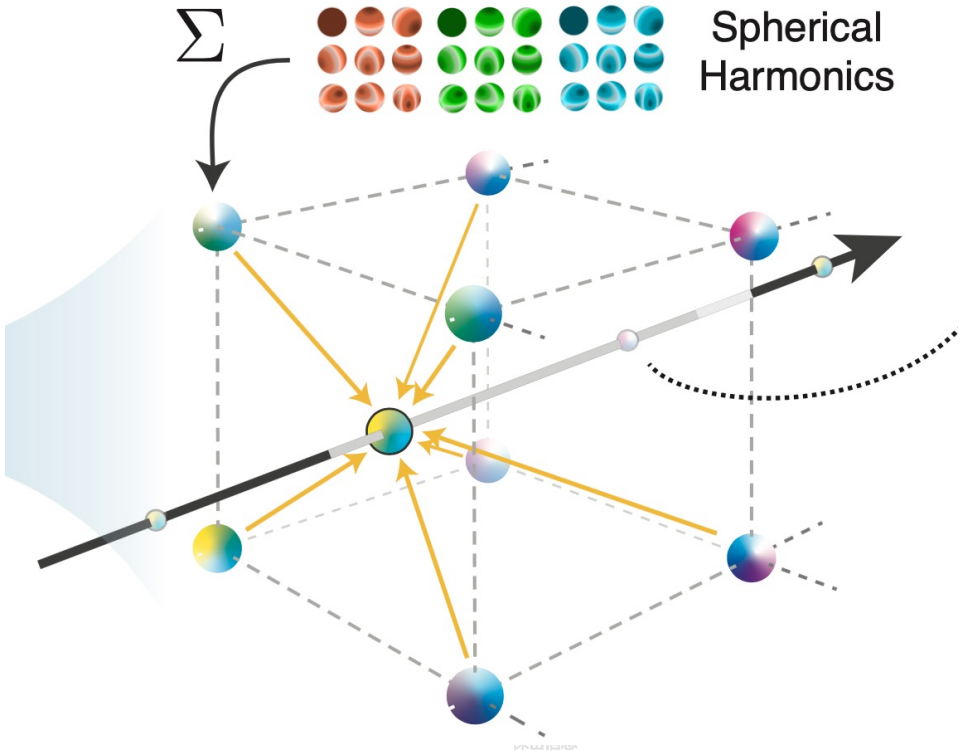


https://en.wikipedia.org/wiki/Spherical_harmonics#/media/File:Spherical_Harmonics.png

