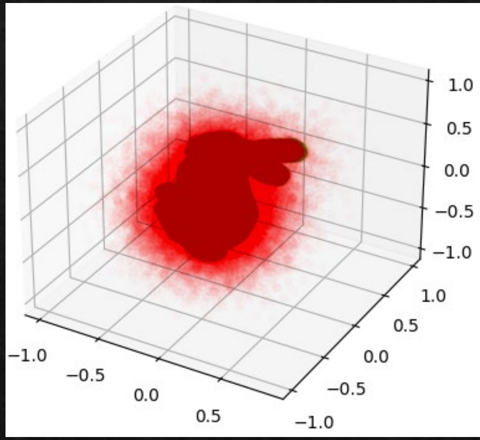
Comparison of NeRF models for Surface Reconstruction

Sihui Wang

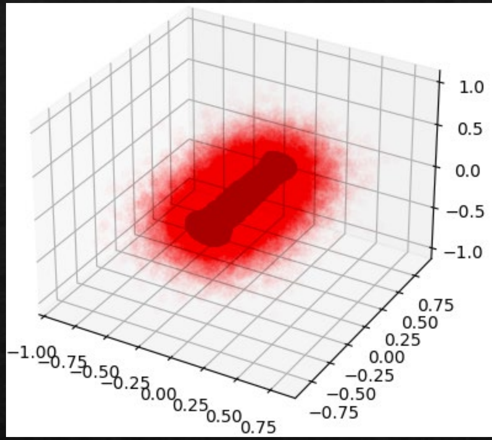
April 6th, 2023

Single LoD Dense Grid NeRF for Surface Reconstruction

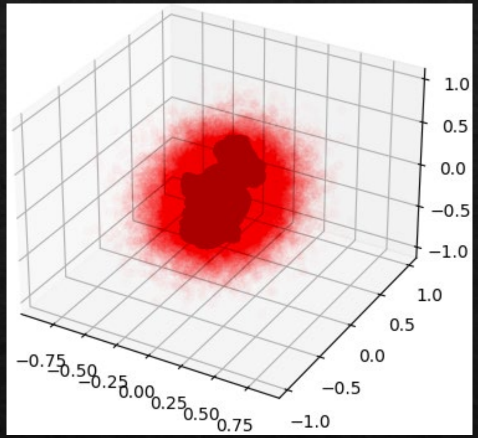
Qualitative Results



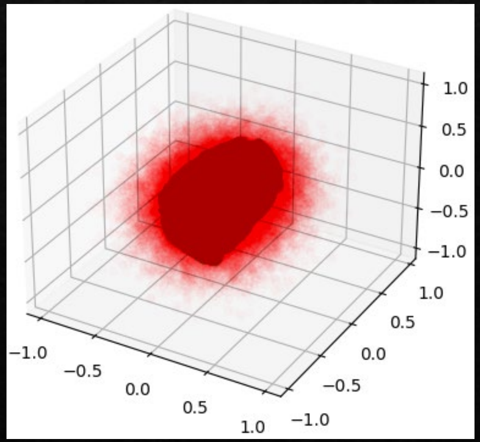
Bunny



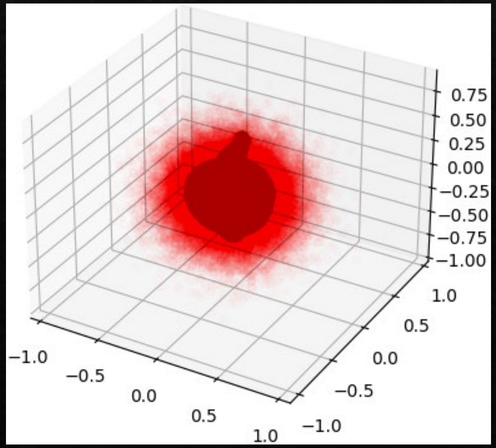
Column



Dragon_original



Serapis



Utah_teapot

Single LoD Dense Grid NeRF for Surface Reconstruction

Quantitative Results: Reconstruction Error

	Bunny	Column	Dragon_original	Serapis	Utah_teapot
Chamfer	0.0225	0.0097	0.0104	0.0312	0.0275
Hausdorff	0.2576	0.0878	0.1020	0.3054	0.2068

Implementation Choices:

Grid: Resolution: $256 \times 256 \times 256$

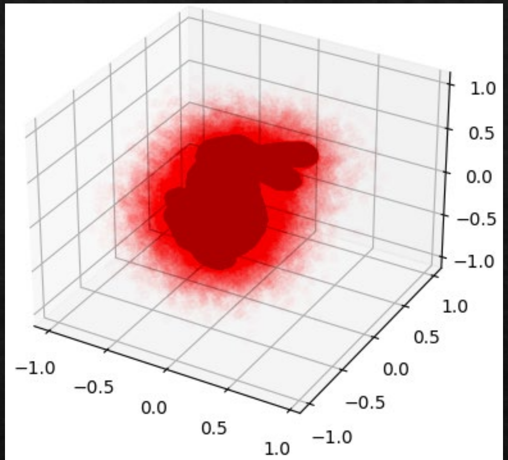
MLP Structure: Input feature dimension: 3; Hidden layer feature dimension: 64; Number of hidden layers: 3

