基于NeRF和SDF的三维重建与实践

MEGVII 旷视

主讲人：许伟欣

2022年05月19日

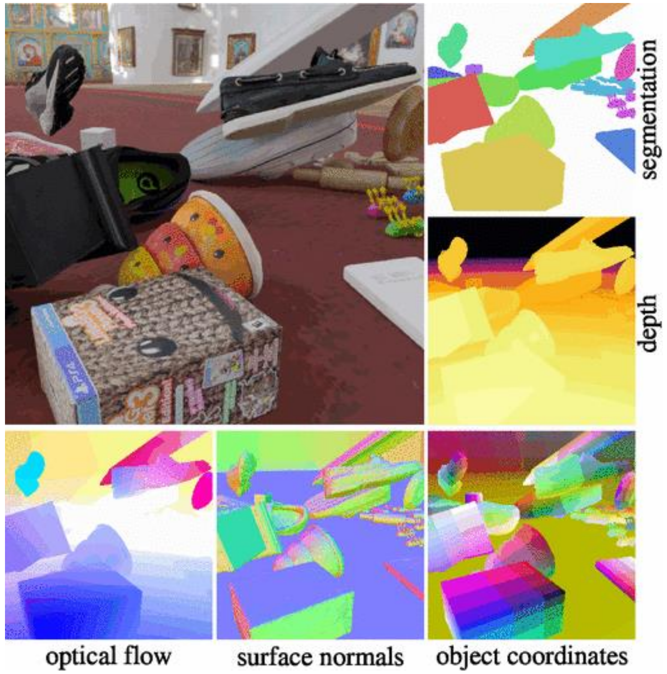
目录
Contents

/ 00

Today's Topic

Today's Topics

- Speeding up & reduce memory footprint, while still maintaining high quality
    - Hybrid Implicit-Explicit method
    - Tensor Decomposition (for the explicit part)
- 3D reconstruction
    - More relevant to NeRF / Volume rendering
    - Build mesh from only 2D RGB inputs
        - Based on SDF
        - Will be compatible to Computer Graphics pipelines
            - *And ready for 3D Printing!*



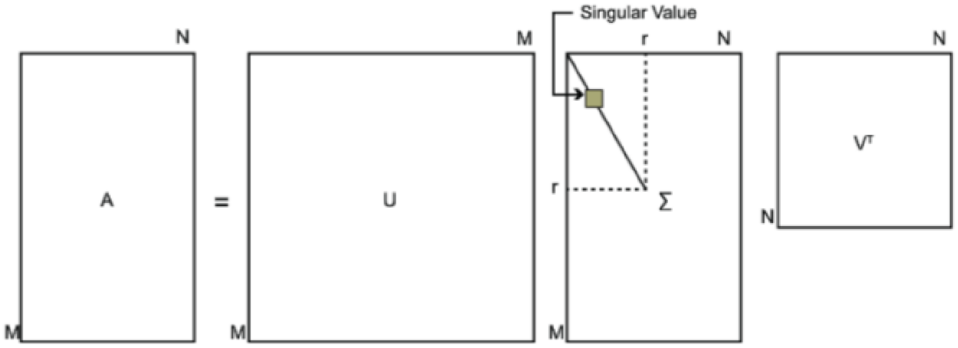
https://github.com/google-research/kubric

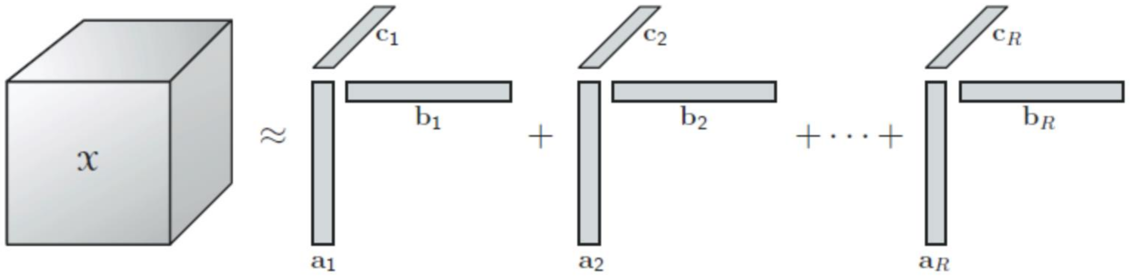
/01

Fundamentals

Fundamentals

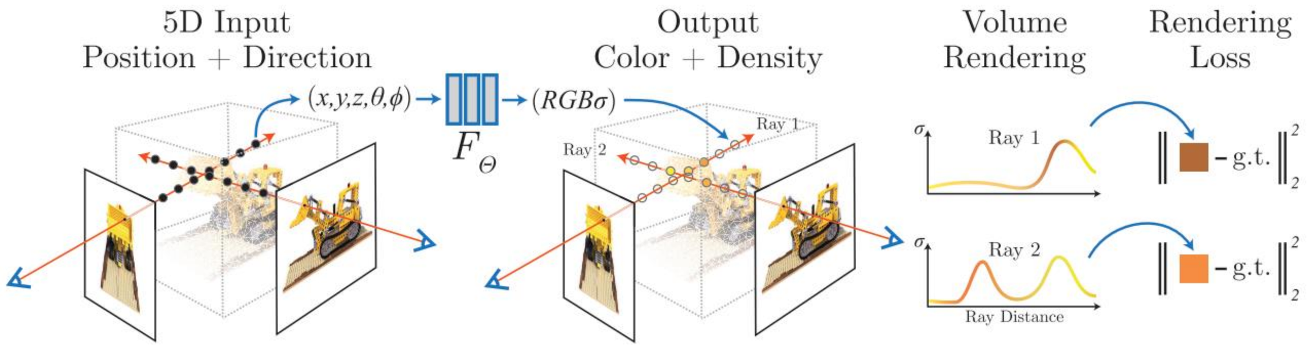
- Volume Rendering (Recap)
- Tensor Decomposition
  - Singular Value Decomposition, CP Decomposition, Tucker Decomposition, Kronecker Product Decomposition





Fundamentals - Volume Rendering

- Neural Radiance Fields (NeRF)
  - High quality, but time consuming, ~8 hours to train per scene



- Volume Rendering
  - render a 2D projection of the 3D data set

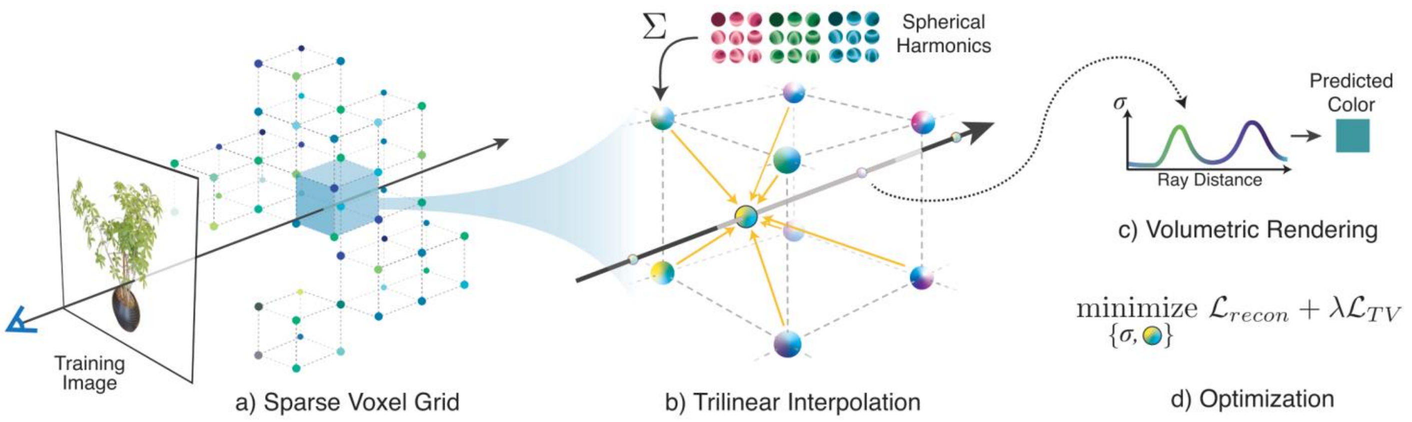
$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t),\mathbf{d})dt\,,\ \text{where } T(t) = \exp\left(-\int_{t_n}^{t}\sigma(\mathbf{r}(s))ds\right)$$

