Machine Learning in Computer Vision

(3D Reconstruction of Data from 2D Images)

Name: Abdul Basit

Student N': 20246646

M1-Int track in Electrical Engineering

Submitted to:

Professor Lydie Nouveliere

Contents

Table of Contents

Introduction:

1.1 Overview of 3D Reconstruction

The process of creating 3-D model of scene or an object is known as 3-D reconstruction of data. 3-D reconstruction of scene or object can be done having input data as 2-D images from different view point, video, point clouds or LiDAR etc. Traditionally, 3-D reconstruction has been done through Active or Passive methods. Active methods mostly relies on actively interacting with the object or scene by projecting some form of energy onto it and sensing how that energy has been modified such as depth cameras, laser scanners that beam and infrared light to capture depth information in real time. These methods can be costly, difficult to calibrate sensing system, and with limited range. Alternatively, passive methods do not actively interact with the object. They use the light that is already present in the object or scene such as using 2-D images of the scene.

1.2 Role of Machine Learning in 3D Reconstruction

Machine learning is playing a crucial role in revolutionizing 3-D reconstruction. In past identifying features (corner, shapes, texture) across multiple images was a time consuming task but now machine learning especially deep neural networks can automatically identify and match features even in the complex scene like view dependent images etc. Machine learning models can also be trained to handle noise in the data. They can fill the gaps, reconstruct more accurate and complete 3-D scenes. Unlike traditional methods, Machine learning models can handle unfamiliar data and scenes more accurately.

1.3 Scope and structure of this paper

There are different methods for 3-D reconstruction of data from different types of input data such as images, point clouds, depth sensors etc. But in this paper we will be focusing on 3-D reconstruction of scene or an object from 2-D images (passive method) taken from different viewing directions.

In this case the availability of the data will not be an issue, compared to other methods like getting data from depth sensors like (LiDAR etc) this is more affordable, and also image capture the color or texture of an image which can be directly mapped onto the 3-D model resulting in visually appealing and more realistic results.

On the other this processing the images and reconstructing a 3-D model from them can be computationally very intensive especially for high resolution images or large datasets, the accuracy of 3-D model will be dependent on the number of images, number of overlapping images, quality of images and lightning conditions. Changing light can be very sensitive in some cases.

3-D Reconstruction Methodologies

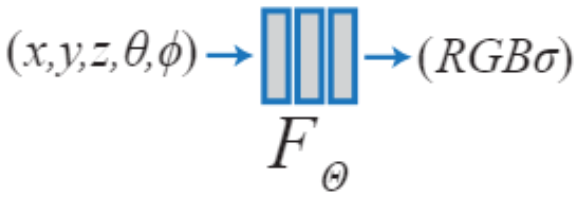
Neural Radiance Fields (NeRF)

2.1 Introduction

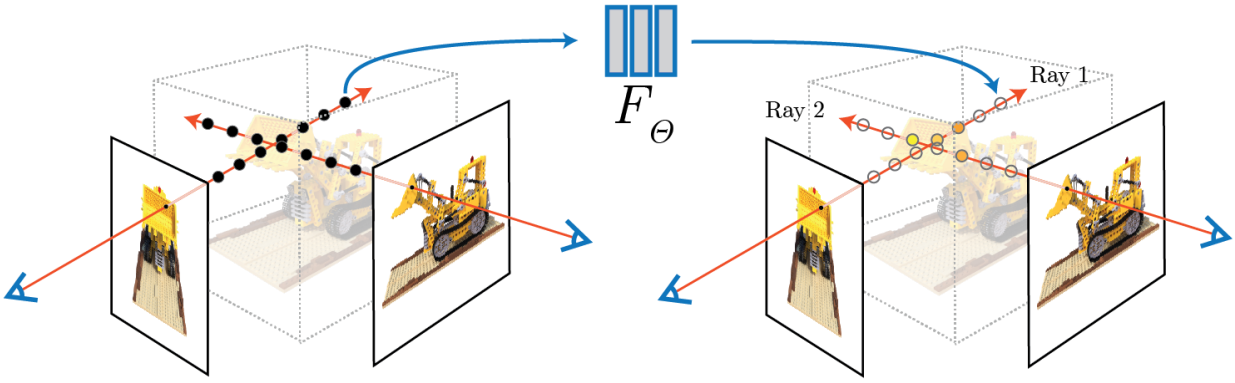
Neural Radiance Fields also known as NeRF is research project completed with the collaboration of researchers from Google and University of California, Berkley. In the name NeRF, neural represents a neural network that learns how light behaves in a 3-D scene by mapping spatial positions and viewing direction into color and density, radiance means how much light is emitted by a point in space in each direction and field represents a continuous function over time.