Regularization: Number of sampling points: 10000; Shifting scale: 0.15; Regularization Weight: 0.02

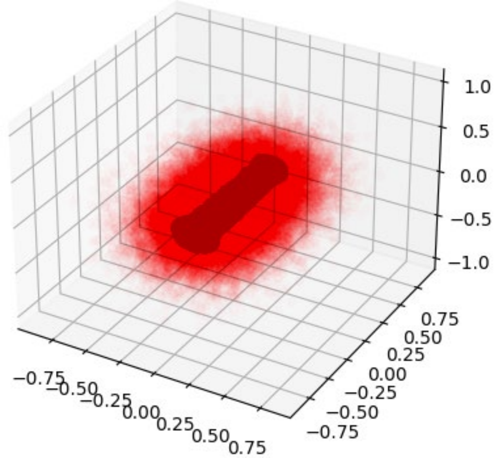
Training Termination Criteria: loss < 0.02, or loss < 0.05 and change of loss < 0.00001

Multiple LoD Dense Grid NeRF for Surface Reconstruction

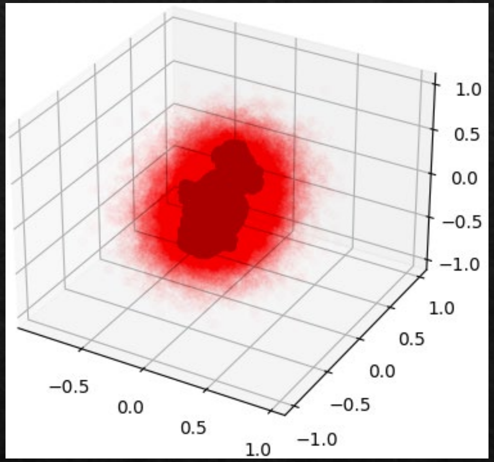
Qualitative Results



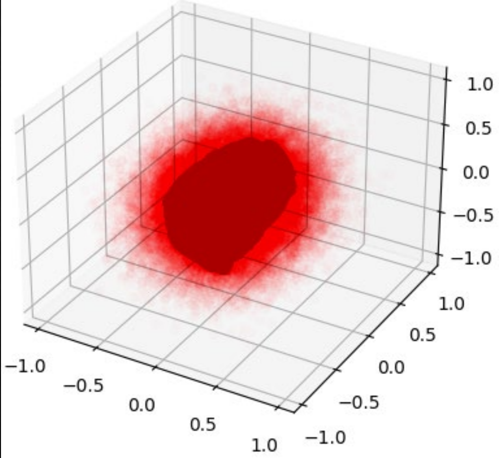
Bunny



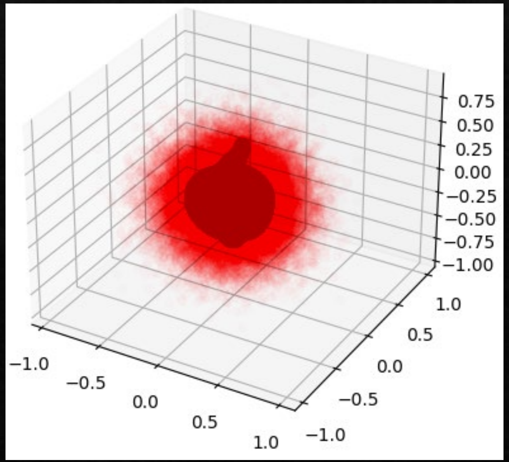
Column



Dragon_original



Serapis



Utah_teapot

Multiple LoD Dense Grid NeRF for Surface Reconstruction

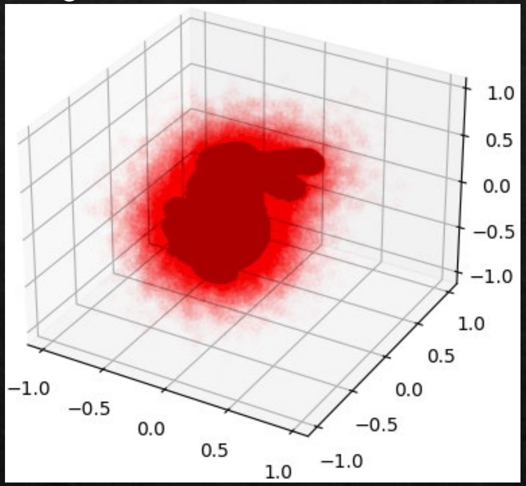
Quantitative Results: Reconstruction Error