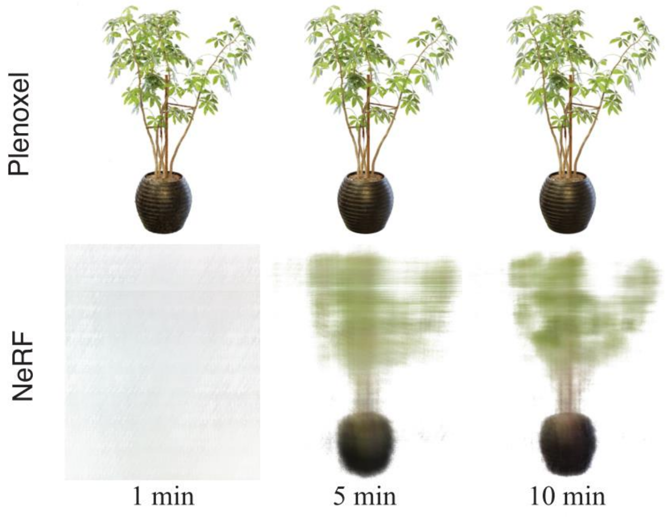
$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1-\exp(-\sigma_i\delta_i))\mathbf{c}_i\,,\ \text{where } T_i = \exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right)$$

Mildenhall B, Srinivasan P P, Tancik M, et al. Nerf: Representing scenes as neural radiance fields for view synthesis[C]//European conference on computer vision. Springer, Cham, 2020: 405-421.

Fundamentals – Explicit Method

- Plenoxels
  - Explicit representation based on Voxels
  - Fast, high quality, but heavy, large memory requirement and large model size, hundreds / thousands of MB, grows with the voxel ***volume***
- Voxel is Tensor, can be ***decomposed!***



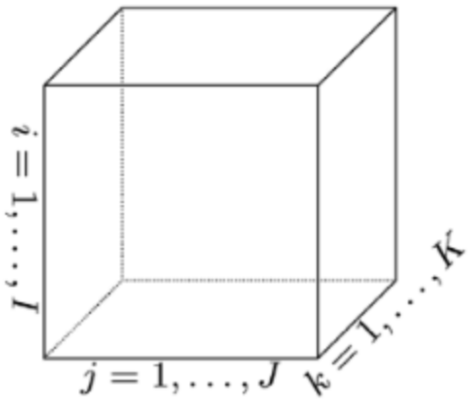


Yu A, Fridovich-Keil S, Tancik M, et al. Plenoxels: Radiance Fields without Neural Networks[J]. arXiv preprint arXiv:2112.05131, 2021.

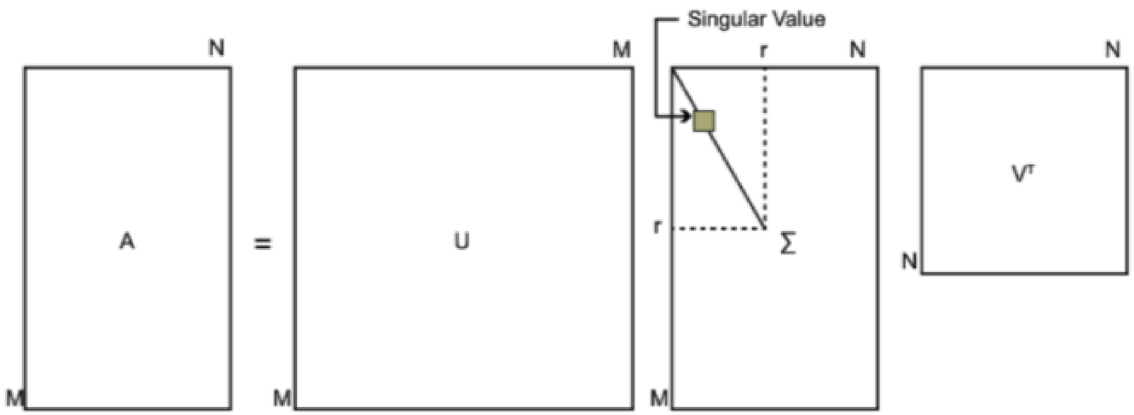
Fundamentals - Tensor Decomposition

MEGVII 旷视

- Tensor
  - Scalar (0th-order), Vector (1st-order), Matrix (2nd-order), Tensor (order >= 3)
- Singular Value Decomposition
  - U and V are semi-unitary matrix
  - The diagonals of Σ are non-negative and are in descending order
  - Rank of M <= min{m, n}



$$\mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^*$$
$$\mathbf{U}^*\mathbf{U} = \mathbf{V}^*\mathbf{V} = \mathbf{I}_r$$

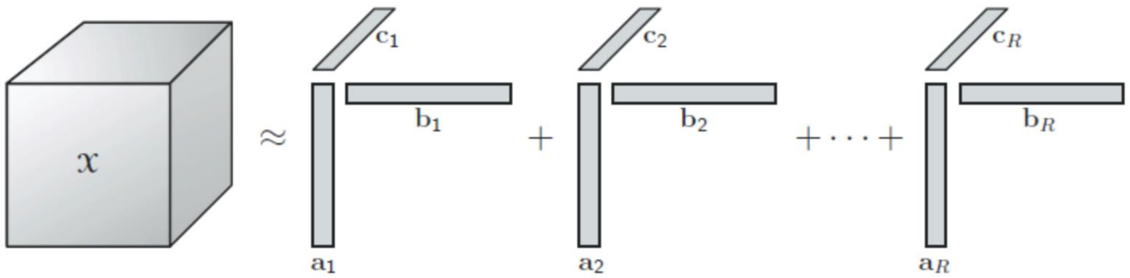


Kolda T G, Bader B W. Tensor decompositions and applications[J]. SIAM review, 2009, 51(3): 455-500.

Fundamentals - Tensor Decomposition

- CP Decomposition
  - The CP decomposition factorizes a tensor into a sum of component rank-one tensors.
  - For a third-order tensor, we have

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$$

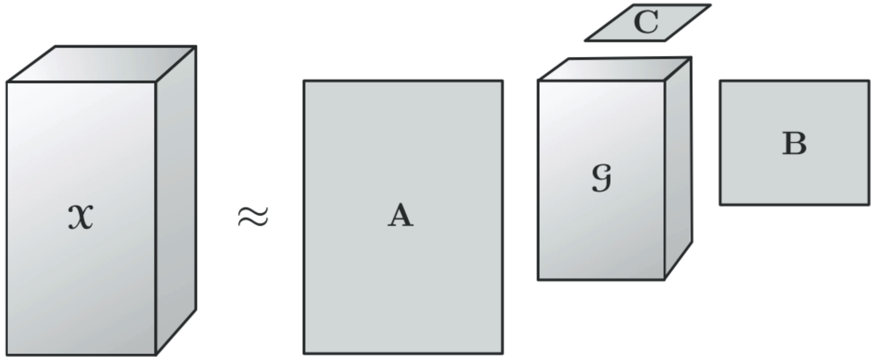


Kolda T G, Bader B W. Tensor decompositions and applications[J]. SIAM review, 2009, 51(3): 455-500.

Fundamentals - Tensor Decomposition

- Tucker Decomposition
  - The Tucker decomposition is a form of higher-order PCA. It decomposes a tensor into a core tensor multiplied (or transformed) by a matrix along each mode.

