Instant neural representation

2022 하계 세미나

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

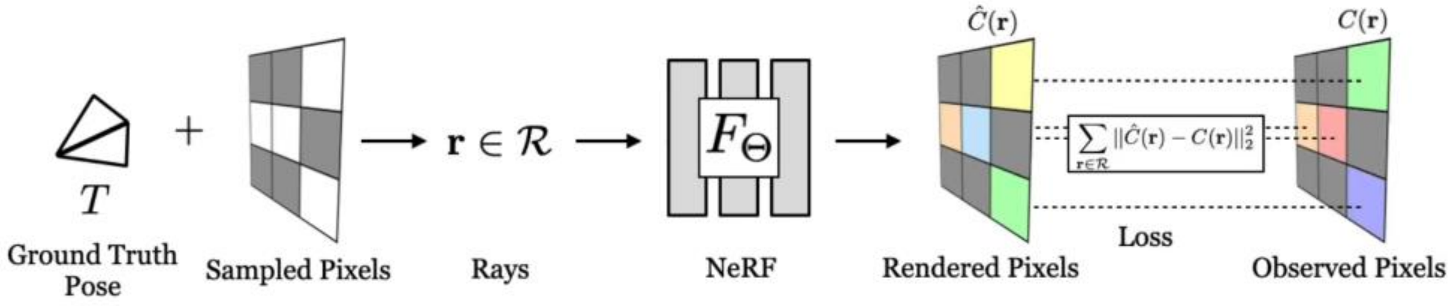
Presented By
조민지

Outline

- Introduction
  - NeRF ⇨ Plenoxels ⇨ Instant-NGP
  - Plenoxels
- Method
  - Instant-NGP
    - Multi-resolution hash encoding
    - How to accelerate ray marching
- Experiments
- Conclusion

Introduction

- Neural Radiance Fields for View Synthesis
  - Learn a scene from image observations and corresponding perspective transforms
    - MLP learns the 3D density and 5D light field of a given scene from input views
  - Training and rendering scenes
    - Takes 1~4 days for training one scene



1) Mildenhall, Ben, et al. "Nerf: Representing scenes as neural radiance fields for view synthesis.“ ECCV 2020.
2) Yu, Alex, et al. "Plenoxels: Radiance fields without neural networks.“ CVPR 2022.
3) Müller, Thomas, et al. "Instant neural graphics primitives with a multiresolution hash encoding.“ SIGGRAPH 2022.

Introduction

- [1]NeRF ⇨ [2]Plenoxels ⇨ [3]Instant-NGP
  - Improvement on runtime of training and rendering
    - Runtime: Days ⇨ Minutes ⇨ Seconds
    - In both models, key component of improving NeRF is not the neural network, differential volumetric rendering.