	Bunny	Column	Dragon_original	Serapis	Utah_teapot
Chamfer	0.0019	0.0029	0.0021	0.0021	0.0020
Hausdroff	0.0093	0.0211	0.0306	0.0195	0.0265

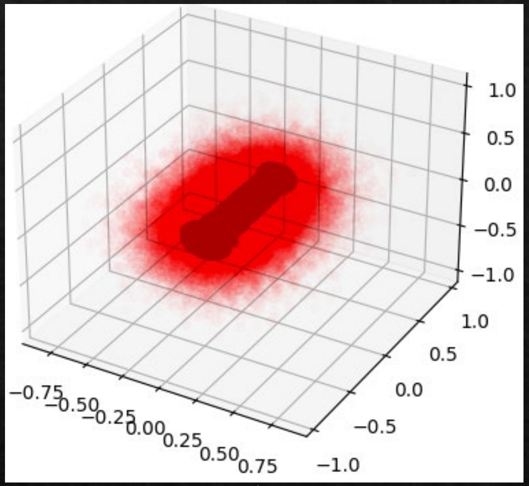
Implementation Choices:
Grid: Resolution: [8, 16,32, 64, 128, 256] (6 Levels)
MLP Structure: Input feature dimension: 32; Hidden layer feature dimension: 128; Number of hidden layers: 1
Regularization: Number of sampling points: 10000; Shifting scale: 0.15; Regularization Weight: 0.02
Training Termination Criteria: loss < 0.02, or loss < 0.05 and change of loss < 0.00001

Hash Grid NeRF for Surface Reconstruction

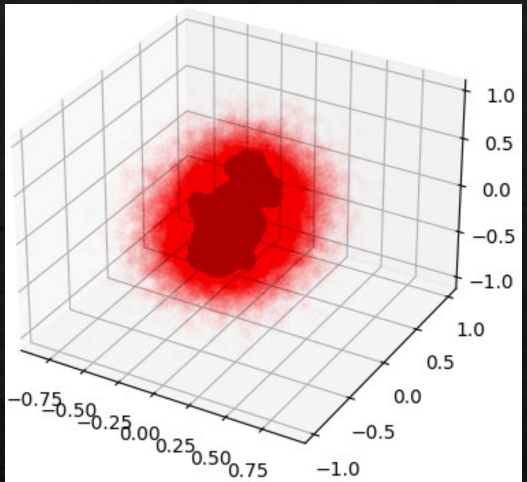
Qualitative Results



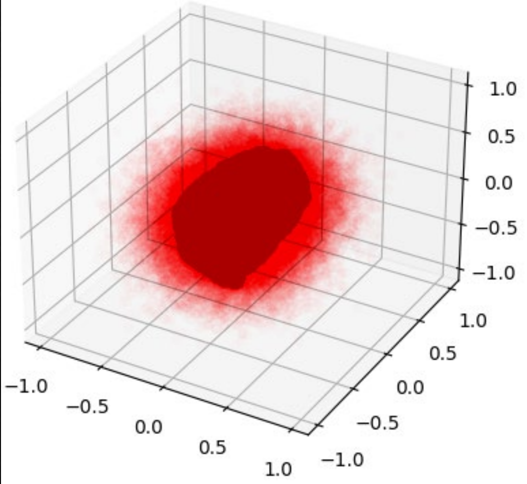
Bunny



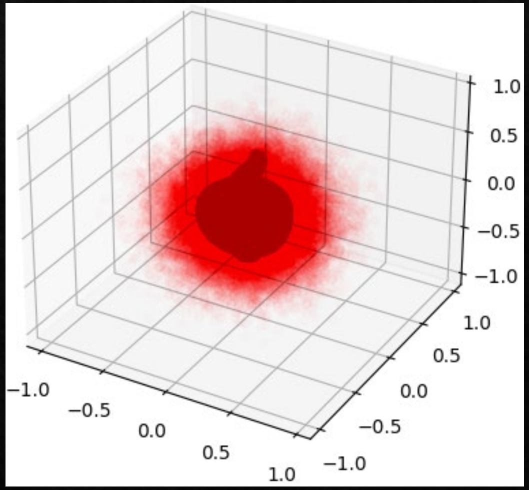
Column



Dragon_original



Serapis



Utah_teapot

Hash Grid NeRF for Surface Reconstruction

Quantitative Results: Reconstruction Error

	Bunny	Column	Dragon_original	Serapis	Utah_teapot
Chamfer	0.0023	0.0028	0.0023	0.0024	0.0024
Hausdroff	0.0432	0.0386	0.0707	0.0614	00363