2.2 Architecture

NeRF has very simple architecture as simple as multilayer perceptron. NeRF as an input takes a 5-Dimensional vector having spatial location (x, y, z) and viewing direction (θ, Φ) and outputs a color (r, g, b), meaning how much light is emitted by radiance and density (σ), where density represent is there any object in that point or not, if there is then how dense it is, 0 if there is nothing and infinity for an object. NeRF have 9 Fully-connected non-convolutional layers each having 256 channels. So weights are going to define the subset of whole world of NeRF, we will optimize the weight on the given images to explain the way we see the world.



We generate views using **volume rendering**, which is a way to create 2D image that shows what this 3-D object would look like if you look at it from certain angle. Through volume rendering we know how light interacts with the material inside the space. It actually works like a ray tracer. We shoot rays at each pixel from camera and drops points evenly along the ray and each of that point (spatial location and viewing direction) is an input vector to the neural network and get output as an emitted radiance and density.



So, now we take the ray and march from back to front or front to back until we reach back to camera. While marching back we alpha-compose (If something is opaque it will get rid of the stuff behind it, it actually the modeling of how semi-transparent stuff interact with each other in physical way) densities on top of each other and at the end we get radiance. The color of pixel is weighted combination of color in the world and alpha-values

$$C \approx \sum_{i=1}^{N} T_i \alpha_i c_i$$

weights

colors

The one issue is that we have high radiance fields defined by neural network and it is hard to drop sample along all the rays to get good rendering, So we do it by evenly dropping point along the rays which give us the information about where the stuff is and then we lay down more points based on the information from point dropped before. In this case the (Weight and alpha) acts as probability distribution for new sample points. This is known as Two-pass rendering.

The way we train this model is that we iterate through pixels, we shoot a ray, we render that array according to our current randomly initialize neural network and then we apply gradient descent and minimize the Mean square error between render pixel value and the observed pixel value and through that we recover the 3-D scene.

2.3 Toy Problem

Another issue is that while learning a mapping from (x, y) which are the coordinates of an image to (rgb) values which are the color of pixel the neural network is unable to memorize the image and it is unable to reconstruct the image that it has fully observed even if you train it for days.

The solution to this that we take the coordinates (x, y, z) and just push them through bunch of sinusoids of increasing frequency and expand the feature vector to a high dimensional space with a lot of these high frequency sinusoids that are concatenated and then feedback into the neural network.

2.4 Conclusion

NeRF can generate realistic novel views with fine details and accurate lighting, even from a sparse set of input images. NeRF do not suffer from fixed resolution limitations, allowing for smooth scaling and rendering. NeRF model view-dependent effects like specular reflections and transparency, making them more physically accurate. But on the other hand NeRF is very data hungry; it cannot generalize well on limited data.