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr} \, \mathbf{a}_p \circ \mathbf{b}_q \circ \mathbf{c}_r = [\![\mathcal{G} \,; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!].$$



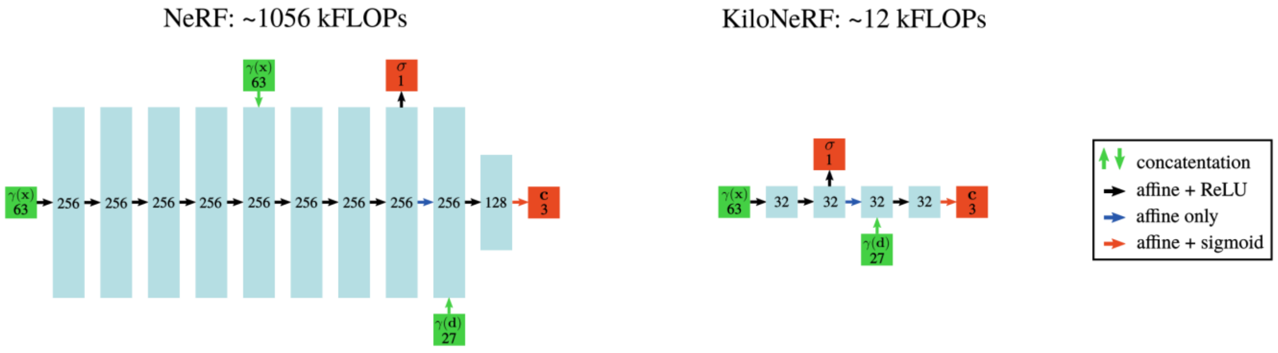
Kolda T G, Bader B W. Tensor decompositions and applications[J]. SIAM review, 2009, 51(3): 455-500.

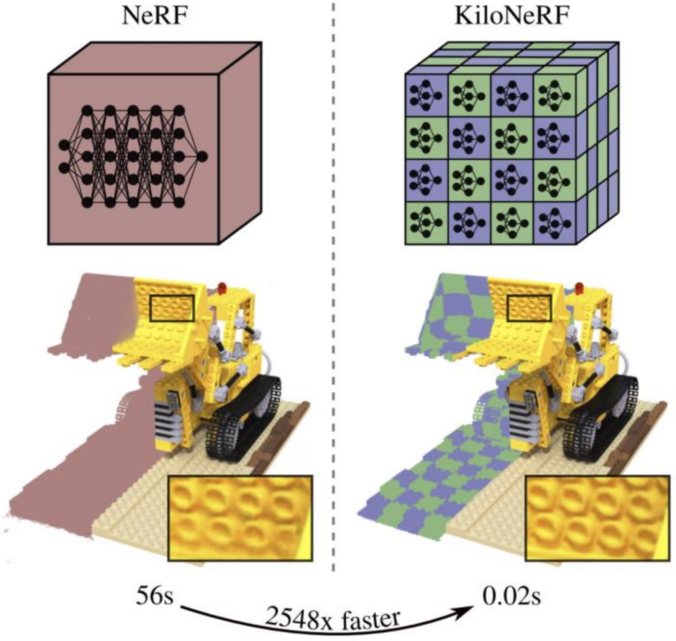
/02

Fast Radiance Fields

Fast Radiance Fields – KiloNeRF

- Real-time rendering
  - Decompose the MLP
  - Each tiny MLP represents part of the scene
    - 9 hidden layers of 128 channels ==> 4 hidden layers of 32 channels
  - Distillation
    - Train the teacher + distillation + finetune ~ up to 2 days

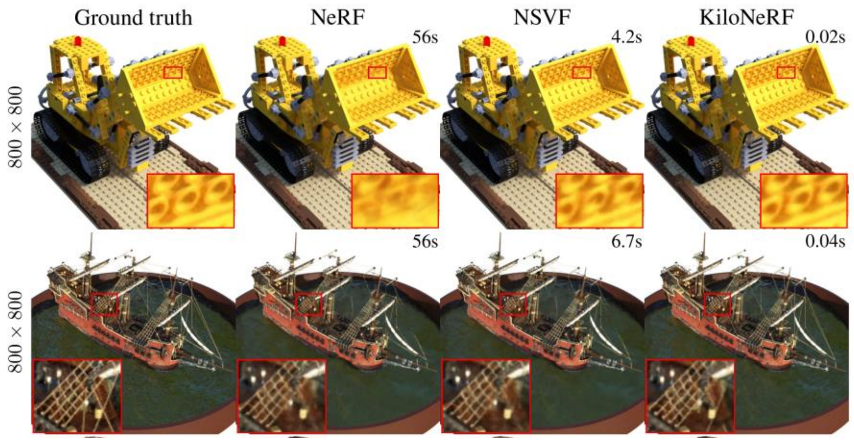


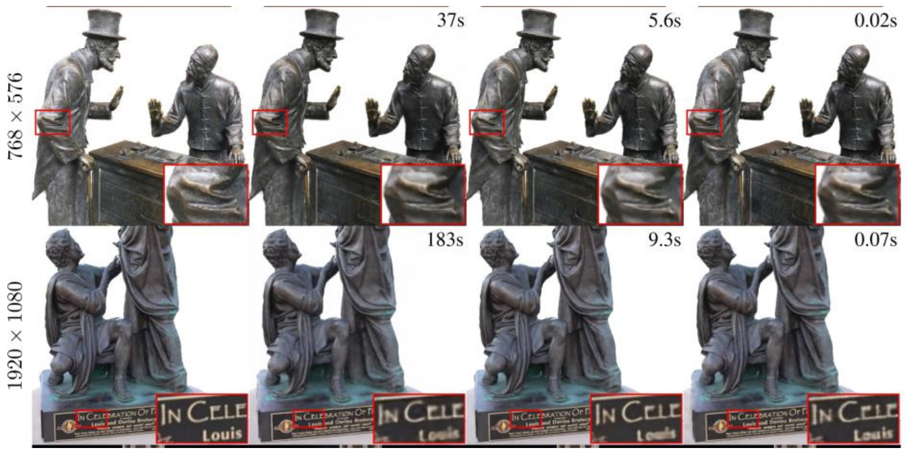


Reiser C, Peng S, Liao Y, et al. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 14335-14345.

Fast Radiance Fields – KiloNeRF

- Extras
  - Empty space skipping
    - Occupancy grid
  - Early ray termination
    - Transmittance > threshold
  - Cuda, simultaneously query thousands of tiny MLPs

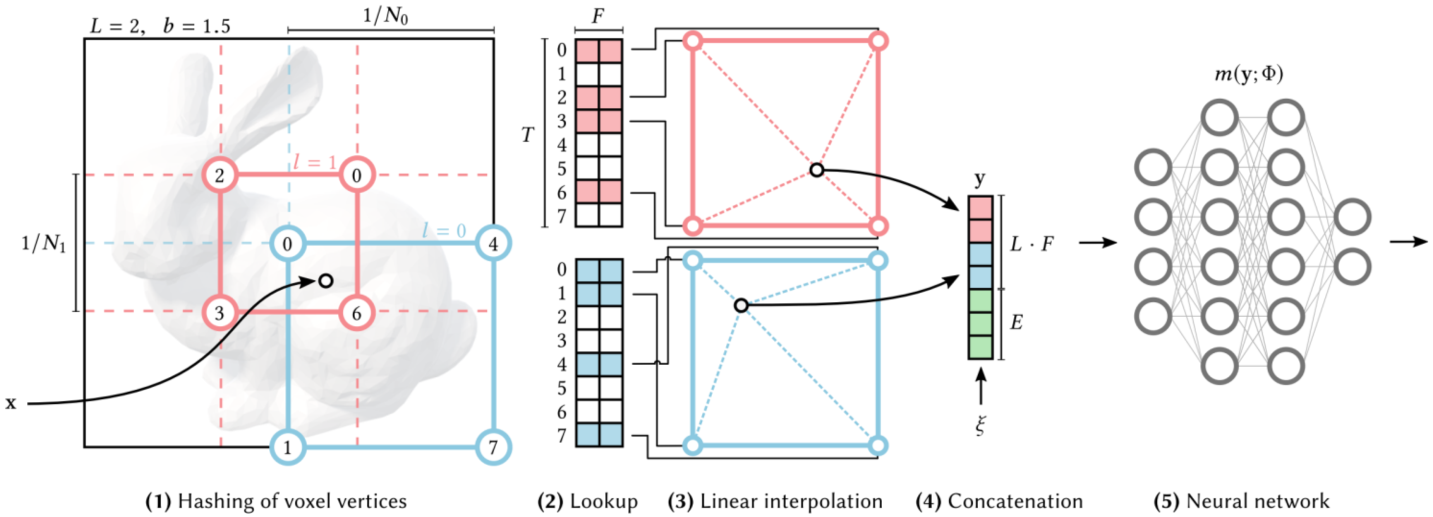




Reiser C, Peng S, Liao Y, et al. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 14335-14345.

