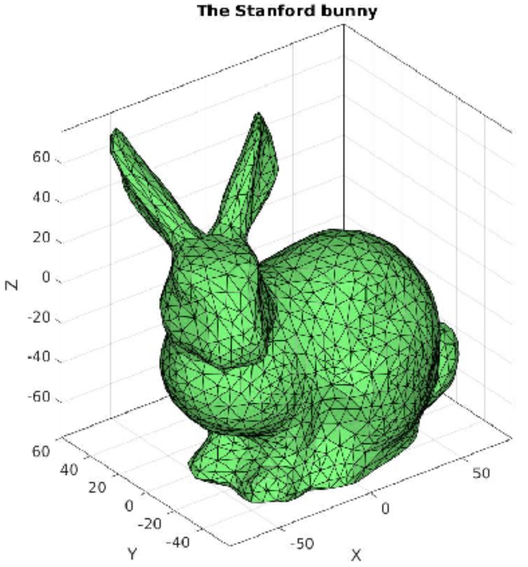
VI: Implicit Representations for 3D

3D CV

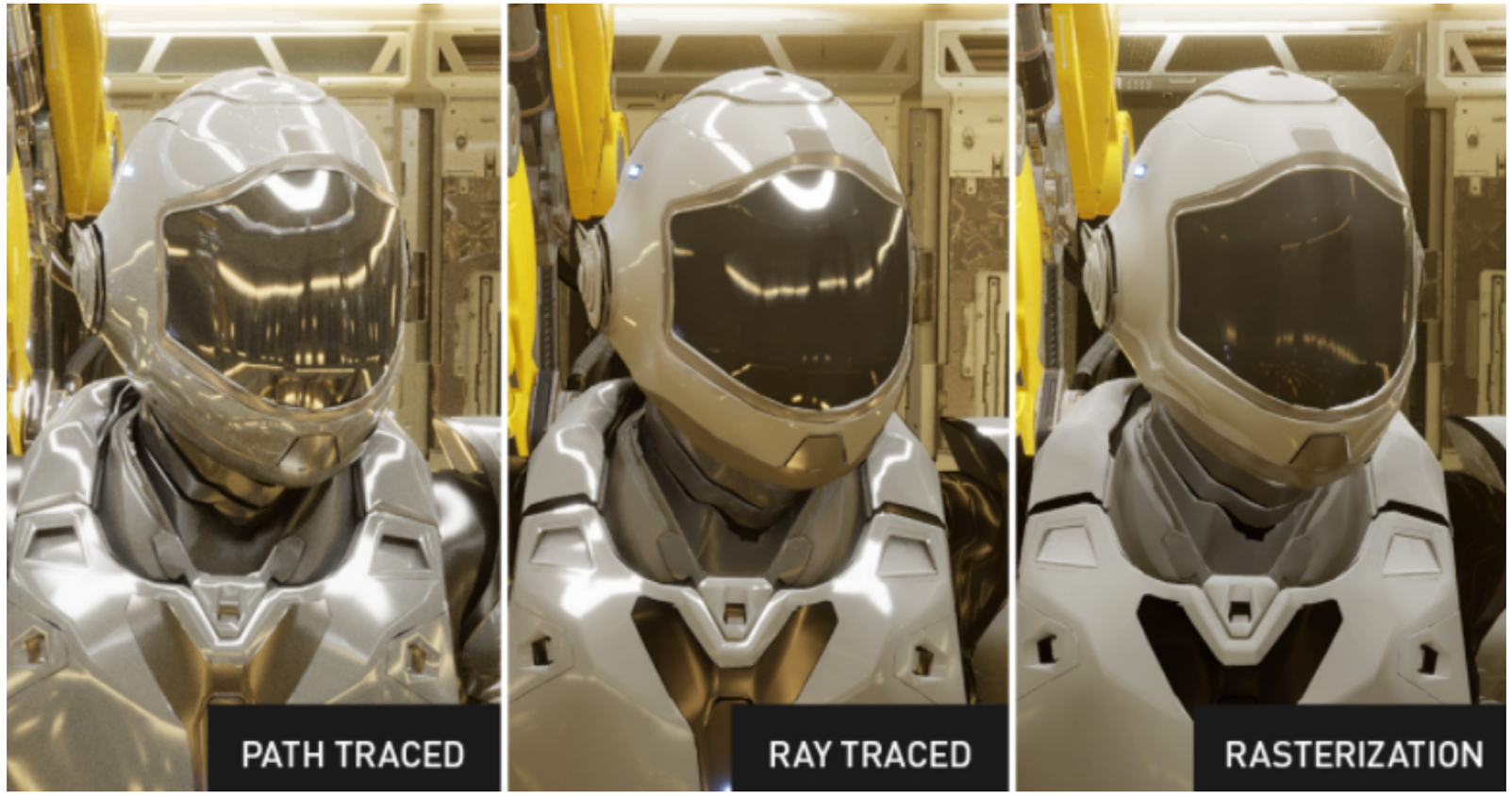
Kirill Struminsky

In the Previous Episode

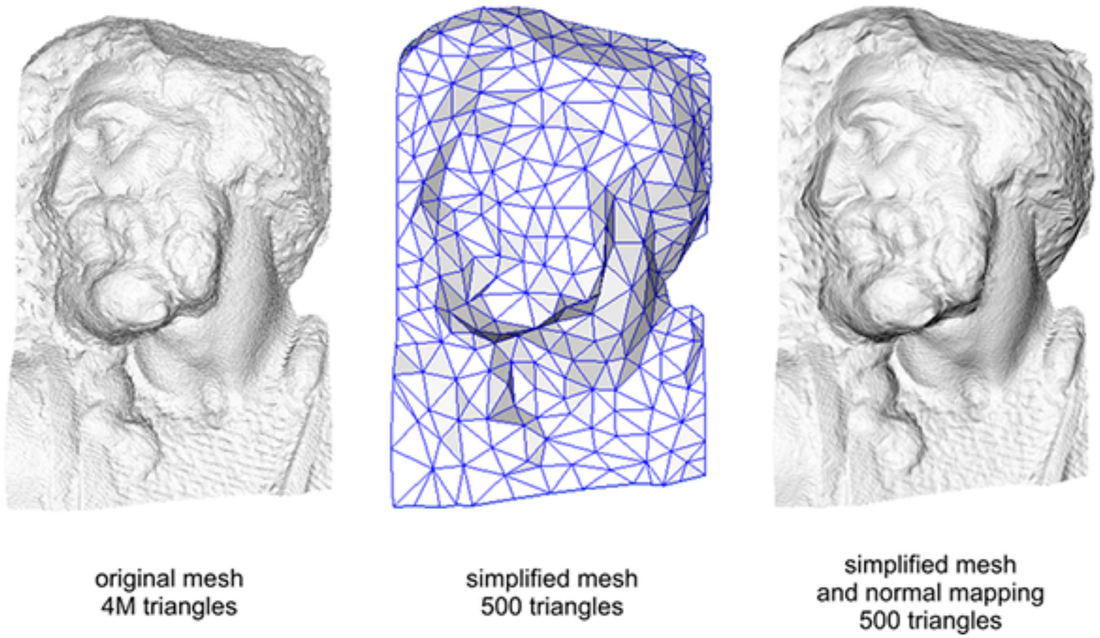
- Triangular meshes for surface representations
- Rendering algorithms
  - Path tracing
  - Rasterization
- Volumetric representations for 3D reconstruction and beyond