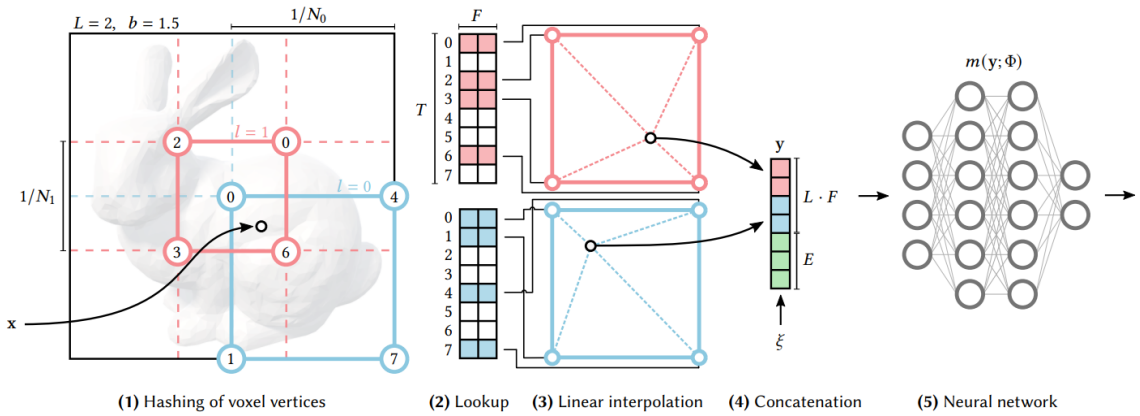
Instant Neural Graphics Primitives with Multi-resolution Hash Encoding

3.1 Introduction

Neural graphics primitives, parameterized by fully connected neural networks, can be computationally expensive to train and evaluate. This paper introduces a novel input encoding method that significantly reduces the computational cost without sacrificing quality. The approach integrates a small neural network with a multiresolution hash table of trainable feature vectors, optimized through stochastic gradient descent. This structure helps resolve hash collisions and enables efficient parallelization on modern GPUs. By implementing fully-fused CUDA kernels, the system minimizes memory bandwidth usage and computation overhead, resulting in a substantial speedup. The proposed method allows training high-quality neural graphics primitives in seconds and rendering in milliseconds at high resolutions.

3.2 Architecture

The technique mentioned in paper [1] for encoding is not hardware friendly because cosine and sine are complex math operation and it also does not able to reflect sharp details. The MLP used in [1] is large that makes it computationally expensive. To overcome all these issues the researchers at NVIDIA proposed this paper. They used Hash grid encoding where trainable features vectors are stored in multi-resolution hash table.



The authors added grid structure to 16 different levels of resolutions. In this technique we only look at grid that contains X and hash its vertices that contain input X. Hash function take the vertices (the corner locations) i.e 1,4,0,6. We look-up to corresponding learnable features vectors from hash table and linear interpolate to get the actual feature while considering only the grids that contains X. Then we concatenate all the feature of input X at different resolutions while also concatenating the parameters. After that these features are passed to small fully connected Neural Network. The NN predicts the density and RGB color at given 3-D point from the given viewing direction. These outputs are then used for volume rendering similar to NeRF [1]. The authors used fully fused cuda kernels where data does not leavae GPU until final output.

3.3 Conclusion

Instant NeRF presents a significant breakthrough in neural scene representation by drastically improving training and rendering efficiency. By replacing large MLPs with a compact neural network augmented by a multiresolution hash grid encoding, it achieves high-quality results with significantly reduced computational cost. The use of fully-fused CUDA kernels enables optimal GPU parallelization, minimizing memory bandwidth waste and improving performance. As a result, Instant NeRF can train in seconds to minutes and render in real-time, making it highly suitable for interactive 3D applications