Fast Radiance Fields – Instant NGP

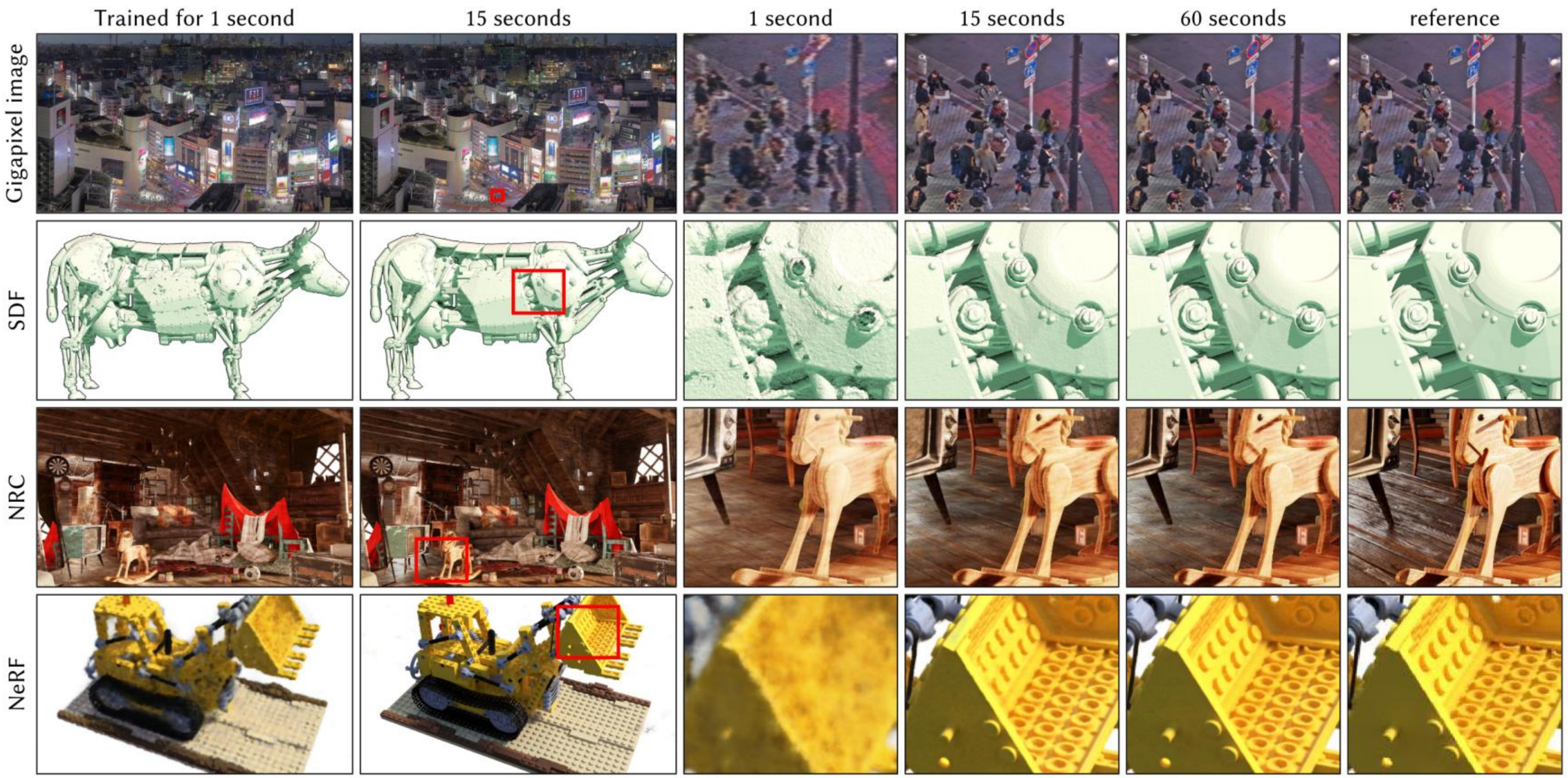
- Multi-resolution Hash Encoding: Voxel ==> Feature
- Lookup Table
  - Coarse level, num_vertices < T ==> 1:1 mapping
  - Fine level, num_vertices > T ==> hash table ==> collision



Müller T, Evans A, Schied C, et al. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding[J]. arXiv preprint arXiv:2201.05989, 2022.

Fast Radiance Fields – Instant NGP

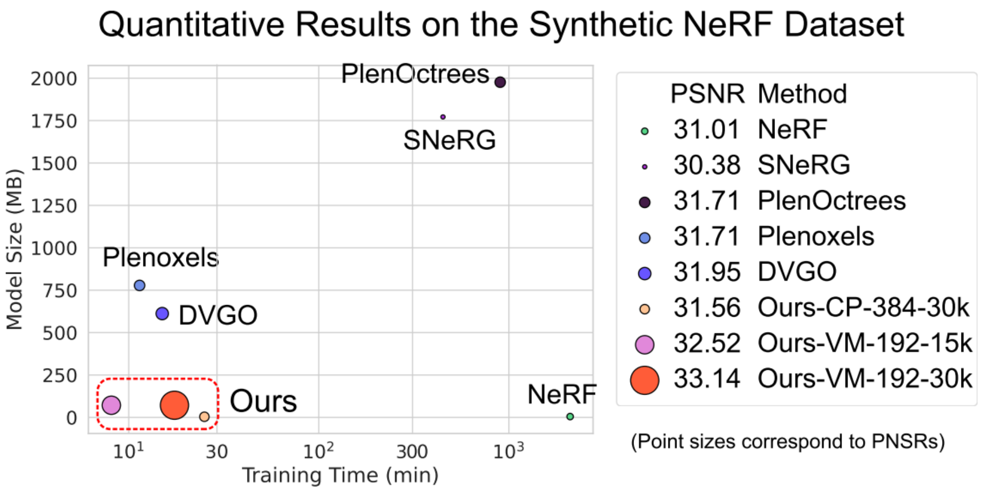
- Collision: only implicitly resolved
  - With multiple resolution, more important samples dominate the collision average
  - Followed by a lightweight MLP as a neural decoder
- Implementing in CUDA boosts performance



Müller T, Evans A, Schied C, et al. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding[J]. arXiv preprint arXiv:2201.05989, 2022.

Fast Radiance Fields – TensoRF

- Basic ideas
  - model the radiance field of a scene as a 4D tensor, which represents a 3D voxel grid with per-voxel multi-channel features
  - factorize the 3D scene tensor (without the feature dimension) into multiple compact low-rank tensor components
  - The factorization is followed by a decoding function (MLP/SH)



Chen A, Xu Z, Geiger A, et al. TensoRF: Tensorial Radiance Fields[J]. arXiv preprint arXiv:2203.09517, 2022.