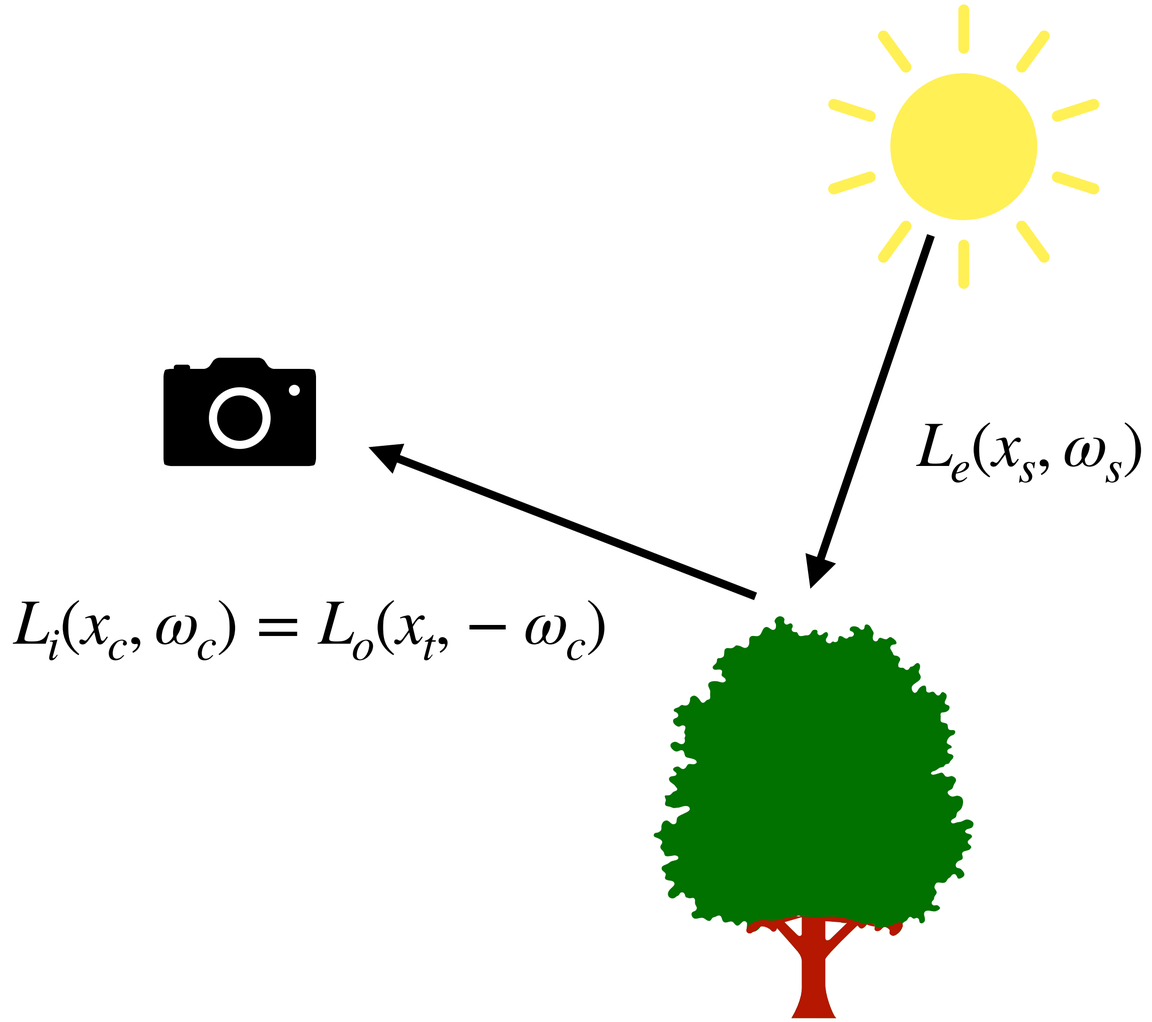


Path Tracing versus Rasterisation



Radiance Functions

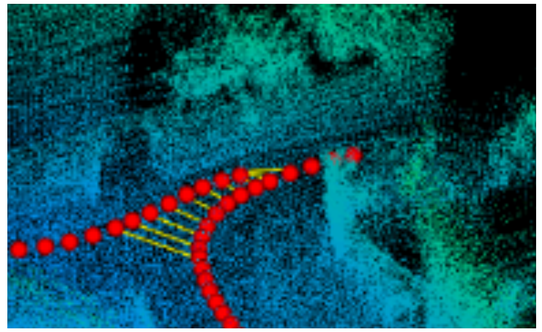


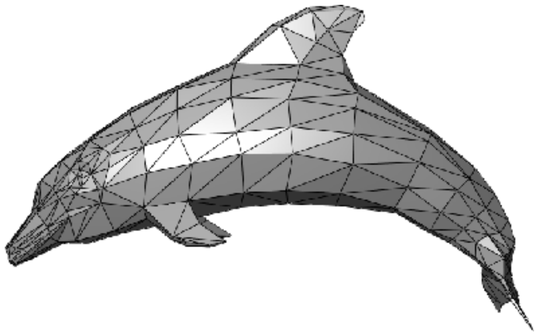


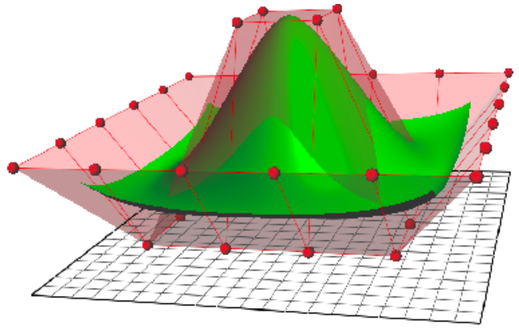
$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_\Omega f(x, \omega_i, \omega_o) L_i(x, \omega_i)(\omega_i \cdot \mathbf{n}) \mathrm{d}\omega_i$$

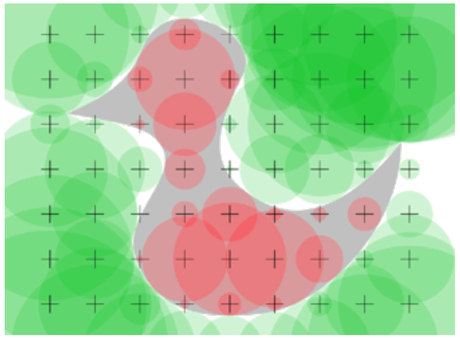
Common Representations

- Explicit
  - Point clouds
  - Polygonal meshes
  - Parametric $x = g(t)$
- Implicit
  - Isosurfaces $x : f(x) = 0$





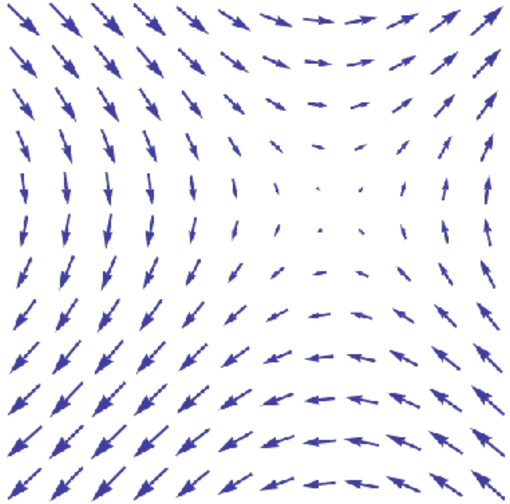


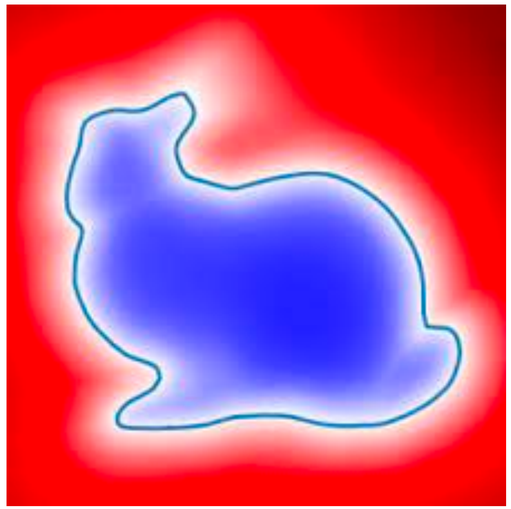


Fields for Volumetric Representations

Vector Fields

- Used in physics and beyond
- Relevant examples
  - Light absorption coefficients
  - Signed Distance Fields
  - Occupancy Grids
  - Images





https://en.wikipedia.org/wiki/X-ray
https://arxiv.org/abs/1901.05103
https://en.wikipedia.org/wiki/Lenna

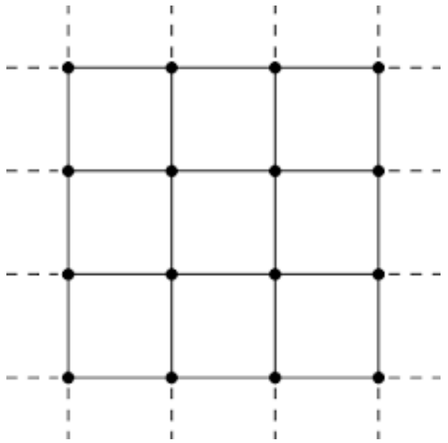
Volumetric Representations

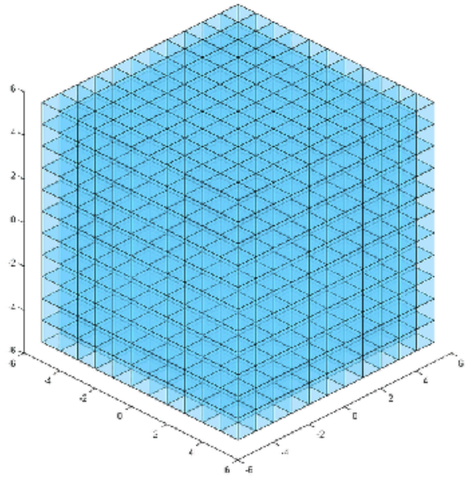
- Surface is a common abstraction in computer graphics
  - Convenience, hardware optimization
  - Sometimes fail to represent real world phenomena
- As an alternative, we will represent scenes with “density” functions
  - Is there something in this point of space?
  - Does light pass through this point in space?
- Scalar **field:** each spatial point stores a scalar value

https://www.reddit.com/r/MicrosoftFlightSim/comments/ymz0ex/is_this_really_what_photogrammetry_looks_like_for/

Grid Representations

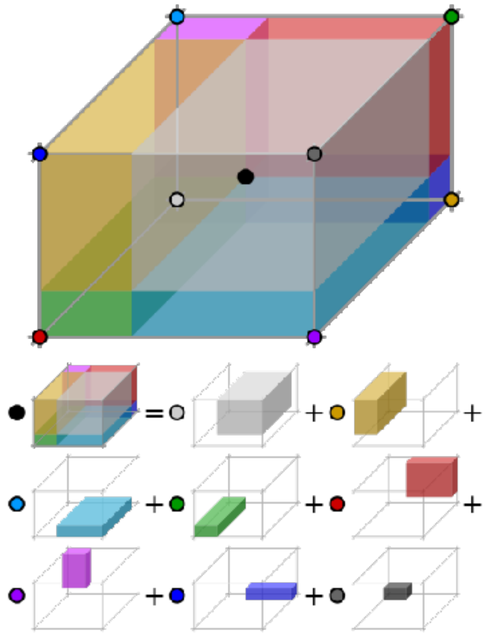
Pixels, Voxels, etc.