Elapsed training time: 0 seconds

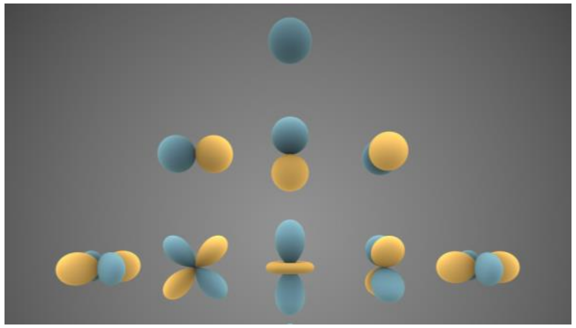
NeRF
1 day

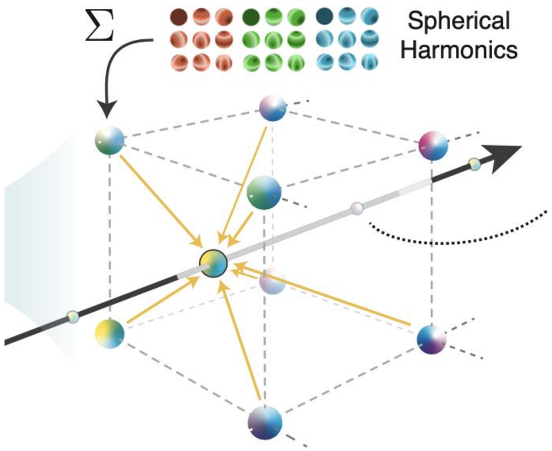
Plenoxel
9 minutes

Instant-NGP
10 seconds

Introduction

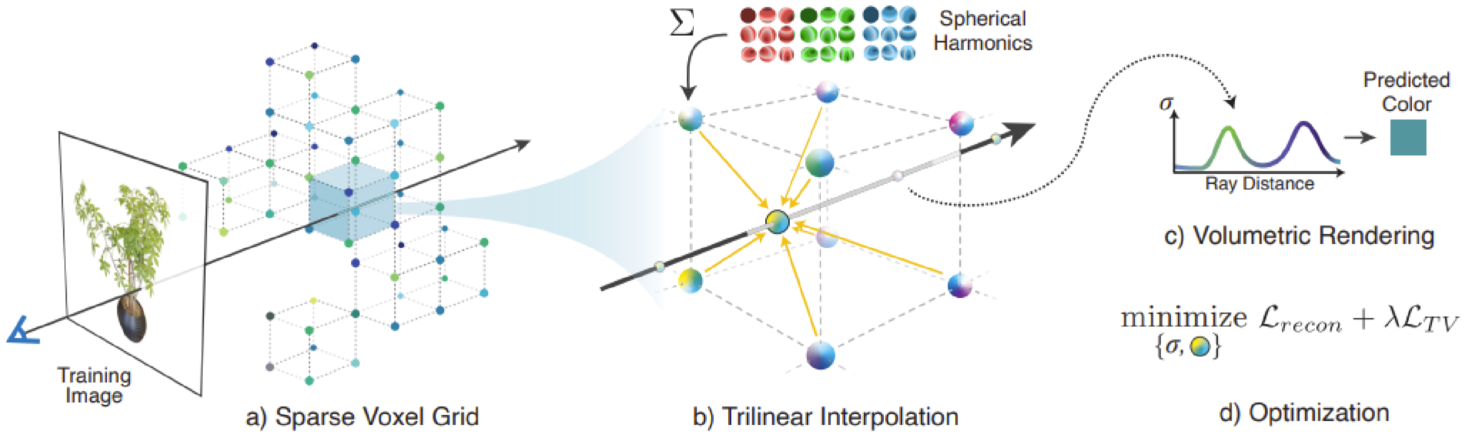
- Sparse voxel grid optimization
  - Plenoxels = Plenoptic + Voxels
    - Scene can be represented as a sparse grid of volume elements (=voxels)
      - Each voxel vertice stores values of scalar density and harmonic coefficients for color
        - ✓ Spherical harmonics: 9 basis vectors for each color channel
      - Spherical harmonics describe changes in color from different view directions
        - ✓ Plenoptic function: function that defines the light field





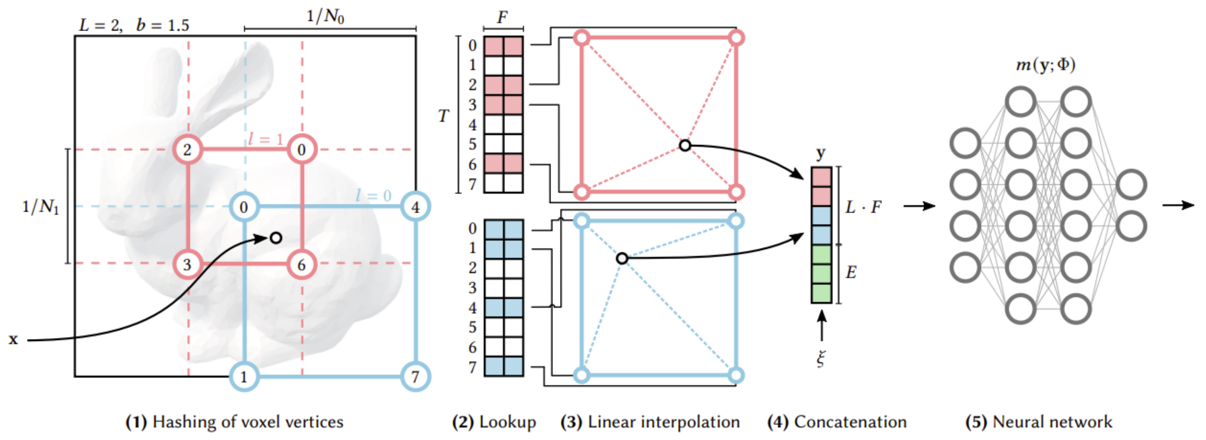
Introduction

- Sparse voxel grid optimization
  - Plenoxels = Plenoptic + Voxels
    - Scene can be represented as a sparse grid of volume elements (=voxels)
      - Trilinear interpolation to get (c, σ) of a sample on a ray
      - Use MSE loss to train a sparse voxel grid, no need of neural network



