Project 14

Motion Compensation and Evaluation of 3D Head Reconstruction

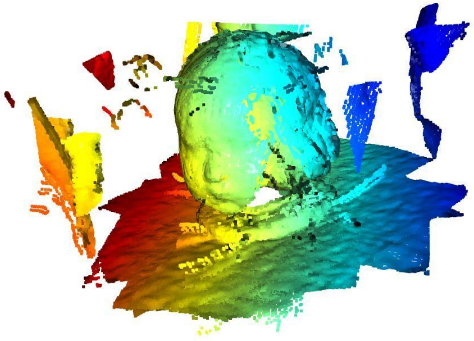
PediaMetrix Inc.

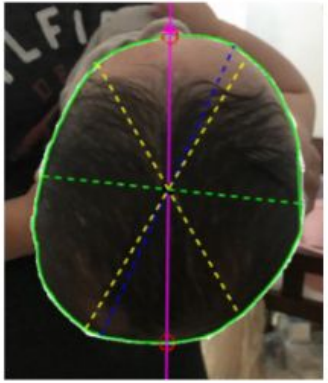
Background Reading Presentation

Student: Unnat Antani
Mentors: Can Kocabalkanli , Ozgur Guler, Reza Seifabadi

A brief review and goal

- Our major aim to be able to capture the head deformities on baby heads.
- To be able to reconstruct a 3d model of the baby head taking into account the distortions due to motion



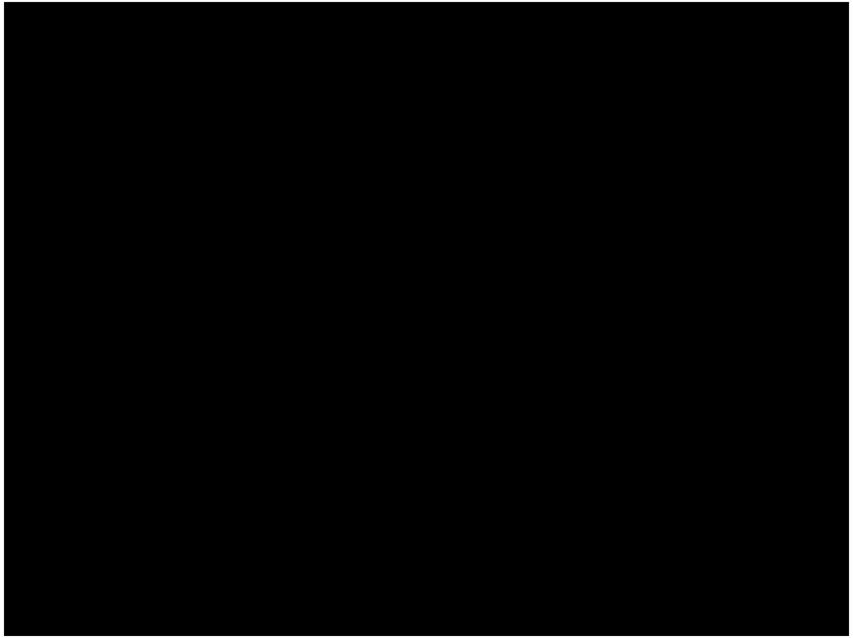


“Instant Neural Graphics Primitives with a Multiresolution Hash Encoding”

- The authors of the paper are Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller.
- It was published on arXiv.org
- The implementation and performance was presented in the GTC conference on March 2022 held by NVIDIA.
- **Summary**: Using hash tables and neural networks for faster and better reconstruction of 3D environment using 2D data.
- **Relevance**:
  - Gets rid of 3D sensor and related costs
  - Works directly on 2D images
  - Does not require pre-training i.e., is task agnostic
  - Can learn additional features like distortions, extrinsics
  - Takes about 30 seconds to generate 3D scene

Problem Statement

- To learn the primitives of images\scenes using a neural network
- Optimizing the task by using hash table for encoding the images\scenes
- To quickly train and reconstruct scenes with vast amount of detail or to reconstruct a 3D environment from 2D images
- Testing on 4 types of reconstruction:
  - Gigapixel
  - Radiance Field
  - Neural Radiance and Density Field
  - Neural Volume



Previous Work

- Similar approach of encoding used in Machine Learning in One-Hot encoding
- Used in RNN and Transformers recently where encoding helps the neural network identify what part it is processing.
- Encoding has been used to reconstruct five dimensions of the light field. This was the first implementation of NeRF in 2020 by Mildenhall et. al.
- Parametrizing the training features in the form of grids, and trees help the performance of neural network by method of interpolation.
- Trade off between size of such encoding and fully connected MLPs, one can have small neural networks that converge faster.

Technical Approach

- A completely Deep Learning based approach for this would mean that we would need to over-fit our neural network on the scene\images we provided.
- The weights learnt over time for this neural network would represent the scene. This would mean that input of any other angle or view would give the output of approximately how the scene would look like from that unknown angle.