- Consider a field $f : [0,1]^3 \rightarrow \mathbb{R}^c$
- Define a $h \times w \times l$ grid in $[0,1]^3$
- Parameterize field with a tensor $G \in \mathbb{R}^{h \times w \times l \times c}$
- Interpolate G to define the field on $[0,1]^3$

$$f(x) = \sum_{i,j,k} G_{ijk} K(x, x_{ijk})$$



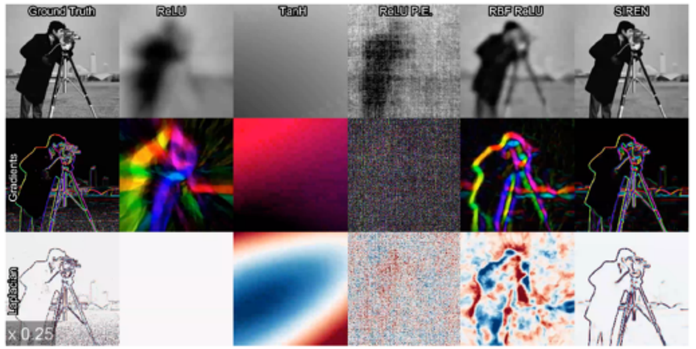
https://en.wikipedia.org/wiki/Trilinear_interpolation

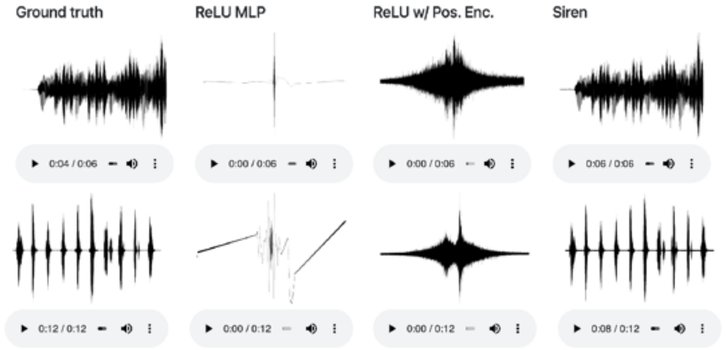
Neural Fields

- Takes a point x, returns a vector
- Parameterized MLP $f(\cdot)$
- Input encodings $\gamma(x)$ / trigonometric activations help

$$\gamma(x) = [\sin(2^k x), \cos(2^k x)]_{k=1}^{h}$$

- Applies to various domains such as photo, video, audio & 3D data





Sitzmann V. et al. Implicit neural representations with periodic activation functions //Advances in Neural Information Processing Systems. – 2020. – Т. 33. – С. 7462-7473.
Mildenhall B. et al. Nerf: Representing scenes as neural radiance fields for view synthesis //Communications of the ACM. – 2021. – Т. 65. – №. 1. – С. 99-106.

Hybrid Representations

Pros and cons of the above solutions

Is there a solution compromising between the two?

MLP

- Size: ~5mb
- Slow
- Trade-of speed for fidelity

Voxel Grid

- Size: ~1Gb
- Fast
- Trade-off memory for fidelity

Fridovich-Keil S. et al. Plenoxels: Radiance fields without neural networks // CVPR 2022
Müller T. et al. Instant neural graphics primitives with a multiresolution hash encoding //ACM Transactions on Graphics (ToG). – 2022. – Т. 41. – №. 4. – С. 1-15.

Hybrid Representations

Voxel Based

- Parameterize a grid with $G \in \mathbb{R}^{h \times w \times l \times c}$
- For an input point x get the representation with interpolation

$$g(x) = \sum_{i,j,k} G_{ijk} K(x, x_{ijk})$$

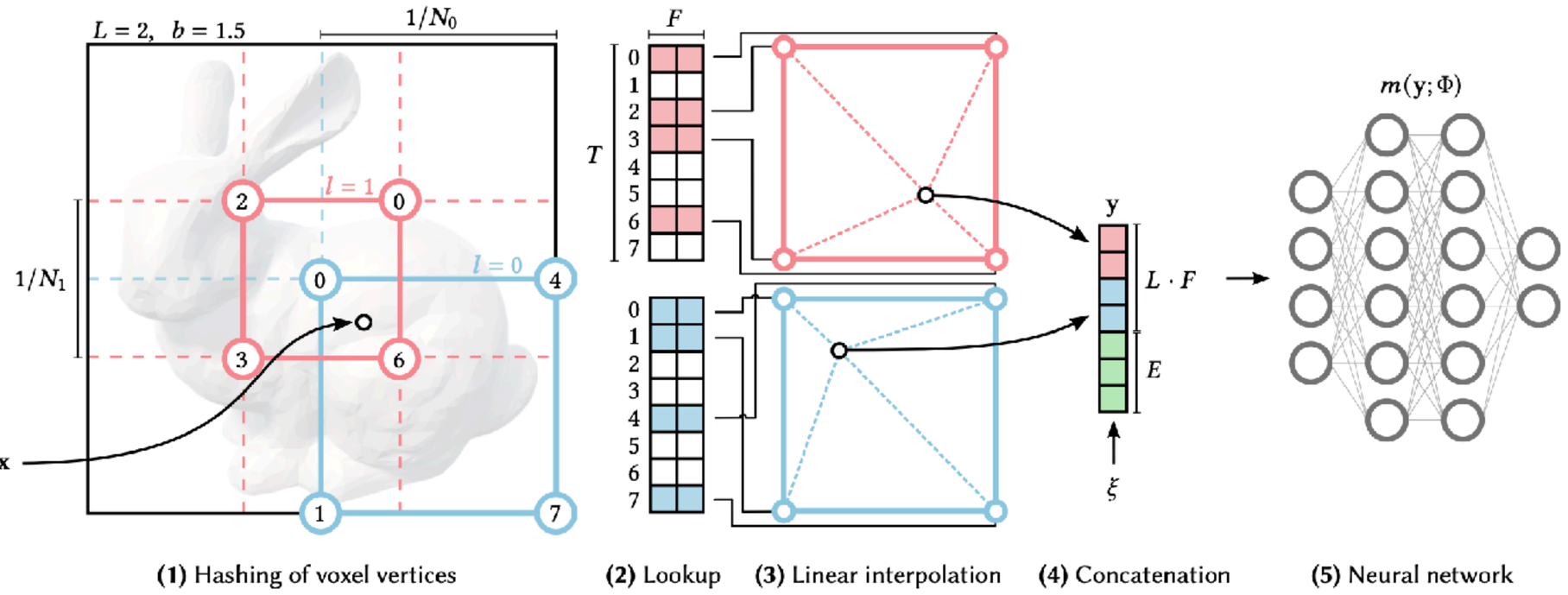
- Define an MLP $h(y)$ to map the representation to field vectors