Fast Radiance Fields – TensoRF

- Two kinds of Decomposition
  - CP Decomposition
  - Vector-Matrix (VM) Decomposition

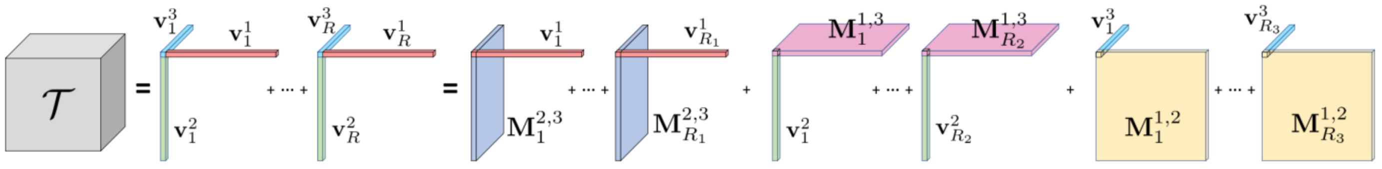
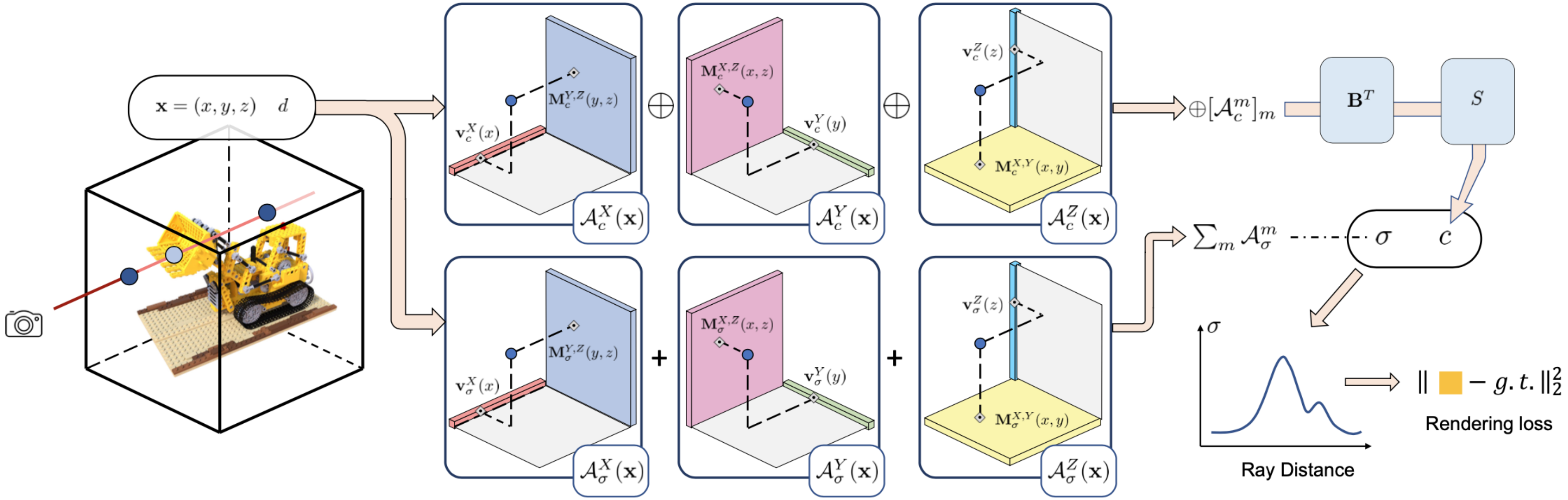


Fig. 2: Tensor factorization. Left: CP decomposition (Eqn. 1), which factorizes a tensor as a sum of vector outer products. Right: our vector-matrix decomposition (Eqn. 3), which factorizes a tensor as a sum of vector-matrix outer products.



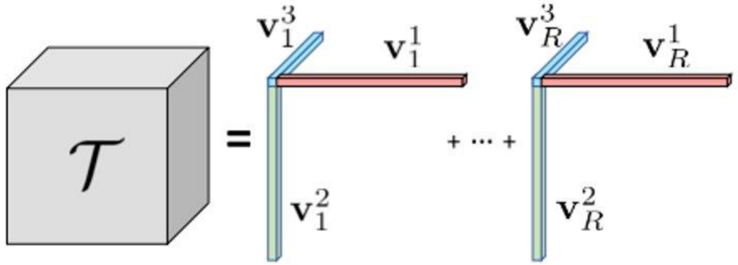
Chen A, Xu Z, Geiger A, et al. TensoRF: Tensorial Radiance Fields[J]. arXiv preprint arXiv:2203.09517, 2022.

Fast Radiance Fields – TensoRF

- CP Decomposition
  - Memory Complexity: $\mathcal{O}(n^3)$ to $\mathcal{O}(n)$
  - Where $\mathbf{v}_r^1 \circ \mathbf{v}_r^2 \circ \mathbf{v}_r^3$ corresponds to a rank-one tensor component

$$\mathcal{T} = \sum_{r=1}^{R} \mathbf{v}_r^1 \circ \mathbf{v}_r^2 \circ \mathbf{v}_r^3$$

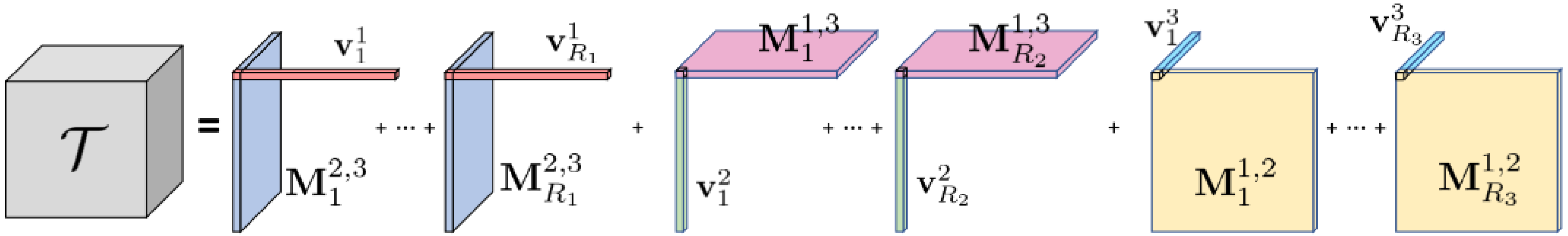
$$\mathcal{T}_{ijk} = \sum_{r=1}^{R} \mathbf{v}_{r,i}^1 \mathbf{v}_{r,j}^2 \mathbf{v}_{r,k}^3$$



Chen A, Xu Z, Geiger A, et al. TensoRF: Tensorial Radiance Fields[J]. arXiv preprint arXiv:2203.09517, 2022.

Fast Radiance Fields – TensoRF

- Vector-Matrix (VM) Decomposition
  - Memory Complexity: $\mathcal{O}(N^3)$ to $\mathcal{O}(N^2)$
  - For each component, we relax its two mode ranks to be arbitrarily large, while restricting the third mode to be rank-one
  - R1, R2, R3 can be set differently and should be chosen depending on the complexity of each mode



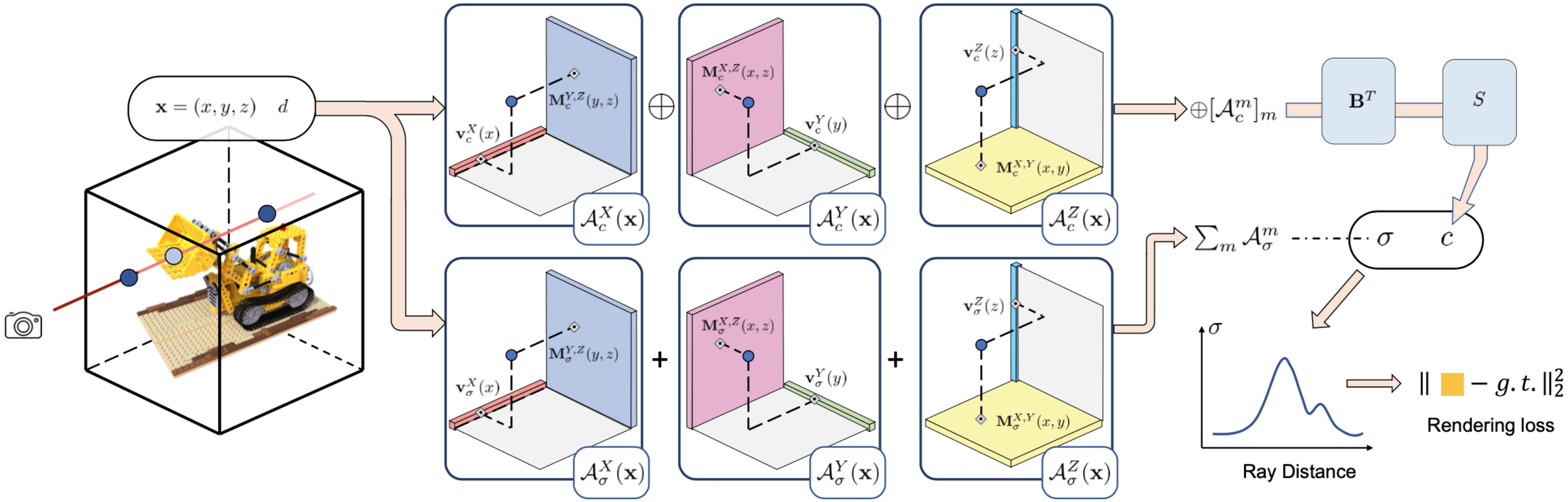
$$\mathcal{T} = \sum_{r=1}^{R_1} \mathbf{v}_r^1 \circ \mathbf{M}_r^{2,3} + \sum_{r=1}^{R_2} \mathbf{v}_r^2 \circ \mathbf{M}_r^{1,3} + \sum_{r=1}^{R_3} \mathbf{v}_r^3 \circ \mathbf{M}_r^{1,2}$$

Chen A, Xu Z, Geiger A, et al. TensoRF: Tensorial Radiance Fields[J]. arXiv preprint arXiv:2203.09517, 2022.