Introduction

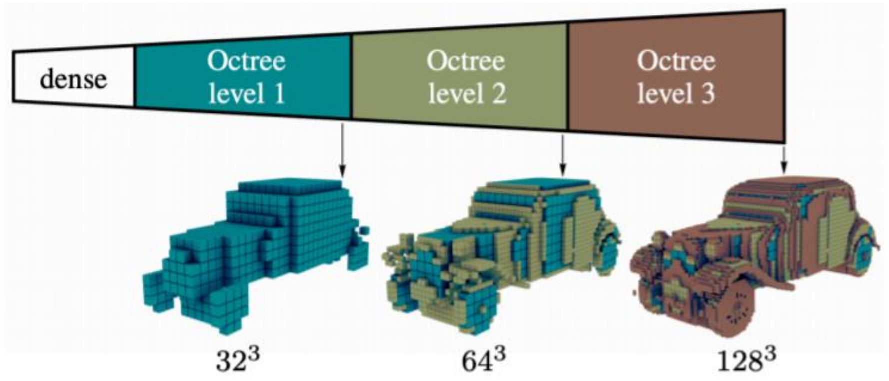
- Instant Neural Graphics Primitives
  - Also learn sparse voxel grid, but 10x faster for train & render
    - Use trained feature vectors rather than spherical harmonics vectors
    - Encode feature vectors of each voxel vertices and interpolate them



Method

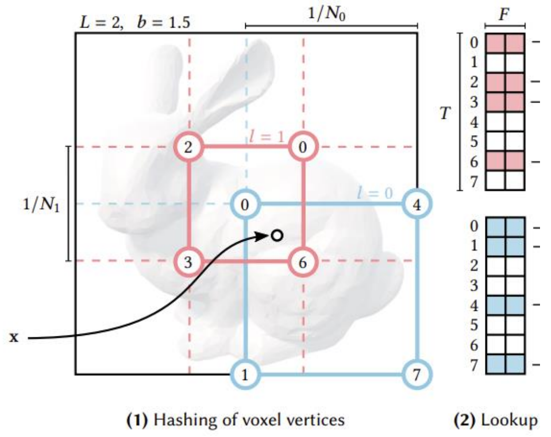
- Instant Neural Graphics Primitives
  - Multi-resolution hash encoding
    - Sparse parametric encoding
      - Dense grid is wasteful as its parameters grows as $O(N^3)$
        - ✓ Allocates as many parameters to unimportant areas of empty space
        - ✓ Visible surface of interest only takes up to only about 3% of total grid
      - Smooth surface can be decomposed into multi-resolution grids

Use voxel grids with heterogenous resolution!



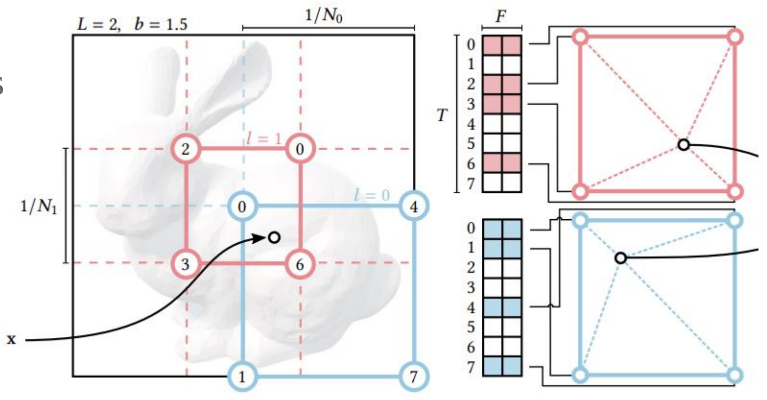
Method

- Instant Neural Graphics Primitives
  - Multi-resolution hash encoding
    - Hash encoding
      - Store the trainable feature vectors of voxel vertices in multiple separate hash tables
        - ✓ Hash table
          - Data structure that maps keys to values
          - Use hash function to compute an index for key
          - Size of table = T, computation = T
            - $O(1)$, computation efficient!
        - ✓ One table per one resolution level voxel grid