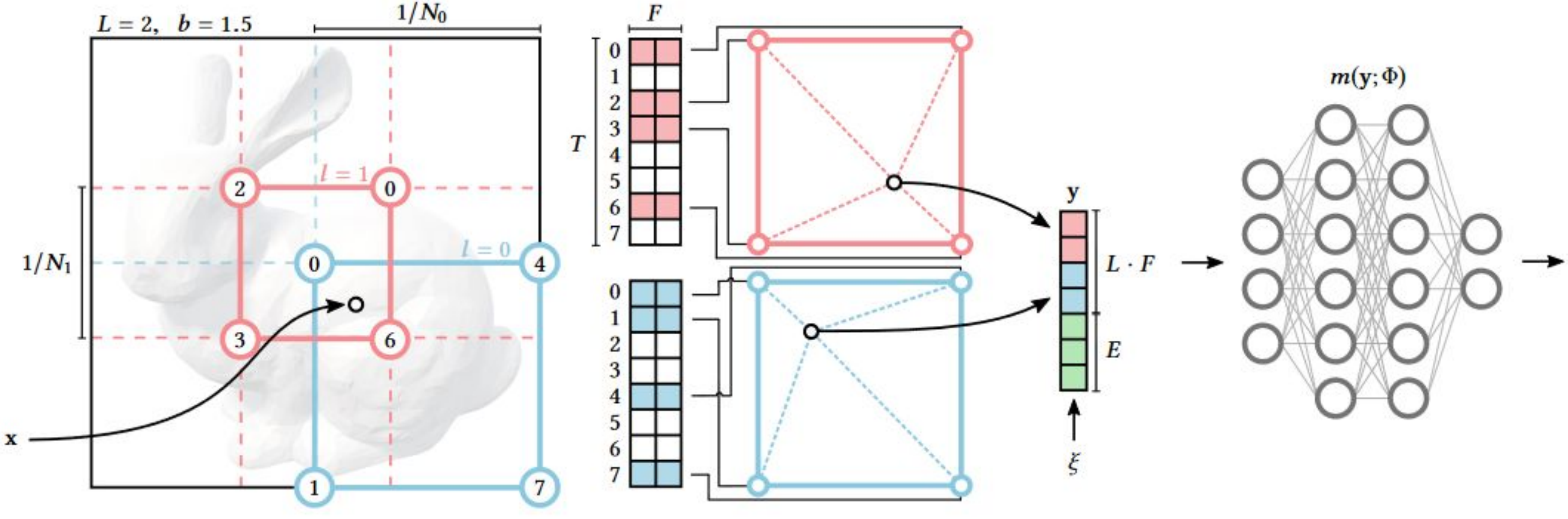
Technical Approach(contd.)

- With the encoding method, we store a few parameters in a hash table, thereby reducing the number of features to learn.
- Certain features are interpolated using the hash-table, thereby reducing the time to learn a new scene as well as the computation needed.
- The grids are assigned an index, both on coarse and fine level and the clashing of index is taken care of by the gradient based approach(avg).
- It uses the power of GPU to decrease the time of computation as well.

Multiresolution Hash Encoding

- Given a fully connected neural network $m(y; \Phi)$, we are interested in an encoding of its inputs $y = enc(x; \theta)$ that improves the approximation quality and training speed
- Our neural network not only has trainable weight parameters Φ, but also trainable encoding parameters θ
- These are arranged into L levels, each containing up to T feature vectors with certain dimensionality
- Each level stores feature vectors at the vertices of a grid, the resolution of which is chosen to be a geometric progression between the coarsest and finest resolutions

Multiresolution Hash Encoding(contd.)