Fast Radiance Fields – TensoRF

- Scene Modeling
  - A regular 3D grid G with per-voxel multi-channel features, followed by a MLP (optional)
  - Split G into a geometry grid $G\sigma$ and an appearance grid Gc, separately modelling the volume density σ a nd view-dependent color c
  - Model $G\sigma$ and Gc as factorized tensors

$$\sigma, c = \mathcal{G}_\sigma(\mathbf{x}), S(\mathcal{G}_c(\mathbf{x}), d)$$



Chen A, Xu Z, Geiger A, et al. TensoRF: Tensorial Radiance Fields[J]. arXiv preprint arXiv:2203.09517, 2022.

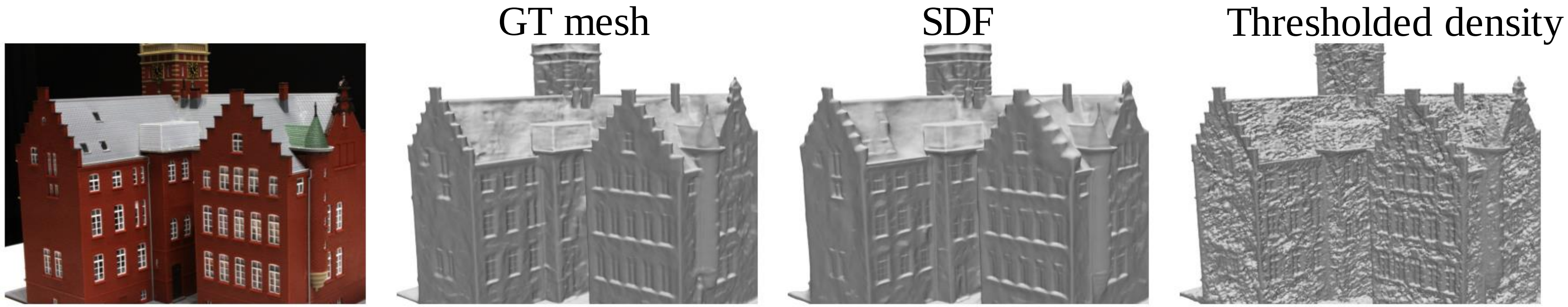
/ 03

Multi-view Reconstruction with SDF

Previous Review – Mesh

- A polygon mesh is a collection of vertices, edges and faces that defines the shape of a polyhedral object.
  - Widely used in Computer Graphics, can represent complicated surface with flexibility
- NeRF to Mesh
  - Density threshold is hard to choose
  - Noisy and may have holes
- Signed Distance Function (SDF) is a better choice
  - Continuous
  - With SDF, can use Marching Cubes to get the mesh

threshold σ	1	5	10	**50**	100	500
scan65	2.29	1.53	1.26	1.27	1.80	3.15
scan105	3.46	2.27	1.85	1.07	1.32	5.99
scan114	2.88	1.74	1.37	1.06	1.21	2.86

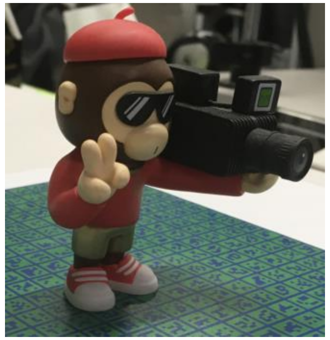


W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In SIGGRAPH, volume 21, pages 163–169. ACM, 1987.

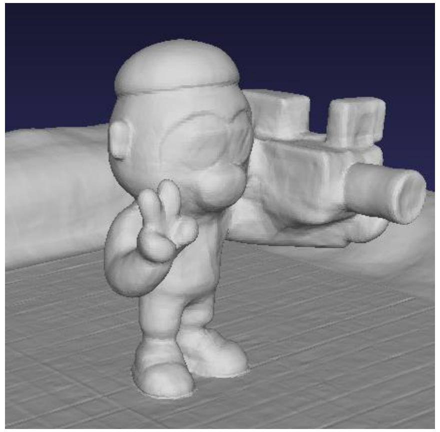
Multi-view Reconstruction by Volume Rendering

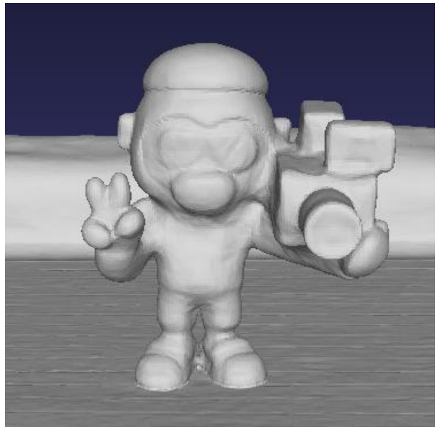
MEGVII 旷视

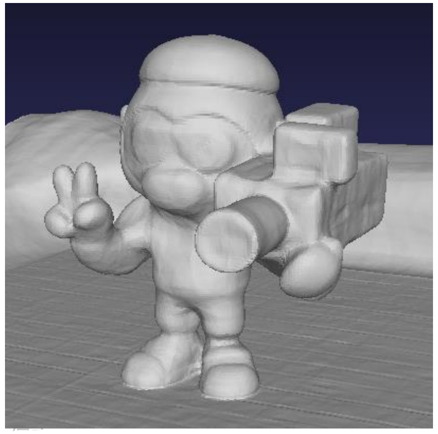
- Task Definition
  - Inputs: 2D RGB images



  - Outputs:
    - Mesh extracted from the 3D object representation (SDF)
    - Synthesized novel view







Fundamentals - Signed Distance Function

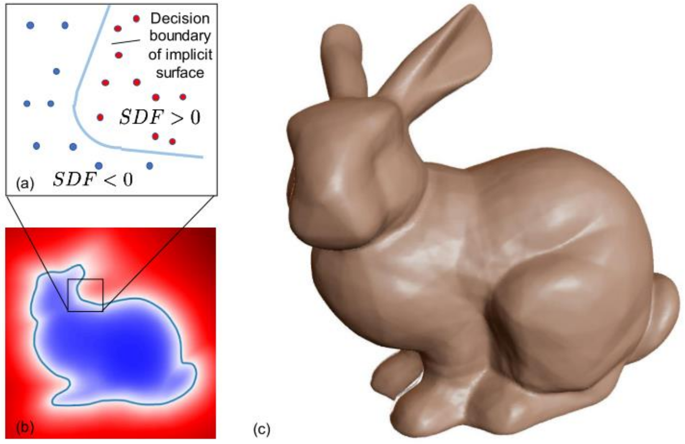
- Signed Distance Function
  - For every point, give distance to the surface
  - Zero level set

If Ω is a subset of a metric space, X, with metric, d, then the *signed distance function*, f, is defined by

$$f(x) = \begin{cases} d(x, \partial\Omega) & \text{if } x \in \Omega \\ -d(x, \partial\Omega) & \text{if } x \in \Omega^c \end{cases}$$

where $\partial\Omega$ denotes the boundary of Ω. For any $x \in X$,

$$d(x, \partial\Omega) := \inf_{y \in \partial\Omega} d(x, y)$$



- Applications
  - Ray marching, Scene representation, Fonts rendering

Park J J, Florence P, Straub J, et al. Deepsdf: Learning continuous signed distance functions for shape representation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 165-174.