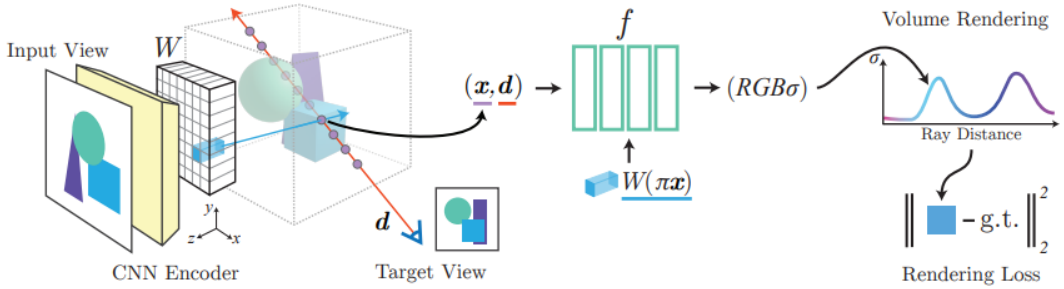
PixelNeRF: Neural Radiance Field from one or few images

4.1 Introduction

PixelNeRF is a neural rendering approach that improves upon standard NeRF by enabling single-image 3D reconstruction and novel view synthesis without requiring per-scene optimization. Unlike traditional NeRF, which needs multiple images and extensive training per scene, PixelNeRF learns a scene-independent prior from a large dataset of images. It extracts **pixel-**aligned image features using a convolutional neural network (CNN) and conditions the NeRF model on these features, allowing it to generate accurate 3D representations from just one or a few input images. This makes PixelNeRF significantly more data-efficient and generalizable.

4.2 Architecture

The techniques mentioned in paper [1] and [2] require Requires **multiple views** of a scene for high-quality results of 3-D reconstruction. They also do not utilize prior common knowledge across scenes. But pixel NeRF learn pixels level information from the data. The author used Convolutinal Neural Network is used as encoder and to extract image features that will be used to infer 3-D structure. The CNN-based encoder learns a **scene-independent prior** from a large dataset. Instead of training a new model for every scene, this prior enables PixelNeRF to generalize across scenes, allowing it to infer 3D information from a single or a few images without retraining.



The CNN architecture that authors used in ResNet34, that uses feature pyramid to get pixels aligned by up sampling it to the view space. The features extracted and 3-D coordinates are guided to the NeRF model that uses volume rendering for 3-D reconstruction of scenes. The reconstructed 3D scene is compared with ground truth, and the model is trained via gradient descent. Loss functions such as pixel-wise differences are used to optimize the model by ajusting the weights in both the CNN feature extractor and the NeRF model itself.

4.3 Conclusion

PixelNeRF offers a significant advancement in 3D scene reconstruction and novel view synthesis by enabling high-quality 3D reconstructions from a **single image** or a few input images, without the need for per-scene optimization. By leveraging **pixel-aligned features** extracted from 2D images using a CNN encoder, PixelNeRF generalizes across scenes and can infer 3D structures and viewpoints without extensive retraining. With its **scene-independent prior** and efficient inference process, PixelNeRF pushes the boundaries of single-image 3D reconstruction, offering a flexible and powerful tool for various real-time]

Neuralangelo: High-Fidelity Neural Surface Reconstruction

5.1 Introduction:

Neuralangelo is a neural surface reconstruction method that uses multi-resolution 3D hash grids and neural surface rendering to achieve high-fidelity 3D reconstructions from multi-view images. It employs numerical gradients for smoothing and a coarse-to-fine optimization strategy, enabling the reconstruction of intricate details from 2D video inputs. It samples 3-D points along camera viewing direction and use multiresolution hash encoding to encode the position. The encoded features are input to an SDF MLP and a color MLP.

5.2 Signed Distance Function:

It is an orthogonal distance of given point x to the boundary of set S, with sign determined by whether or not x is inside or outside of S. It is surface can be represented as zero-level set

$F(x)=S\{x\in R3|f(x)=0\}$

Given 3D point and SDF value, density value at x is calculated. This let us compute rendering loss. MLPs & feature stored in hash entries are trained jointly

$$L = L_{rgb} + L_{eik} + L_{curv}$$

- Lrgb: loss between input and synthesized image
- Leik: Eikonal loss regularize underlying SDF such that the surface normals are unit-norm.
- Curvature loss: regularize underlying SDF such that means curvature is not arbitrary large.