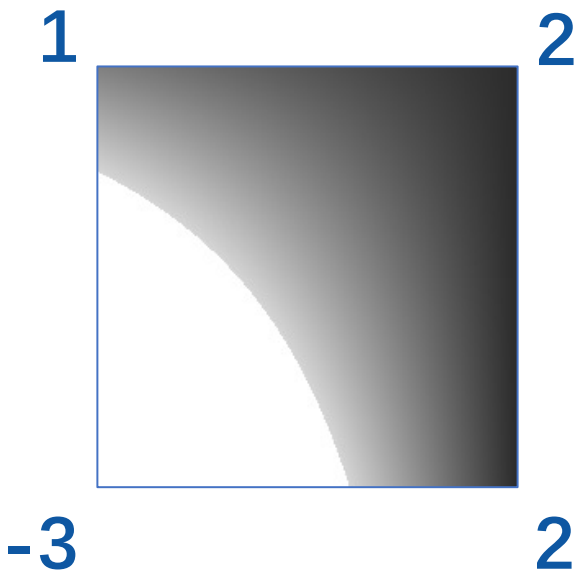
“Plenoptic”：Spherical Harmonics

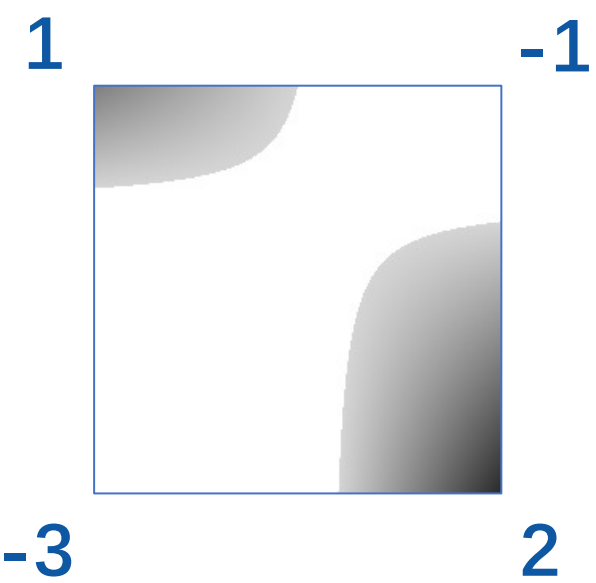


Trilinear Interpolation



Trilinear Interpolation of Density





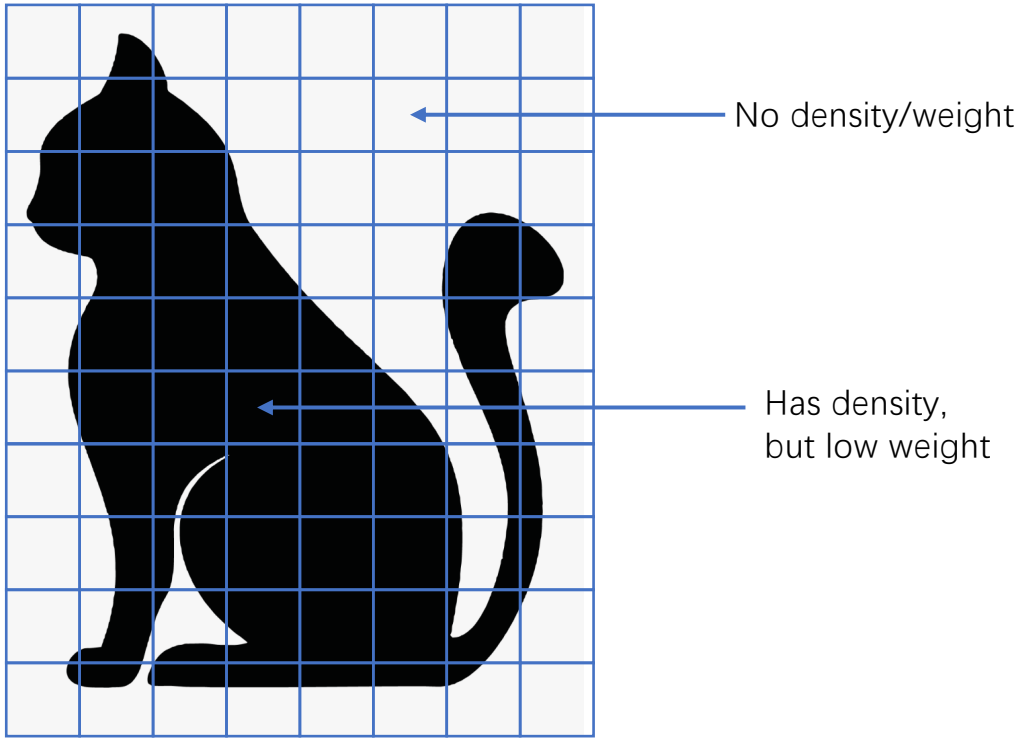
Sparsity: Voxel Pruning

MEGVII 旷视

Weight:

$$T_i(1 - \exp(-\sigma_i \delta_i))$$

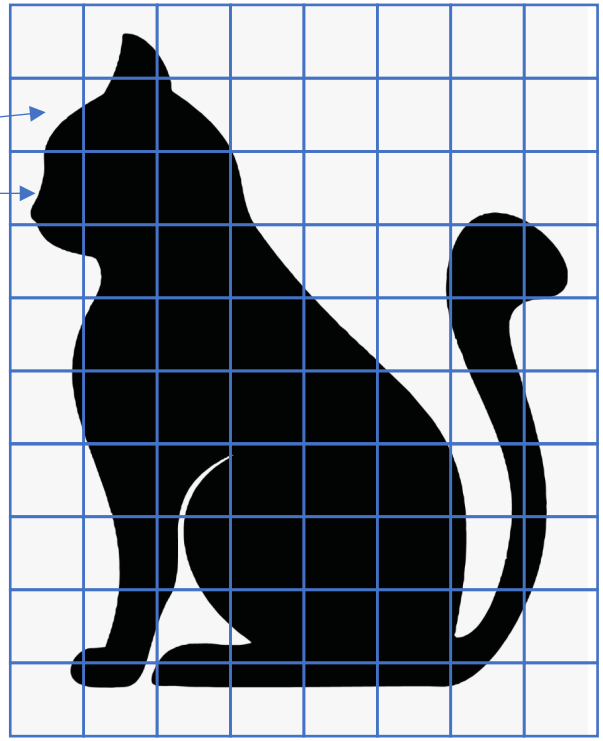
where $T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right)$



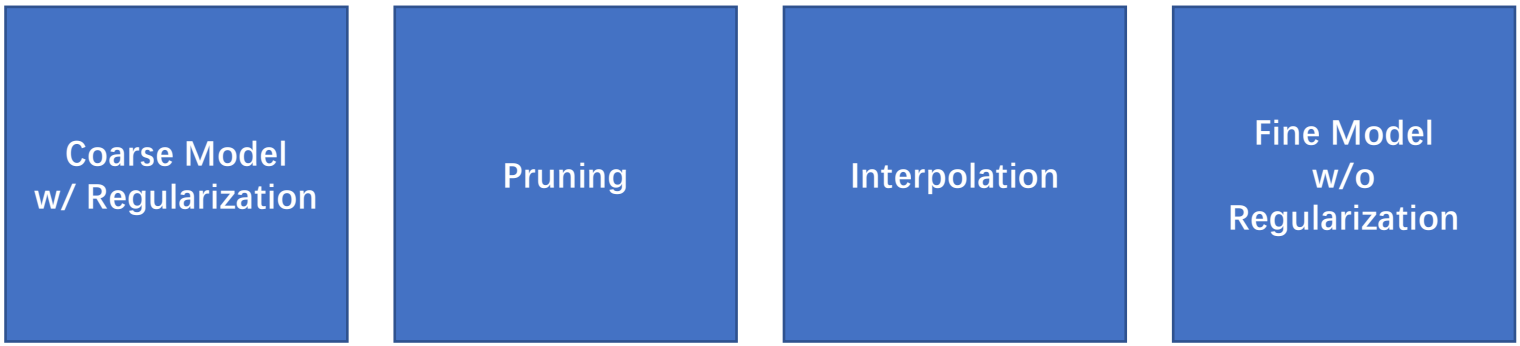
Regularization: Total Variation

Adjacent voxels
should be similar

$$\frac{1}{|\mathcal{V}|}\sum_{\substack{v\in\mathcal{V}\\ d\in[D]}}\sqrt{\Delta_x^2(\mathbf{v},d)+\Delta_y^2(\mathbf{v},d)+\Delta_z^2(\mathbf{v},d)}$$



Coarse to Fine



THANK YOU

MEGVII 旷视