$$f(x) := h(g(x))$$

- Middle ground between MLPs and voxel grids

Instant-NGP

Saving time and memory with HashGrids



HashGrid

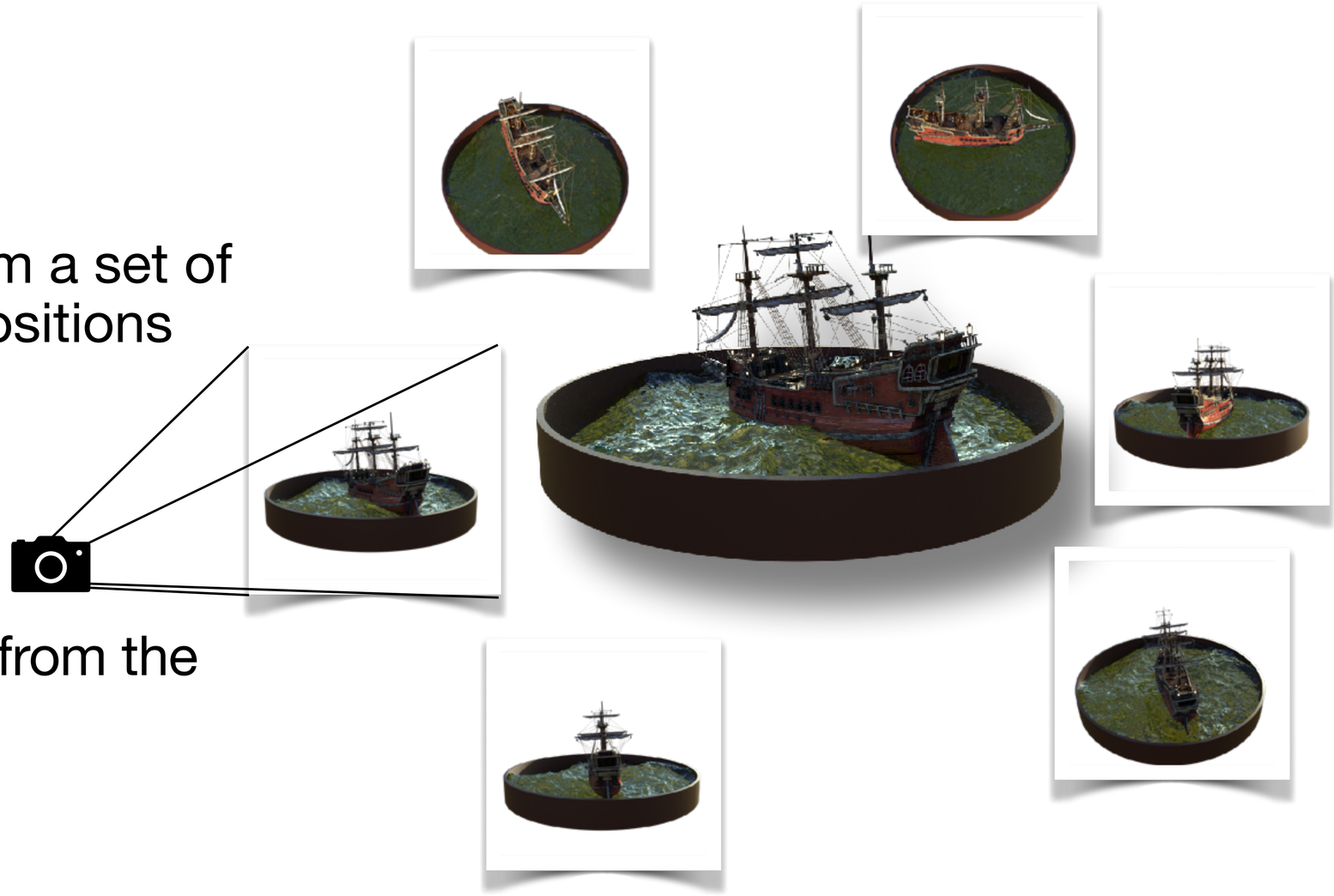
- Size ~100mb
- Average speed
- Flexibility factors
  - HashGrid size
  - MLP component

Müller T. et al. Instant neural graphics primitives with a multiresolution hash encoding //ACM Transactions on Graphics (ToG). – 2022. – Т. 41. – №. 4. – С. 1-15.
https://github.com/NVlabs/instant-ngp

Novel View Synthesis with Neural Radiance Fields

Novel View Synthesis

- Problem setup:
  - A set of pictures taken from a set of pre-determined camera positions
  - Test camera position
- Goal:
  - A picture of a scene taken from the test camera position



Mildenhall B. et al. Nerf: Representing scenes as neural radiance fields for view synthesis //Communications of the ACM. – 2021. – Т. 65. – №. 1. – С. 99-106.

Where Does the Data Come From?

- Structure From Motion (COLMAP)
  - Takes unposed images as input
- ARKit
  - Records device trajectory along with images

Neural Radiance Fields

Representing a Scene with Neural Fields

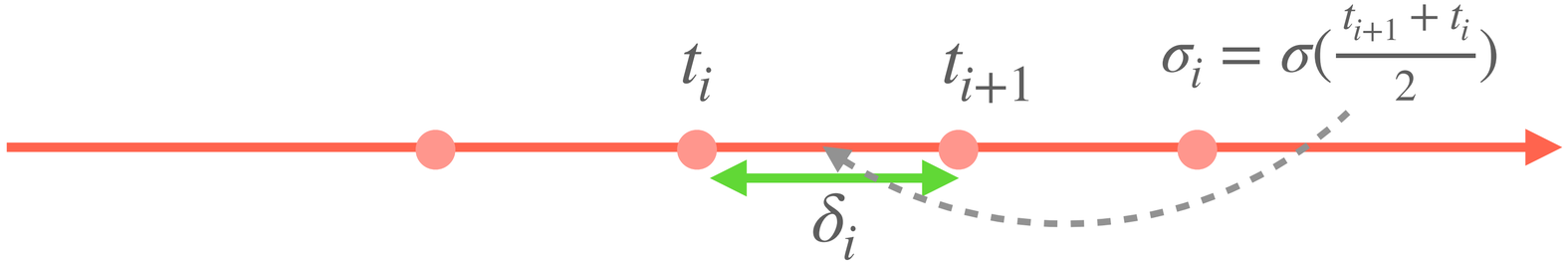
- Represent a scene with two fields
  - Density: $\sigma(x) : \mathbb{R}^3 \rightarrow \mathbb{R}^+$
  - Radiance: $C(x, d) : \mathbb{R}^3 \times S^2 \rightarrow \mathbb{R}^3$
- Density represents is related to opacity
- Radiance represents color of a point
- Architecture: MLP + positional embeddings

Mildenhall B. et al. Nerf: Representing scenes as neural radiance fields for view synthesis //Communications of the ACM. – 2021. – Т. 65. – №. 1. – С. 99-106.

Computing Pixel Color

- Density $\sigma(t) \in [0, +\infty)$ is related to opacity at t
- Divide the ray with points $t_1, \ldots, t_n$ and define

$$\alpha_i = 1 - \exp(-\sigma_i \delta_i)$$



- Expected color along a ray is given by

$$C = \sum_i c(t_i) \cdot \alpha_i \prod_{j<i} (1 - \alpha_j)$$

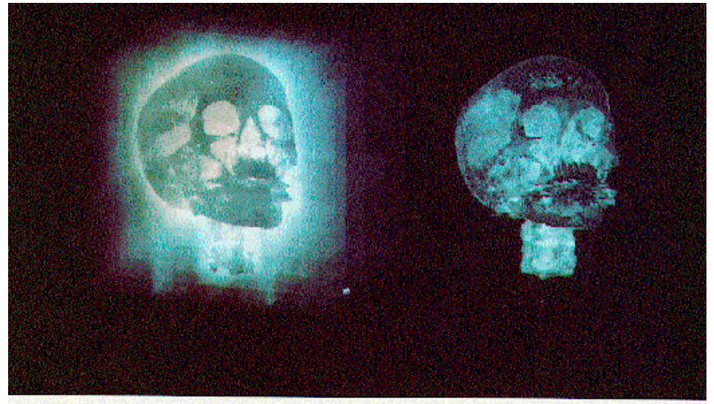


Fig. 11. Costs of rendering Figure 8 using hierarchical enumeration and adaptive termination.