5.3 Scene Initialization:

SDF is initialized as approximately a sphere with this initial shape high curvature loss makes concave shapes difficult to form because it prevents topology. Thus instead of applying L-curv from beginning of optimisation process, a short warm up period is used that linearly increase the curvature loss weight.

5.4 Multi Resolution hash grids and Numerical Gradients

They are features into hash entries, and cell corner is a hash entry. Given input position xi, it maps to correspond grid resolution. Each grid resolution of the feature vector is obtained by linear interpolation of hash entries at corners. Feature vector is consisting of concatenated features from different resolution grids. Then feature vector is passed to shallow MLP. But analytical gradient is not good so authors proposed numerical gradient for continuous space. But it requires to compute 6 more points given xi along x,y,z axes. It allows to simultaneously update adjacent gradient cell given xi, thus become some version of analytical gradient. Use of numerical gradient distributes the back propagation update beyond the local hash grids. For optimization we do progressive activation of grids, finer hash grids are progressively activated when ϵ (step size) decrease to their spatial size. Neuralangelo employs numerical gradients instead of analytical ones to enhance surface reconstruction stability and detail preservation. Analytical gradients often introduce noise, instability, and artifacts, especially in fine details and higher-order derivatives. In contrast, numerical gradients act as a smoothing operation, refining surface details and ensuring more consistent optimization. This approach, combined with multi-resolution hash grids, allows Neuralangelo to achieve high-fidelity 3D reconstructions with improved accuracy and stability.

Summary

This paper explores advancements in 3D reconstruction from 2D images, focusing on neural radiance fields (NeRF) and their variations. It begins by outlining the shift from traditional, often expensive, methods to machine learning-driven approaches, which offer improved accuracy and efficiency through automated feature identification and noise handling.

The core of the paper delves into four key methodologies: NeRF, Instant NeRF, PixelNeRF, and Neuralangelo. NeRF, the foundational method, utilizes a neural network to map spatial positions and viewing directions to color and density, using volume rendering for 3D object representation. While effective in generating detailed novel views, NeRF is computationally intensive and data-dependent.

Instant NeRF addresses these limitations by introducing a multiresolution hash table encoding, significantly reducing computational costs and enabling real-time rendering. This approach replaces complex mathematical operations with efficient hash grid lookups and linear interpolation, enhancing hardware compatibility and detail representation.

PixelNeRF further advances the field by enabling 3D reconstruction from single or few images, leveraging a CNN-based encoder to extract pixel-aligned features. This scene-independent learning approach allows for generalization across datasets, overcoming the data limitations of traditional NeRF.

Finally, Neuralangelo focuses on high-fidelity surface reconstruction using multi-resolution hash grids and numerical gradients for smoothing. It employs a signed distance function and coarse-to-fine optimization, enabling the capture of intricate details from multi-view images.

Collectively, these methodologies represent significant strides in 3D reconstruction, balancing computational efficiency with high-quality output. They underscore the transformative role of machine learning in processing 2D images to generate detailed and accurate 3D models, with applications ranging from interactive 3D environments to single-image reconstruction.

Bibliography

[1] Jonathan Mildenhall, Pratul P. Srinivasan, Mariusz Klain, et al. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. pp. 6346-6355.

[2] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics*, 41(4):1-12, 2022.

[3] Xiuming Zhang, Liangyu Zhao, Yijun Li, et al. PixelNeRF: Neural 3D Reconstruction with Pixel-Aligned Implicit Functions. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. pp. 2616-2625.

[4] Chen-Hsuan Lin, Jun Gao, Luming Tang, Xiuming Zhang, Zhiding Yu, Sergey Tulyakov, Shalini De Mello, Sanja Fidler. Neuralangelo: High-Fidelity Neural Surface Reconstruction. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.