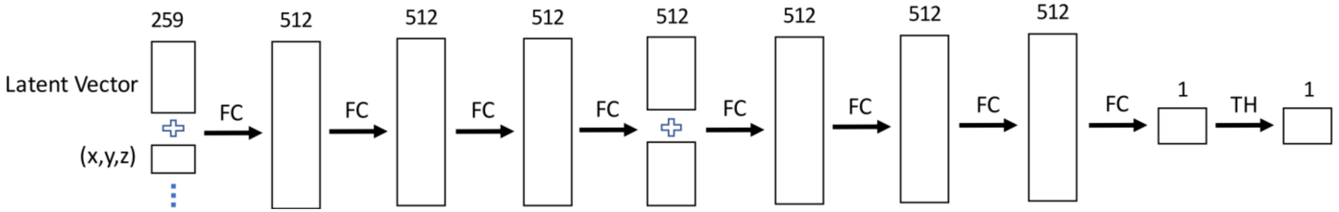
Multi-view Reconstruction by Volume Rendering

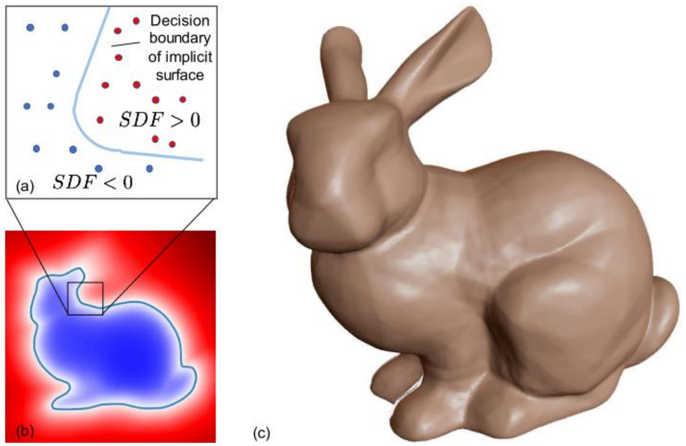
- DeepSDF
  - Model the shapes as the zero iso-surface decision boundaries of feed-forward networks trained to represent SDFs

$$SDF(\boldsymbol{x}) = s : \boldsymbol{x} \in \mathbb{R}^3,\ s \in \mathbb{R}$$

$$SDF(\cdot) = 0$$

$$f_\theta(\boldsymbol{x}) \approx SDF(\boldsymbol{x}),\ \forall \boldsymbol{x} \in \Omega$$





Park J J, Florence P, Straub J, et al. Deepsdf: Learning continuous signed distance functions for shape representation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 165-174.

Multi-view Reconstruction by Volume Rendering

- How?
  - 2D (inputs RGB images) ==> 3D (represented scene)
- What does NeRF do?
  - Density Field + Color Field ==> Rendered 2D images $\overset{\text{loss}}{\Longleftrightarrow}$ Ground Truth 2D images

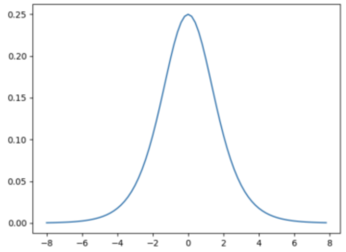
$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt\,,\ \text{where } T(t) = \exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$$

- What can we do?
  - Signed Distance Field + Color Field ==> Rendered 2D images $\overset{\text{loss}}{\Longleftrightarrow}$ Ground Truth 2D images
  - Optional: there exists other choices besides SDF, e.g. Occupancy Field, but will not be introduced today
- Problems
  - *How to transform the SDF into the Density Field?*

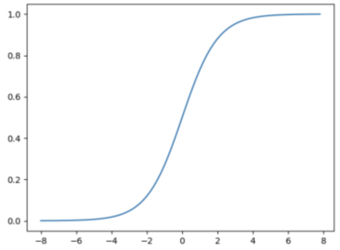
Density Modeling - NeuS

- *Density Modeling*: Signed Distance Field ==> Density Field
  - Volume Rendering $C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t),\mathbf{d})dt\,,$ where $T(t) = \exp\left(-\int_{t_n}^{t}\sigma(\mathbf{r}(s))ds\right)$
  - SDF for scene representation $\mathcal{S} = \left\{\mathbf{x} \in \mathbb{R}^3 | f(\mathbf{x}) = 0\right\}$
- NeuS
  - S-density: any unimodal density distribution centered at 0
    - For convenience, choose logistic distribution $\phi_s(f(\mathbf{x}))$
  - Weight function w(t)
    - Unbiased
    - Occlusion-aware

$$C(\mathbf{o},\mathbf{v}) = \int_0^{+\infty} w(t)c(\mathbf{p}(t),\mathbf{v})\mathrm{d}t$$



Logistic Distribution



Sigmoid

Wang P, Liu L, Liu Y, et al. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction[J]. arXiv preprint arXiv:2106.10689, 2021.

Density Modeling - NeuS

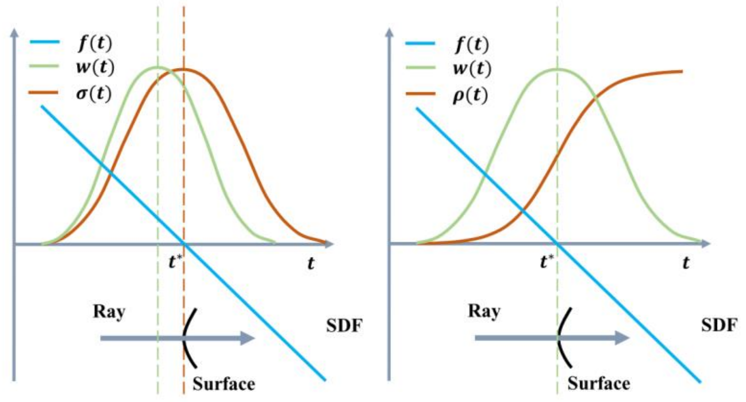
- Density Modeling
  - Volume rendering $C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t),\mathbf{d})dt\,,\ \text{where } T(t) = \exp\left(-\int_{t_n}^{t}\sigma(\mathbf{r}(s))ds\right)$
  - Weight function w(t)
    - Unbiased
    - Occlusion-aware

$$C(\mathbf{o},\mathbf{v}) = \int_0^{+\infty} w(t)c(\mathbf{p}(t),\mathbf{v})\mathrm{d}t$$

  - Naïve solution, straight forward, let $w(t) = T(t)\sigma(t)$ $\sigma(t) = \phi_s(f(\mathbf{p}(t)))$
    - Occlusion-aware, but biased
  - NeuS solution

$$w(t) = T(t)\rho(t),\quad \text{where } T(t) = \exp(-\int_0^t \rho(u)\mathrm{d}u)$$

$$\rho(t) = \max\left(\frac{-\frac{\mathrm{d}\Phi_s}{\mathrm{d}t}(f(\mathbf{p}(t)))}{\Phi_s(f(\mathbf{p}(t)))}, 0\right)$$



(a) Naive Solution (b) Our Solution

Wang P, Liu L, Liu Y, et al. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction[J]. arXiv preprint arXiv:2106.10689, 2021.

Density Modeling - NeuS

MEGVII 旷视

- Discretization
  - NeRF

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

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

  - NeuS

$$\hat{C} = \sum_{i=1}^{n} T_i\alpha_i c_i$$

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

$$\alpha_i = \max\left(\frac{\Phi_s(f(\mathbf{p}(t_i))) - \Phi_s(f(\mathbf{p}(t_{i+1})))}{\Phi_s(f(\mathbf{p}(t_i)))}, 0\right)$$

Wang P, Liu L, Liu Y, et al. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction[J]. arXiv preprint arXiv:2106.10689, 2021.