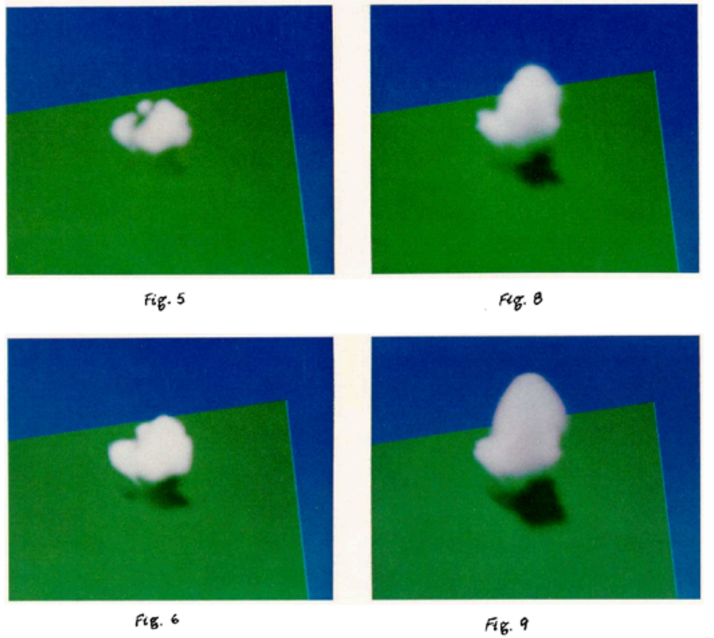


Fig. 5 Fig. 8

Fig. 6 Fig. 9

Neural Radiance Fields

Optimisation

- Training: minimise the mean squared error $\mathbb{E}_D \| C - C_{gt} \|^2$

$$C = \sum_i c_i \, \alpha_i \prod_{j<i} (1 - \alpha_j)$$

  - Optimise w.r.t. parameters of radiance C and density σ

Training Visualisation

- Synthetic data

- Real forward-facing

https://github.com/sunset1995/DirectVoxGO

NeRF pros and cons

+ Simplicity & Flexibility

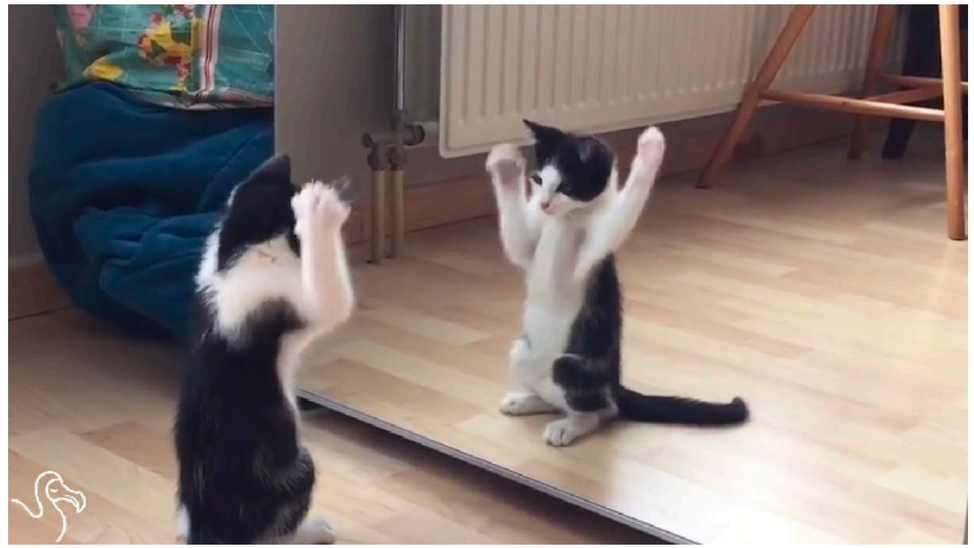
+ Photorealism

+ Compression *(5mb / scene)*

- Ill-posed problem

- Long training time *(48 h. / scene)*

- Slow rendering *(1 min. / frame)*

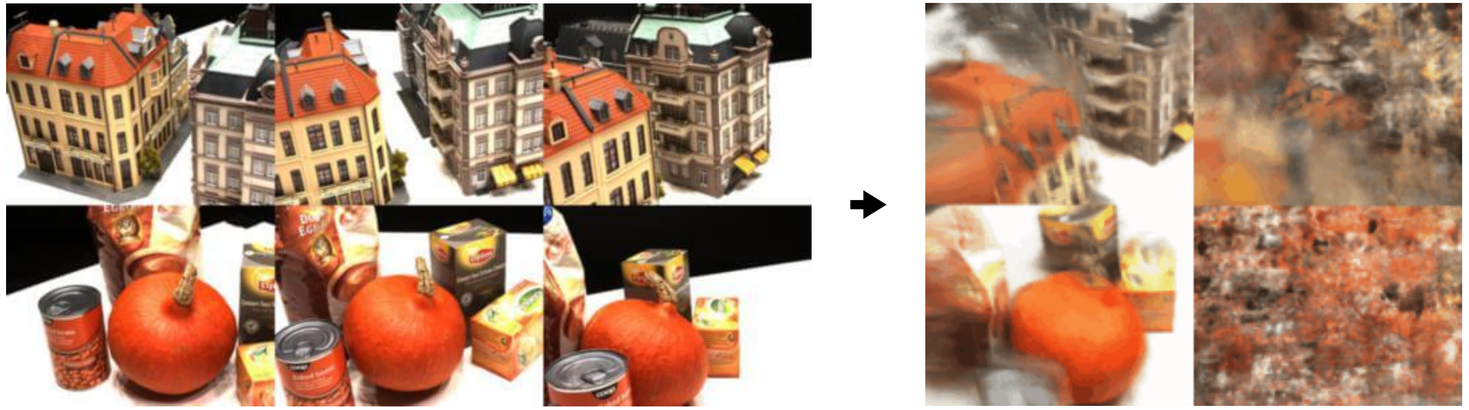


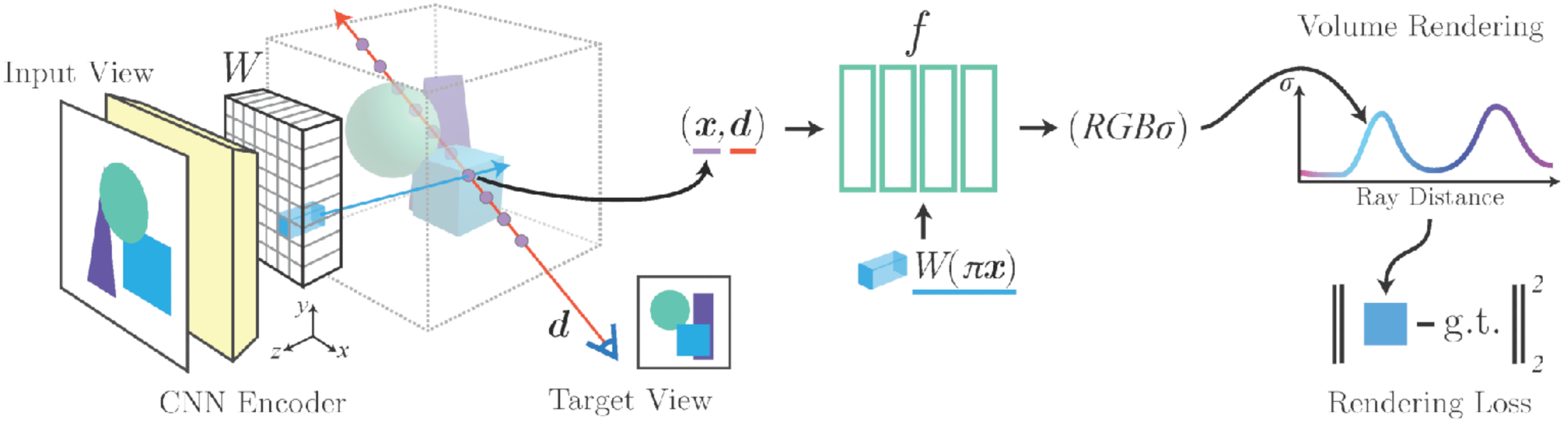
NeRF in The Wild & Block NeRF

- Radiance field handle embeddings capturing variability in lightning conditions
- Approach is scalable to large scene reconstruction

https://nerf-w.github.io/ https://waymo.com/research/block-nerf/

PixelNerf





https://alexyu.net/pixelnerf/

Language Embedded Radiance Fields

- One can replace MSE reconstruction loss with fancier training signals
  - CLIP-embeddings for text retrieval and 3D segmentation

Kerr J. et al. LERF: Language Embedded Radiance Fields //arXiv preprint arXiv:2303.09553. – 2023.

Improving Base Model: Time

Computational Overhead

$$\hat{C} = \sum_i c_i \alpha_i \prod_{j<i} (1 - \alpha_j)$$

- Picture contains $\approx 10^6$ pixels with 10^2 field evaluations per pixel
- Modern GPUs deliver poor frame rate at this cost
- How can we speed up rendering?

Sparsification

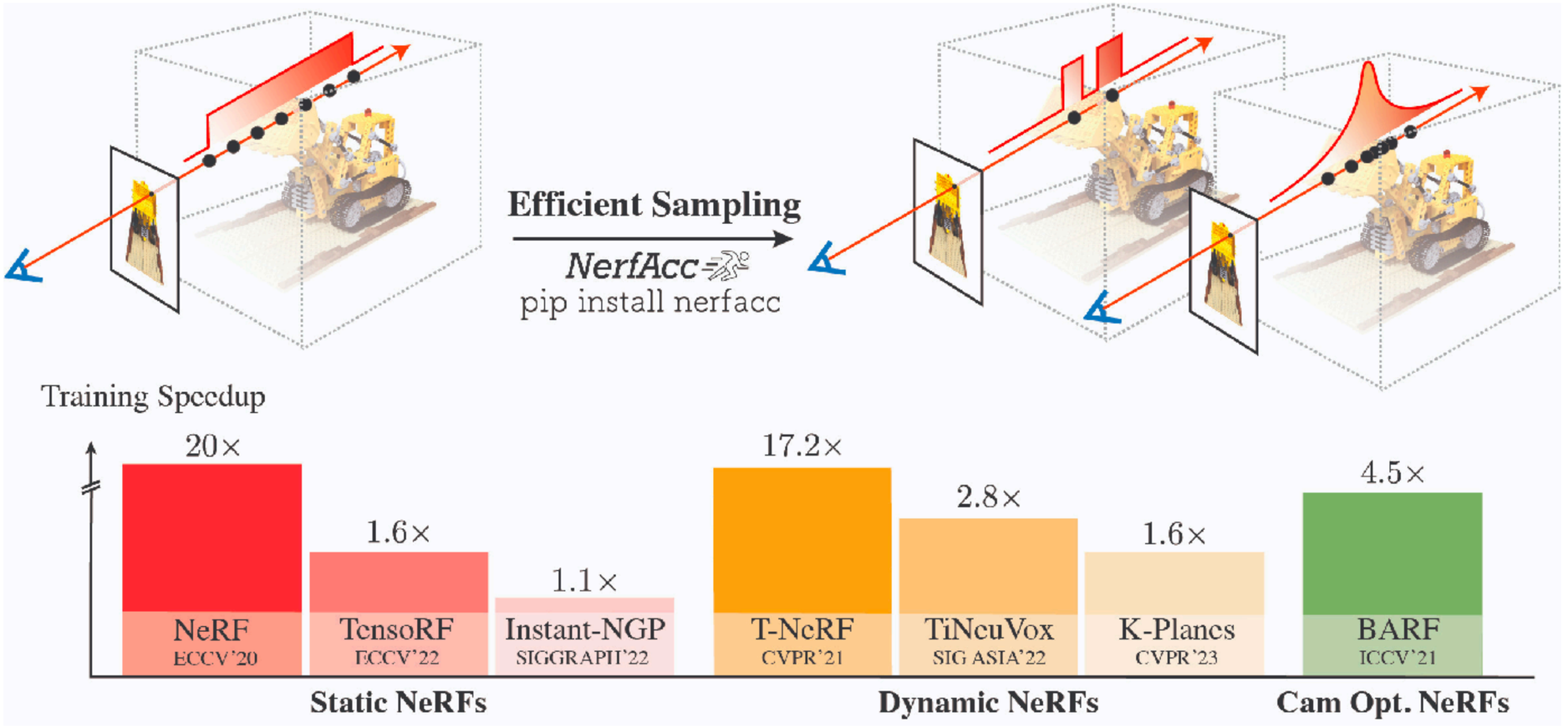
- Each terms in the sum requires running an MLP

$$C = \sum_i c_i \alpha_i \prod_{j<i} (1 - \alpha_j)$$

- If $\sigma(x_i) = 0$, then $1 - \alpha_i = 0$, we can omit i-th term
- Idea: cache $\sigma(\cdot)$, omit i if we know $\sigma(x_i) = 0$
- Outcome: training and rendering 10-30 times faster