Method

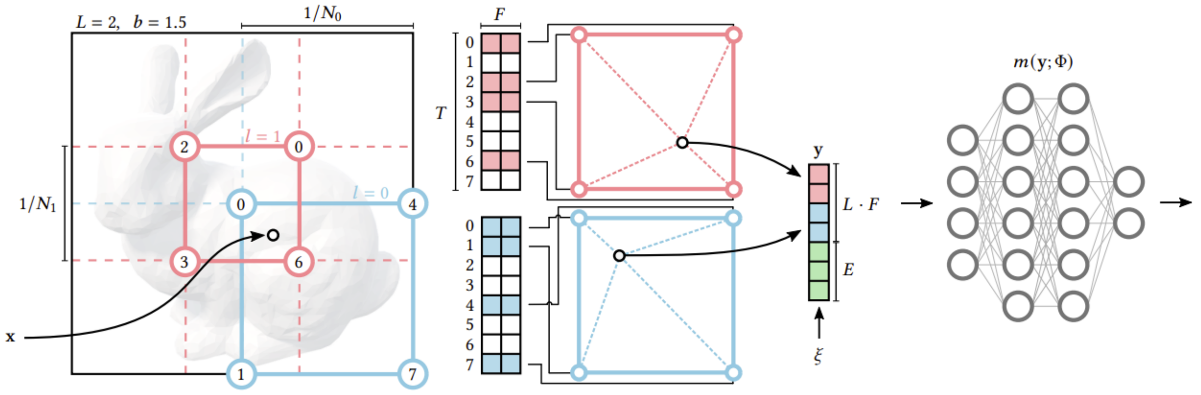
- Instant Neural Graphics Primitives
  - Multi-resolution hash encoding
    - Hash encoding
      - Store the trainable feature vectors of voxel vertices in multiple separate hash tables
        - ✓ If hash function ($h : Z^d \rightarrow Z_T$) maps more vertices than size of hash table, hash collision could likely occur
        - ✓ No explicit collision handling…
          - Points in visible areas of the scene, gradients are larger than collision average
          - Effect of collision can be alleviated
        - ✓⇨ See experiments for the effect



(1) Hashing of voxel vertices (2) Lookup (3) Linear interpolation

Method

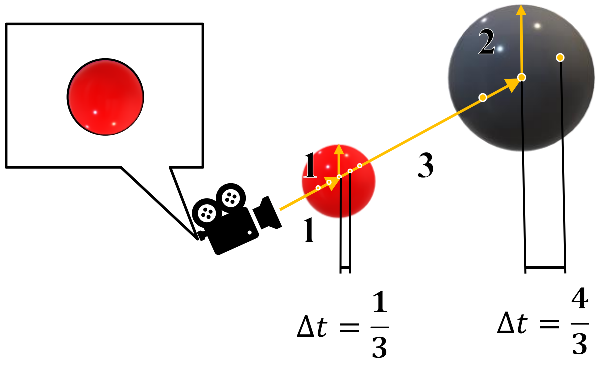
- Instant Neural Graphics Primitives
  - Multi-resolution hash encoding
    - Hash encoding
      - Represent sample coordinate x as d-linear interpolation of d voxel vertices
        - ✓ Concatenate interpolated feature vectors of each level grids and auxiliary inputs and use as the input of MLP
        - ✓ While training, loss gradients are backpropagated all through hash lookup tables and accumulated in feature vectors



(1) Hashing of voxel vertices (2) Lookup (3) Linear interpolation (4) Concatenation (5) Neural network

Method

- Instant Neural Graphics Primitives
  - How to accelerate ray marching
    - Exponential stepping for large real scenes
      - Set the step size proportional to the distance t along the ray
        - ✓ Intercept theorem: If size and distance of objects is propotional, appearance of them stays the same
        - ✓ Step size grows propotional to distance t, thus sparse sampling for far objects
          - $(\Delta t = t/c)$