Density Modeling - NeuS

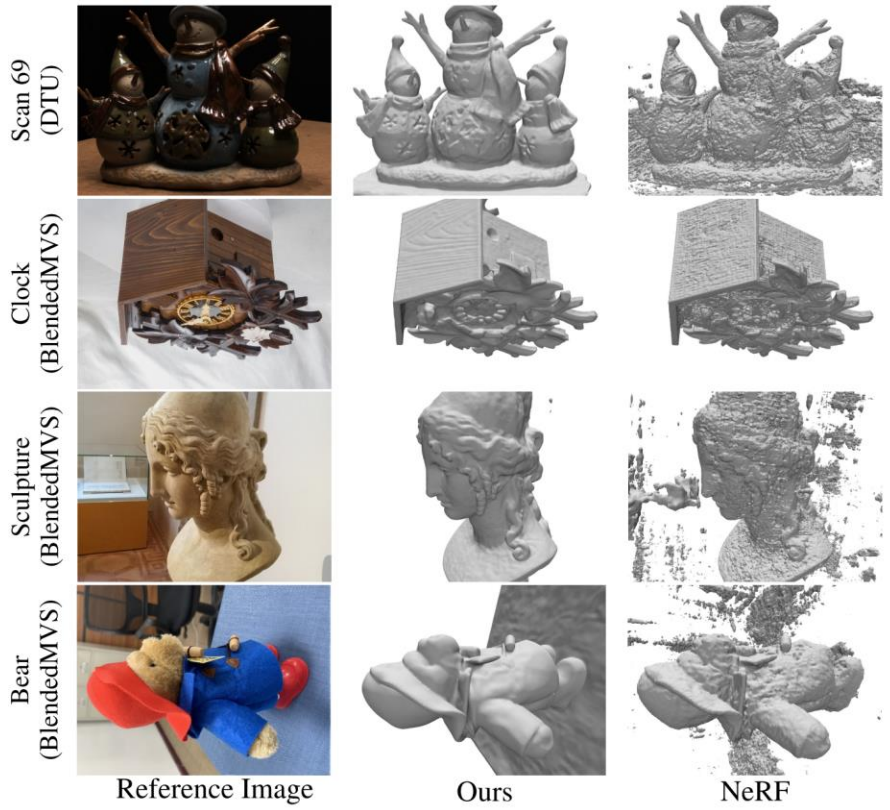
- Training
  - Rendering loss + Eikonal Loss (+ Mask Loss)

$$\mathcal{L} = \mathcal{L}_{color} + \lambda \mathcal{L}_{reg} + \beta \mathcal{L}_{mask}$$

$$\mathcal{L}_{color} = \frac{1}{m} \sum_k \mathcal{R}(\hat{C}_k, C_k)$$

$$\mathcal{L}_{reg} = \frac{1}{nm} \sum_{k,i} (|\nabla f(\hat{\mathbf{p}}_{k,i})| - 1)^2$$

$$\mathcal{L}_{mask} = \mathrm{BCE}(M_k, \hat{O}_k)$$



Wang P, Liu L, Liu Y, et al. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction[J]. arXiv preprint arXiv:2106.10689, 2021.
Gropp A, Yariv L, Haim N, et al. Implicit geometric regularization for learning shapes[J]. arXiv preprint arXiv:2002.10099, 2020.

Density Modeling - Extras

- VolSDF
  - Cumulative Distribution Function (CDF) of the Laplace distribution with zero mean and β scale

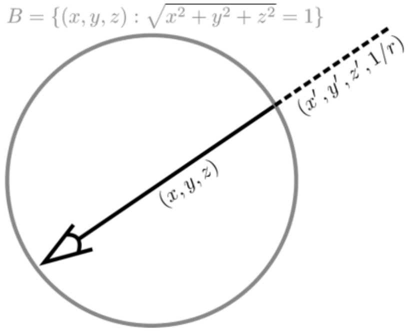
$$\sigma(\boldsymbol{x}) = \alpha \Psi_\beta \left(-d_\Omega(\boldsymbol{x})\right)$$

$$\Psi_\beta(s) = \begin{cases} \frac{1}{2}\exp\left(\frac{s}{\beta}\right) & \text{if } s \leq 0 \\ 1 - \frac{1}{2}\exp\left(-\frac{s}{\beta}\right) & \text{if } s > 0 \end{cases}$$

- Unisurf
  - Occupancy Field instead of SDF

$$o_\theta(\mathbf{x}) : \mathbb{R}^3 \to [0, 1]$$

- Background
  - NeRF++
  - Inverted sphere parameterization



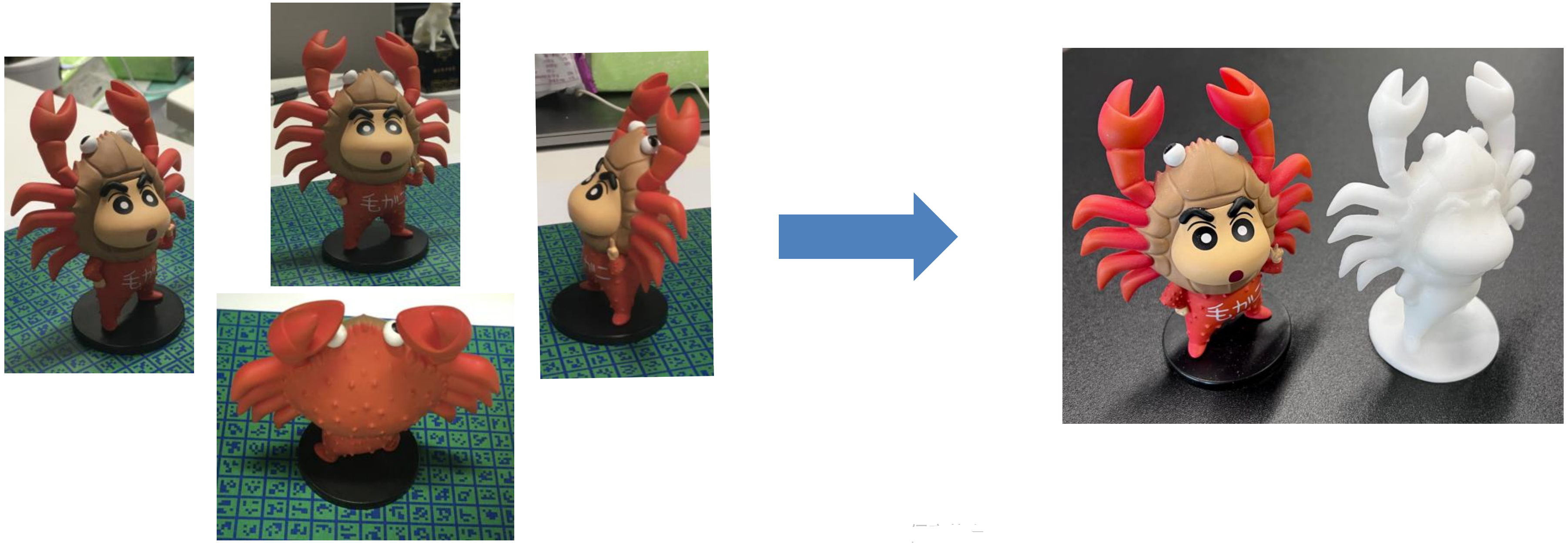
Yariv L, Gu J, Kasten Y, et al. Volume rendering of neural implicit surfaces[J]. Advances in Neural Information Processing Systems, 2021, 34.
Oechsle M, Peng S, Geiger A. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 5589-5599.
Zhang K, Riegler G, Snavely N, et al. Nerf++: Analyzing and improving neural radiance fields[J]. arXiv preprint arXiv:2010.07492, 2020.

/ 04

Practice

Practice

- Find an object that you are interested in (For convenience, not taller than 20 cm)
- Take ~40 pictures around the object
- Preprocess the pictures
- Run code to get your reconstructed object, 3D print it (optional, we will choose some of your works and print for you)



用人工智能造福大众

MEGVII 旷视