Liu L. et al. Neural sparse voxel fields //Advances in Neural Information Processing Systems. – 2020. – T. 33. – C. 15651-15663.
Hedman P. et al. Baking neural radiance fields for real-time view synthesis //Proceedings of the IEEE/CVF International Conference on Computer Vision. – 2021. – C. 5875-5884.
Sun C., Sun M., Chen H. T. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2022. – C. 5459-5469.
Fridovich-Keil S. et al. Plenoxels: Radiance fields without neural networks //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2022. – C. 5501-5510.
Li R., Tancik M., Kanazawa A. NerfAcc: A General NeRF Acceleration Toolbox //arXiv preprint arXiv:2210.04847. – 2022.

NerfAcc

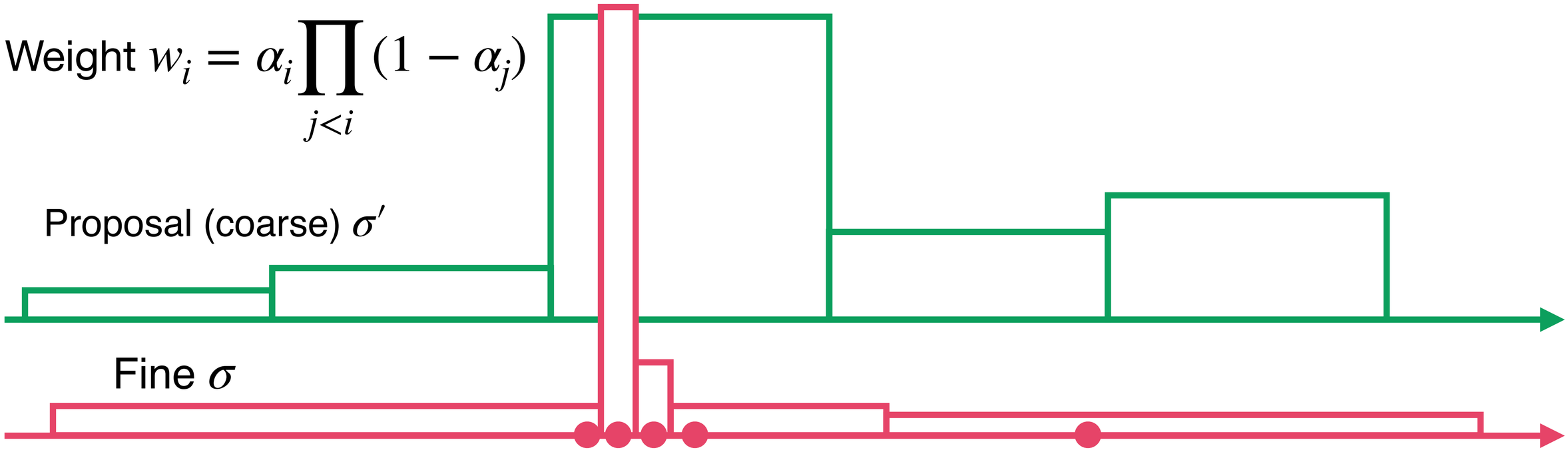


Li R. et al. NerfAcc: Efficient Sampling Accelerates NeRFs // ICCV – 2023

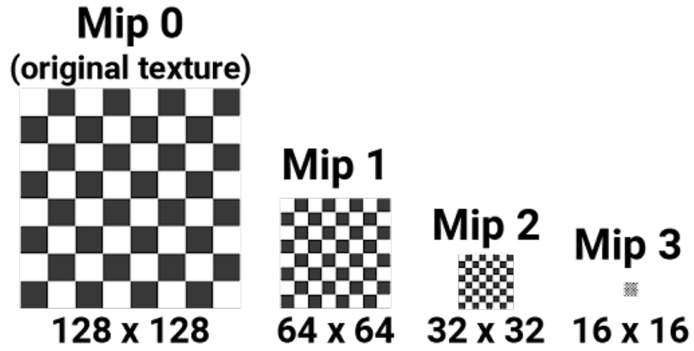
Improving Base Model: Fidelity

Hierarchical Sampling (3)

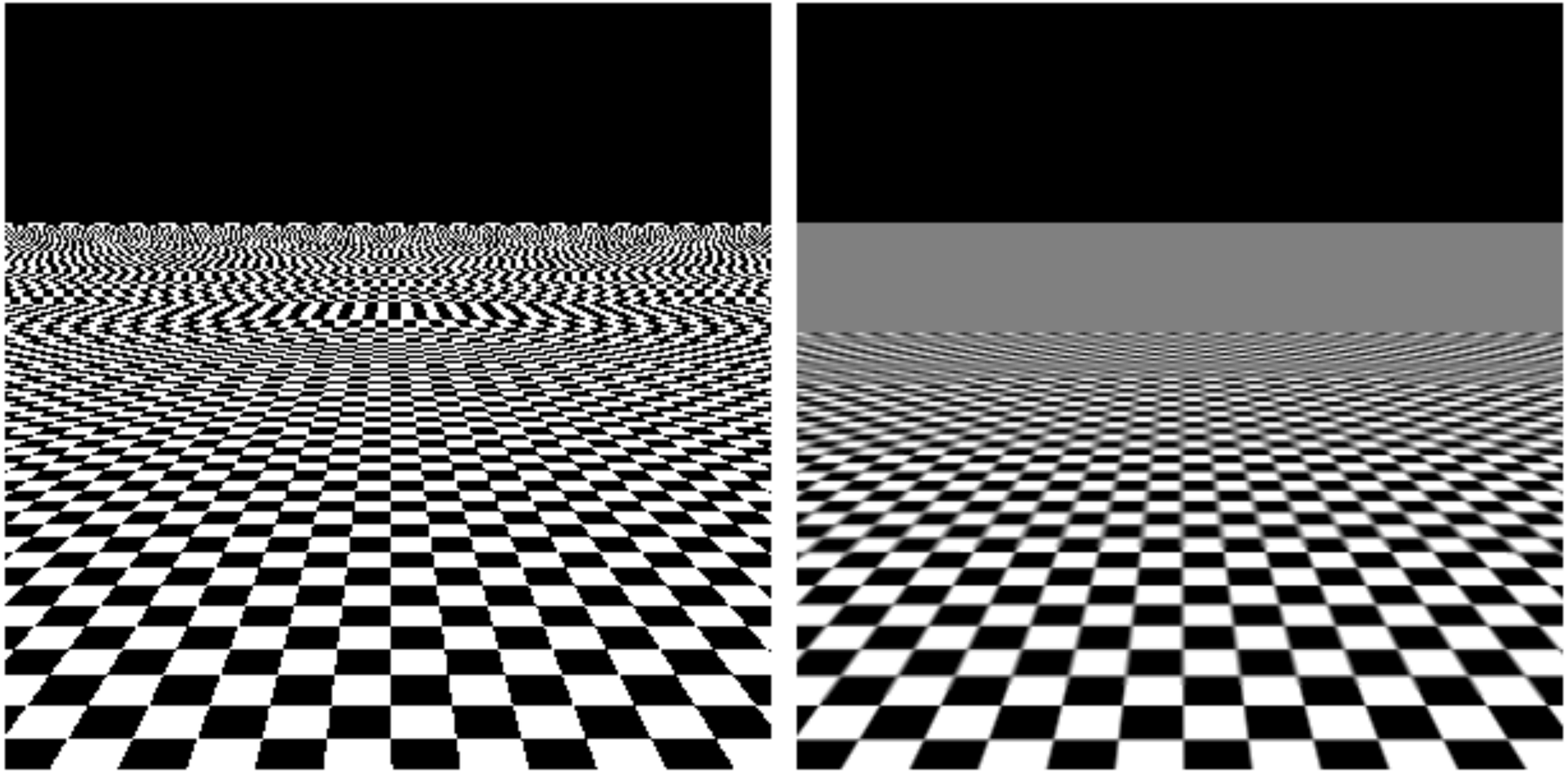
- Until now, we picked m knots uniformly on the ray
- Hierarchical approach picks the knots adaptively
- Weight $w_i = \alpha_i \prod_{j<i} (1 - \alpha_j)$