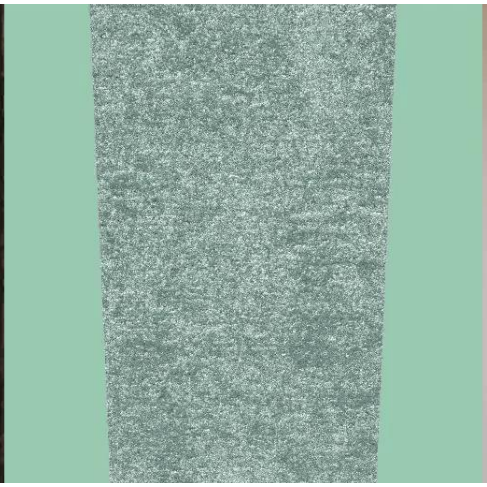
Method

- Instant Neural Graphics Primitives
  - How to accelerate ray marching
    - To skip ray marching steps in empty space, use a multiscale occupancy grids
      - Occupancy grids
        - ✓ Single bit which tells whether each cell is free or occupied
      - Skipping gradient updates for empty space and occluded regions
        - ✓ Update occupancy grid for certain iterations
          - Decay the density value and update the occupancy bits by threshold with 0.01
        - ✓ If the grid cell where sample is placed according to step size has low bit, the sample is skipped

Experiments

- Training results for multiple tasks
  - Realtime training results for distinct graphics primitives

Neural gigapixel images


Neural SDF

Neural Radiance Fields

Experiments

- Gigapixel Image Approximation
  - Learns 2D to RGB mapping of image coordinates to colors
    - Popular benchmark for testing a model’s ability to represent high-frequency detail
      - Approximating an RGB image of resolution 20000 × 23466 (total 469M RGB pixels)
    - This paper achieves similar result as previous SOTA model with much shorter training
      - [1]Adaptive coordinate networks (ACORN) takes 36 hours for PSNR 38.59
      - With similar number of parameters, this paper takes 2.5 minutes for same PSNR and takes 4 minutes for convergence
    - Whether [1]ACORN needs adaptive subdivision of the scene, no need of such effort for training

Experiments

- [1]Signed Distance Function (SDF)
  - Learns an SDF in 3D space whose zero level-set represents a 2D surface