Implementation Choices:
Grid: Resolution: [7, 8, 11, 14, 18, 23, 30, 38, 50, 64] (10 Levels), Bandwidth: 19
MLP Structure: Input feature dimension: 16; Hidden layer feature dimension: 128; Number of hidden layers: 2
Regularization: Number of sampling points: 10000; Shifting scale: 0.1; Regularization Weight: 0.02
Training Termination Criteria: loss < 0.01, or loss < 0.05 and change of loss < 0.00001

Comparison: Single LoD VS Multiple LoD

- Codebook Size:
  - Single LoD: $256^3 \times 3 = 50{,}331{,}648$
  - Multiple LoD: $(8^3 + 16^3 + 32^3 + 64^3 + 128^3 + 256^3) \times 32 = 613{,}564{,}416$
- Model Size (this is a rough estimation, as we disregard the bias terms):
  - Single LoD: $3 \times 64 + 64 \times 64 + 64 \times 64 + 64 \times 64 = 12{,}480$
  - Multiple LoD: $32 \times 128 + 128 \times 128 = 20{,}480$
- Speed (Training Time):

	Bunny	Column	Dragon	Serapis	Teapot
Single LoD	4s	10s	4s	6s	3s
Multiple LoD	47s	3min 45s	1min 34s	49s	1min 26s

- Quality (Reconstruction Error - Chamfer):

	Bunny	Column	Dragon	Serapis	Teapot
Single LoD	0.0225	0.0097	0.0104	0.0312	0.0275
Multiple LoD	0.0019	0.0029	0.0021	0.0021	0.0020

Conclusion

- The trade-off between Single LoD and Multiple LoD Regular Grids is that:
- Single LoD grids requires fewer storage space for the codebook and the network
- Network with single LoD grids is faster to train
- Network with multiple LoD grids generates reconstruction of better quality.
- The trade-off is: Storage and Speed VS Quality

Comparison: Regular Grid VS Hash Grid

- Codebook Size:
  - Single LoD: $256^3 \times 3 = 50{,}331{,}648$
  - Multiple LoD: $(8^3 + 16^3 + 32^3 + 64^3 + 128^3 + 256^3) \times 32 = 613{,}564{,}416$
  - Hash Grid: $2^{19} \times 16 = 8{,}388{,}608$
- Model Size:
  - Single LoD: $3 \times 64 + 64 \times 64 + 64 \times 64 + 64 \times 64 = 12{,}480$
  - Multiple LoD: $32 \times 128 + 128 \times 128 = 20{,}480$
  - Hash Grid: $16 \times 128 + 128 \times 128 + 128 \times 128 = 34{,}816$
- Speed (Training Time):

	Bunny	Column	Dragon	Serapis	Teapot
Single LoD	4s	10s	4s	6s	3s
Multiple LoD	47s	3min 45s	1min 34s	49s	1min 26s
Hash Grid	1min 51s	5min 53s	3min 9s	1min 43s	2min 59s

- Quality (Reconstruction Error - Chamfer)

	Bunny	Column	Dragon	Serapis	Teapot
Single LoD	0.0225	0.0097	0.0104	0.0312	0.0275
Multiple LoD	0.0019	0.0029	0.0021	0.0021	0.0020
Hash Grid	0.0023	0.0028	0.0023	0.0024	0.0024

Conclusion

- Hash grids require much fewer storage space for the codebook. Although there are more network parameters, the overall storage burden for hash grids is much fewer than regular grids;
- Network with hash grids is much slow to train;
- Network with hash grids generates far better reconstructions than single LoD regular grids. The reconstruction quality is slightly inferior to multiple LoD regular grids.

Comparison: Resolution at Reconstruction Stage

Resolution = 256

	Bunny	Column	Dragon_original	Serapis	Utah_teapot
Chamfer	0.0014	0.0014	0.0013	0.0016	0.0011
Hausdorff	0.0087	0.0110	0.0271	0.0155	0.0113

Resolution = 128

	Bunny	Column	Dragon_original	Serapis	Utah_teapot
Chamfer	0.0019	0.0029	0.0021	0.0021	0.0020
Hausdorff	0.0093	0.0211	0.0306	0.0195	0.0265

Resolution = 64

	Bunny	Column	Dragon_original	Serapis	Utah_teapot
Chamfer	0.0044	0.0062	0.0048	0.0044	0.0040
Hausdorff	0.0257	0.0316	0.0707	0.0322	0.0325

Resolution = 32

	Bunny	Column	Dragon_original	Serapis	Utah_teapot
Chamfer	0.0103	0.0109	0.0119	0.0105	0.0115
Hausdorff	0.0572	0.0422	0.1589	0.0599	0.0771

Conclusion: Lower resolution at reconstruction stage leads to worsened reconstruction quality.