Aliasing

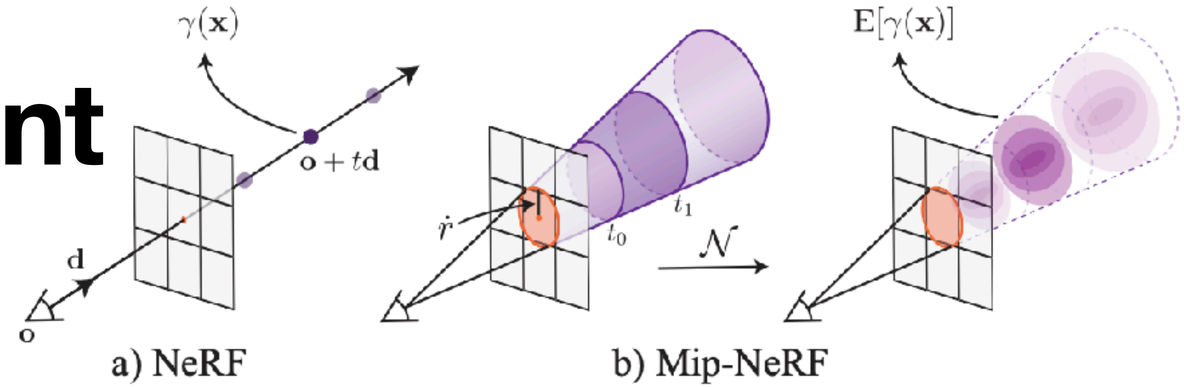


multum in parvo



Taking Scale into Account

A Parallel Line of Reseach



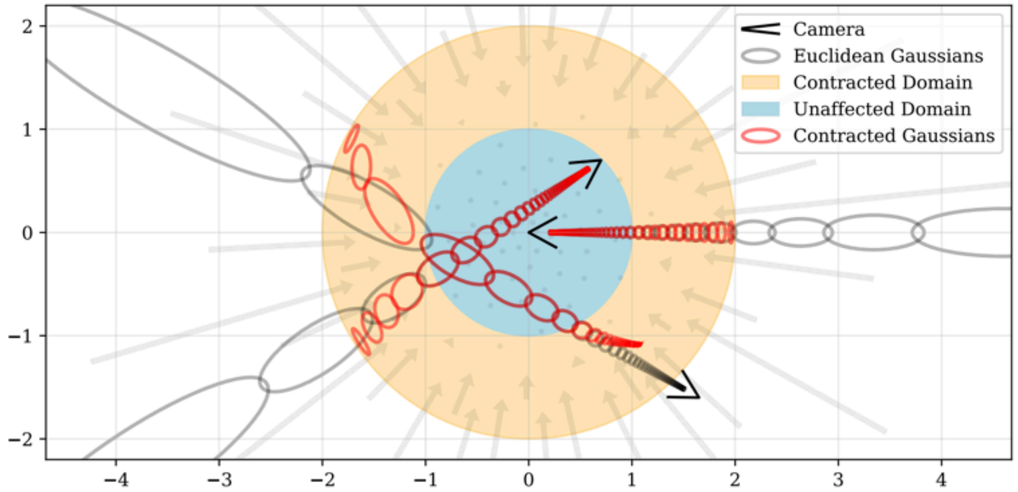
- Original NeRF experiments had biased data
  - Camera distances did not change by a significant amount
- Pixels cover a cone-like region in space
- Ideal solution - compute the average radiance over the whole cone
- Mip-NeRF approximates the average radiance/density over a cone segment
- Idea: compute embedding average rather than output average

Barron J. T. et al. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields //Proceedings of the IEEE/CVF International Conference on Computer Vision. – 2021. – С. 5855-5864.

https://jonbarron.info/mipnerf/

Unbounded Scenes

- Original NeRF experiments had biased data
  - All the training scenes were bounded
- NeRF++ models foreground and background with two networks
- Mip-NeRF-360 maps the space into a ball
  - Does not transform scene center
  - The rest is mapped onto a shell



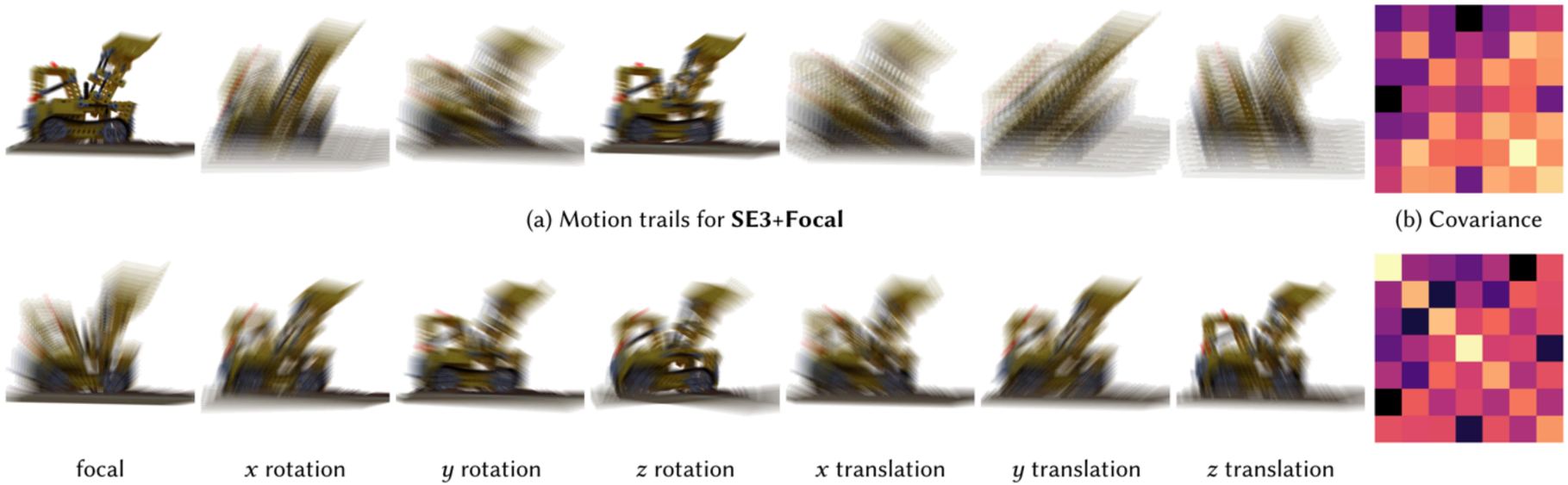
Zhang K. et al. Nerf++: Analyzing and improving neural radiance fields //arXiv preprint arXiv:2010.07492. – 2020.
Barron J. T. et al. Mip-nerf 360: Unbounded anti-aliased neural radiance fields //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2022. – С. 5470-5479.

Zip-NeRF

- Merges two research branches
  - Speed: Instant-NGP
  - Fidelity: Mip-NeRF-360
- Mip-NeRF is designed for MLPs
- Zip-NeRF adapts Mip-maps to Instant-NGP

Barron J. T. et al. Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields //arXiv preprint arXiv:2304.06706. – 2023. https://jonbarron.info/zipnerf/