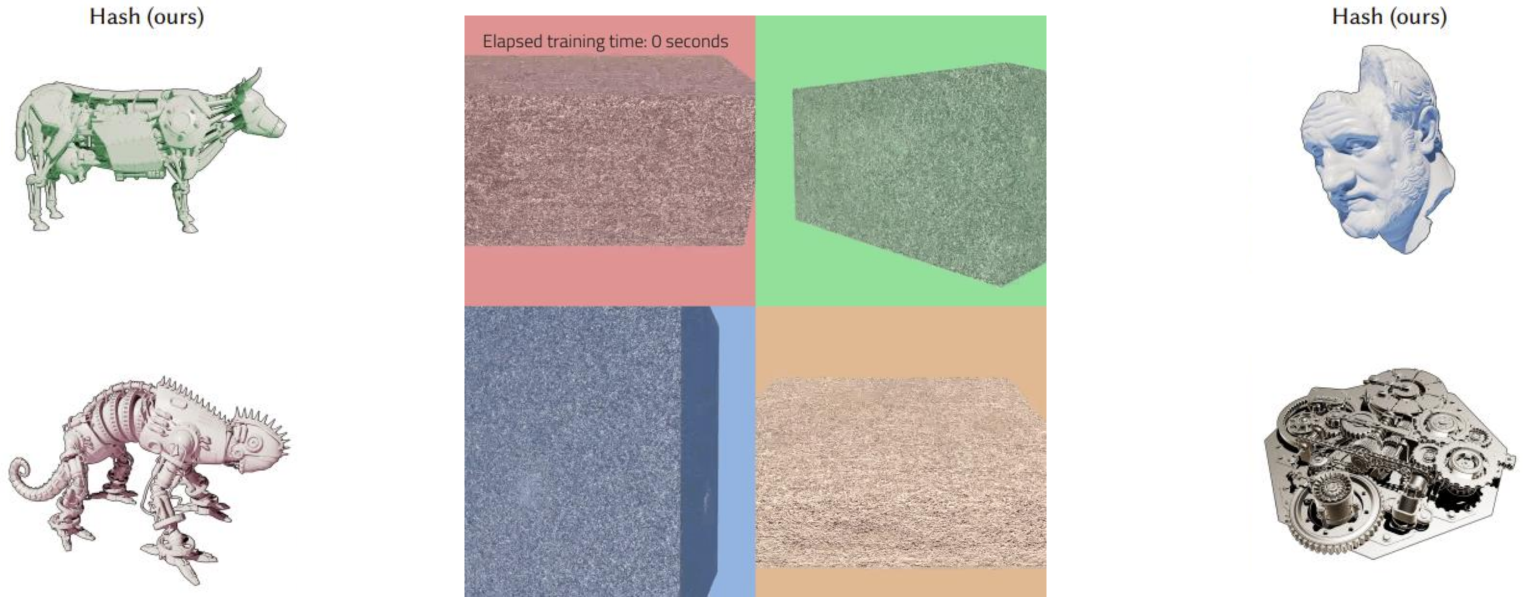


Fig. 7. Neural signed distance functions trained for 11 000 steps. The frequency encoding [Mildenhall et al. 2020] struggles to capture the sharp details on these intricate models. NGLOD [Takikawa et al. 2021] achieves the highest visual quality, at the cost of only training the SDF inside the cells of a close-fitting octree. Our hash encoding exhibits similar numeric quality in terms of intersection over union (IoU) and can be evaluated anywhere in the scene. However, it also exhibits visually undesirable surface roughness that we attribute to randomly distributed hash collisions. Bearded Man ©Oliver Laric (CC BY-NC-SA 2.0)

Experiments

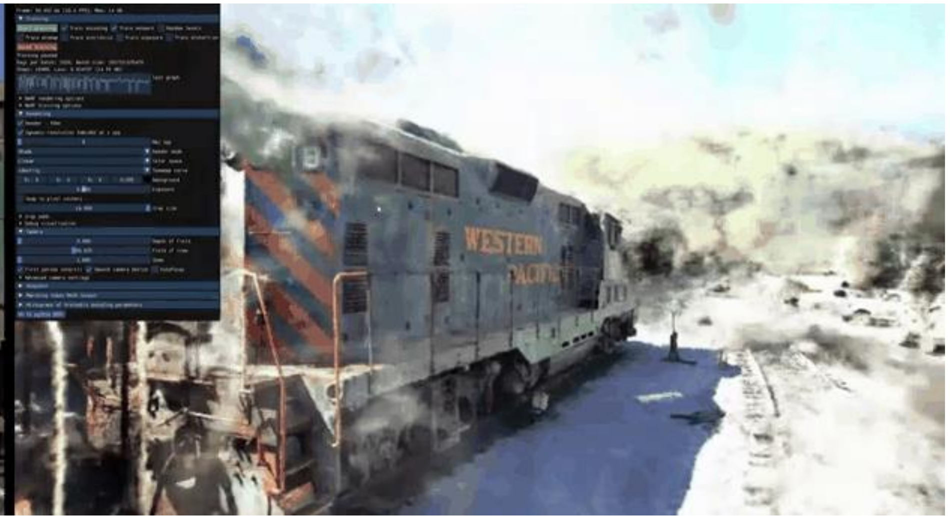
- Neural Radiance Fields

1) Rückert, et al., "ADOP: Approximate differentiable one-pixel point rendering." *arXiv* 2021.

Experiments

- Microstructure due to hash collisions

713
WESTERN
PACIFIC



12 hours trained [1]ADOP

10 seconds trained Instant-NGP

1) https://github.com/NVlabs/instant-ngp

Conclusion

- Improvement on both training and rendering runtime by sparse hash encoding
  - Multi-resolution hash tables encode 3D representation by sparse voxel grids which can be queried in parallel thus maps well to GPUs

- Neural rendering application을 고려하고 계시다면 학습 feasibility 확인하시는 용도로 추천 드립니다.[1]
  - 기존 NeRF 말고도 NGLOD 등 neural SDF 모델 및 Dynamic scene을 다루는 DNeRF 등 다양한 task에 implement 되어있음

서강대학교 SOGANG UNIVERSITY

VDS LAB

감사합니다 :)