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

Implementation

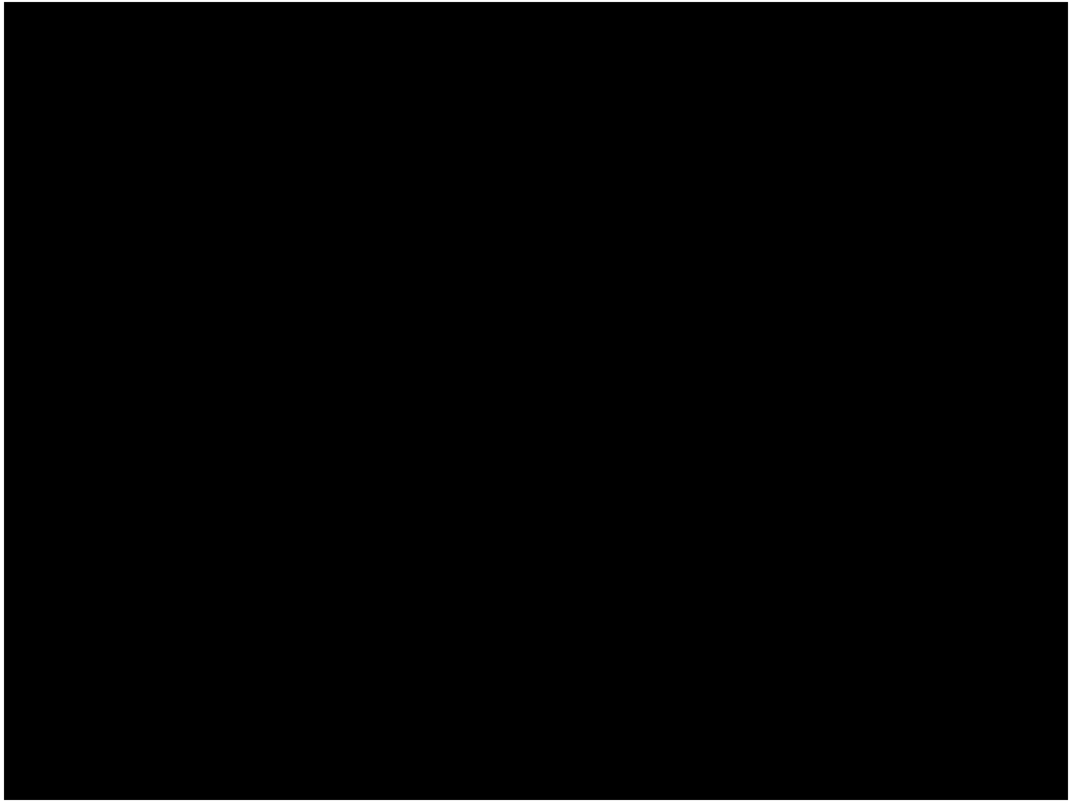
- Implementation is done with CUDA using the Nvidia's tiny-cuda-nn framework.
- For NeRF, the architecture of the Neural Network is as follows:
  - 2 concatenated MLPs each with 2 hidden layers and width of 64 neurons:
    - A density MLP
    - Color MLP
  - The density MLP maps the hash encoded position $y = enc(x; \theta)$ to 16 output values the first of which we treat as log-space density.
  - The input of color MLP are the 16 coefficients of harmonic basis function which determines the projection and the output of density MLP
- The final output is an RGB triplet for each position

Implementation(contd.)

- So a ray is simulated from the data learnt and the density network classifies whether in the path of the ray does it encounter an obstruction
- If it does, then the RGB network outputs an RGB triplet at that position to render.

	Mic	Ficus	Chair	Hotdog	Materials	Drums	Ship	Lego	avg.
Ours: Hash (1 s)	26.09	21.30	21.55	21.63	22.07	17.76	20.38	18.83	21.202
Ours: Hash (5 s)	32.60	30.35	30.77	33.42	26.60	23.84	26.38	30.13	29.261
Ours: Hash (15 s)	34.76	32.26	32.95	35.56	28.25	25.23	28.56	33.68	31.407
Ours: Hash (1 min)	35.92 ●	33.05 ●	34.34 ●	36.78	29.33	25.82 ●	30.20 ●	35.63 ●	32.635 ●
Ours: Hash (5 min)	36.22 ●	33.51 ●	35.00 ●	37.40 ●	29.78 ●	26.02 ●	31.10 ●	36.39 ●	33.176 ●
mip-NeRF (~hours)	38.04 ●	33.19 ●	37.14 ●	39.31 ●	32.56 ●	27.02 ●	33.08 ●	35.74 ●	34.510 ●
NSVF (~hours)	34.27	31.23	33.19	37.14 ●	32.68 ●	25.18	27.93	32.29	31.739
NeRF (~hours)	32.91	30.13	33.00	36.18	29.62	25.01	28.65	32.54	31.005
Ours: Frequency (5 min)	31.89	28.74	31.02	34.86	28.93	24.18	28.06	32.77	30.056
Ours: Frequency (1 min)	26.62	24.72	28.51	32.61	26.36	21.33	24.32	28.88	26.669

Results



Pros

- Significance of this is that it is incredibly fast on any given data
- It is task agnostic and learns on the fly
- Possible to reconstruct 3D model in real time
- Only required 2D images and obviates use of 3D sensors
- Potential for improvement by improving the hashing method

Cons

- Relies on GPU
- Data prepping requires use of another open-source software with its own bottleneck
- The available marching cubes algorithm doesn’t give great 3D models

References

[1] https://nvlabs.github.io/instant-ngp/

[2] Muller, Thomas and Evans, Alex and Schied, Christoph and Keller, Alexander, “Instant Neural Graphics Primitives with a Multiresolution Hash Encoding”, 2022, arXiv

[3] https://blogs.nvidia.com/blog/2022/03/25/instant-nerf-research-3d-ai/

Thank you !