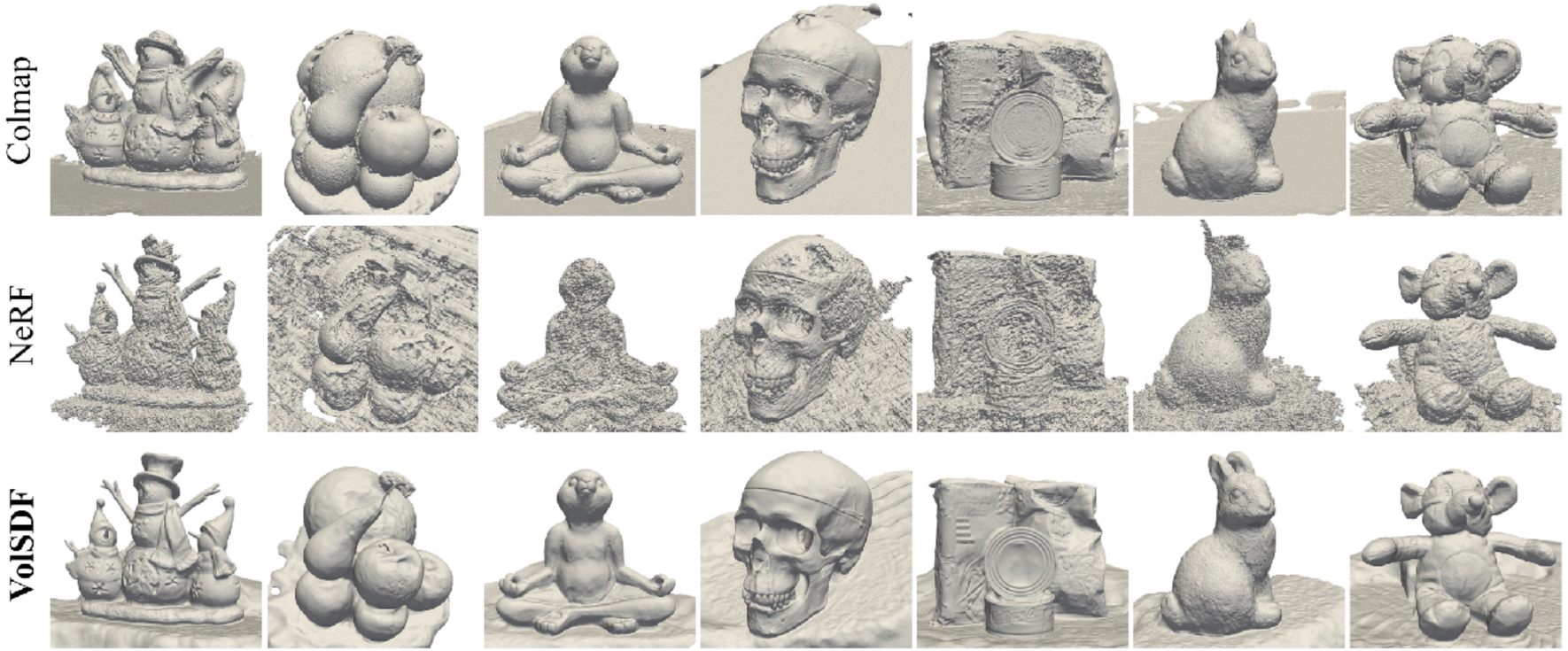
Fine-tuning Camera Positions



https://camp-nerf.github.io/

Improving Scene Geometry

Novel View Synthesis



https://lioryariv.github.io/volsdf/

Combining Density Fields and SDFs

- Let d_Ω be the signed distance field for shape Ω
- Parameterise density as $\sigma(x) = \alpha\Phi\left(-\frac{d_\Omega(x)}{\beta}\right)$
- Where $\Phi(x)$ is the Laplace distribution CDF

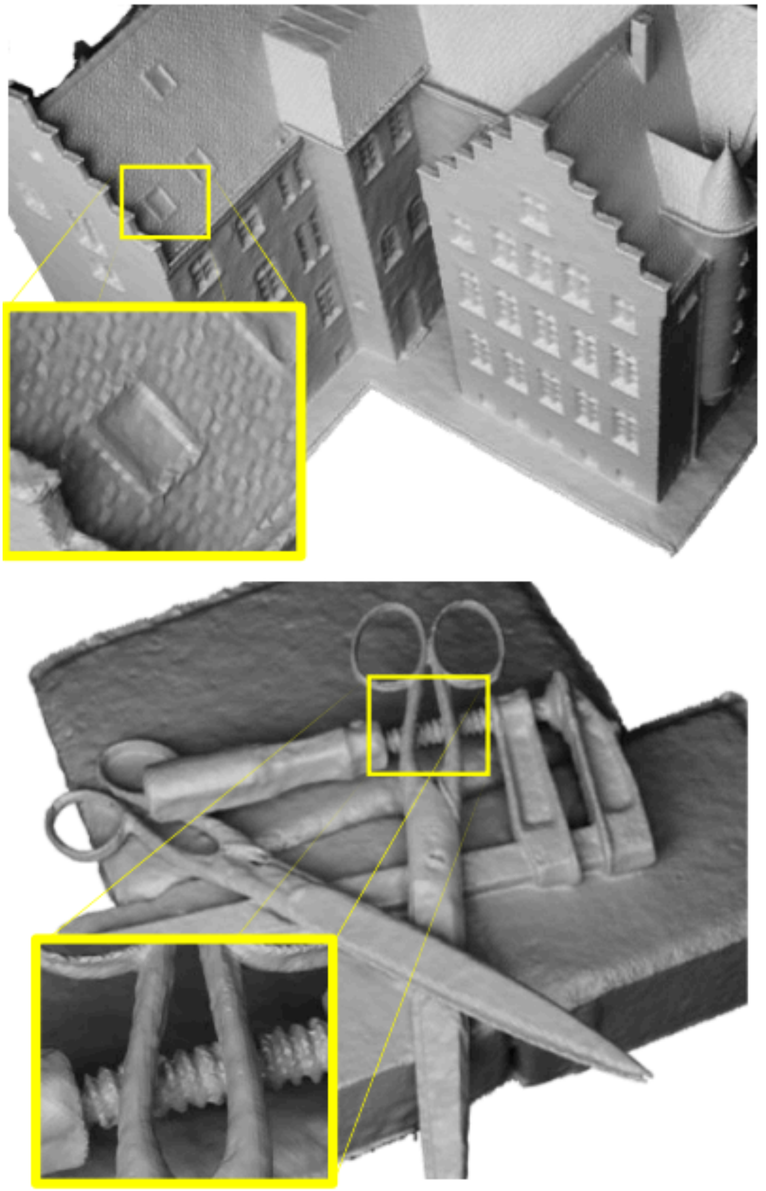
$$\Psi(s) = \begin{cases} \frac{1}{2}\exp(s) & ifs \leq 0 \\ 1 - \frac{1}{2}\exp(-s) & ifs > 0 \end{cases}$$

https://lioryariv.github.io/volsdf/ https://research.nvidia.com/labs/dir/neuralangelo/

Eikonal Equation

- How does the gradient of SDF look like?
  - The gradient $\nabla d_\Omega(x)$ points away from the closest surface point
  - The norm of the gradient is constant: $\|d_\Omega(x)\|^2 = 1$
- Eikonal loss:

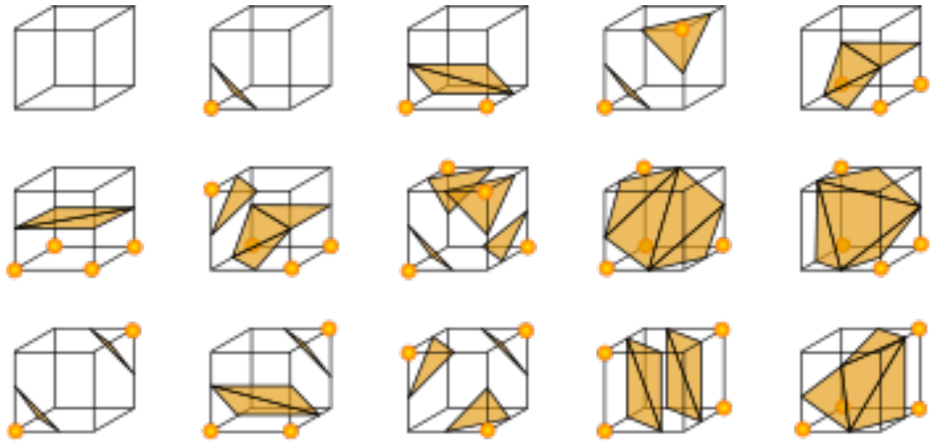
$$\mathscr{L}_{SDF}\mathbb{E}_z \left(\|\nabla d_\Omega(z)\| - 1\right)^2$$



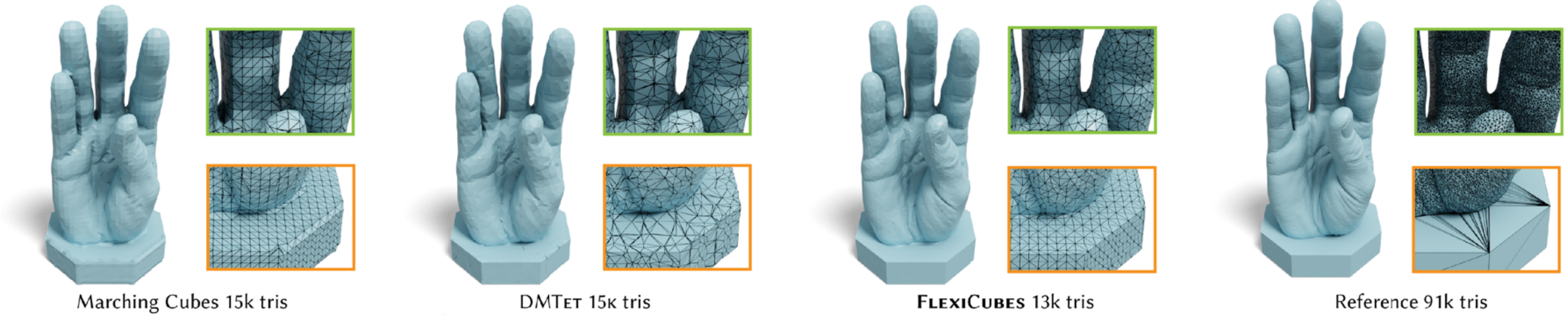
https://lioryariv.github.io/volsdf/ https://research.nvidia.com/labs/dir/neuralangelo/

Marching Cubes

Converting Fields to Polygonal Meshes



Isosurfacing w/ gradients



Marching Cubes 15k tris

DMTet 15k tris

FlexiCubes 13k tris

Reference 91k tris

https://en.wikipedia.org/wiki/Marching_cubes https://research.nvidia.com/labs/toronto-ai/flexicubes/

Key Takeaways

- Fields and volumetric representations for 3D data
  - Voxel, neural and hybrid representations
- Volumetric rendering for novel view synthesis
- Various applications and incremental improvements
  - Amortised models
  - CamP + ZipNeRF still delivers best quality as of May 2024
- Next time: Gaussian Splatting