INFO8010: Reading assignment

Julien Gustin,[1] **Lucas Michel**,[2] and **Joachim Houyon**[3]
[1] *julien.gustin@student.uliege.be (s180337)*
[2] *lucas.michel@student.uliege.be (s170492)*
[3] *joachim.houyon@student.uliege.be (s181539)*

This report consists of a summary of the problem that is tackled by the paper [1] named *"NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis"* from Mildenhall et al, 2020. We explain why such a problem is challenging from the perspective of the wider research context. We conclude this report by a discussion of the advantages and shortcomings of the contributions of the paper.

I. VIEW SYNTHESIS PROBLEM

The article [1] presents a method to address the long-standing problem of view synthesis tackled by a new approach. View synthesis aims to create any new kind of view from any direction, given a set of input images along with its known camera poses, of a given scene.

A. Related works and wider research context

In the computer vision community, there exist two main ways to tackle the view synthesis problem. Here is a short overview of both two lines.

Image-based rendering Given a dense sampling of different views of a scene, one can construct original photorealistic views of this same scene (ref 21,5,7 Nerf). Such solutions are now quite old in the computer vision community. When it comes to a sparser sampling of different views of the same scene, there exist two popular classes of approaches. The first approach uses a mesh-based representation of scenes thanks to methods such as *differentiable rasterizer* or *path tracers*, one can optimize mesh representations using gradient descent in order to reproduce the scene associated to the input image. In this approach, the user needs to fix the topology, which is usually unavailable for real problems, and the gradient-based mesh optimization is usually difficult because of the local minima. The second approach uses volumetric representation; with this approach, impressive results are obtained. However, because of the discrete sampling, their ability to scale to higher resolution is limited because of poor time and space complexity.

Neural scene representation The idea is to encode a scene in the weights of a multilayer perceptron. For example, (ref 11,27 Nerf) has constructed neural networks to simulate an occupancy field (ie: map 3D coordinates of the scene to 0 or 1 whether there is an object or not in that location). Second example, (ref 42 Nerf) has built a neural network mapping every location to an RGB color together with a rendering function which tells us, along each ray, where the first incident object is located. Such methods work well only on simple scenes with low geometric complexity.

II. NERF

Representing Scenes as Neural Radiance Fields for View Synthesis (NeRF) achieves state of the art results by representing a scene using a fully-connected deep neural network. NeRF tries to solve the weaknesses of the previous techniques presented in the section I A. Given a viewing direction $\boldsymbol{d} = (\theta, \phi)$ and a spacial location $\boldsymbol{x} = (x, y, z)$, the neural network outputs an emitted color $\boldsymbol{c} = (r, g, b)$ and a volume density/opacity σ. While the emitted colour depends among other things on the direction of observation, the density only depends on the location of the voxel.

The optimisation process works as follow; a single neural network is (over)-fit to one particular scene given images of that scene from different positions and viewing directions. Then, synthetic 2D frames are compared to the ground truth by minimizing the error between these two, across multiple views using gradient descent. To get a synthetic image, one needs to perform:

1. Each pixel of that image sends a ray through the scene
2. Query the MLP at each location, where the ray traverses the scene with the viewing direction to get a vector of colors and densities
3. Use classical volume rendering algorithm to render these colors and the density into a pixel

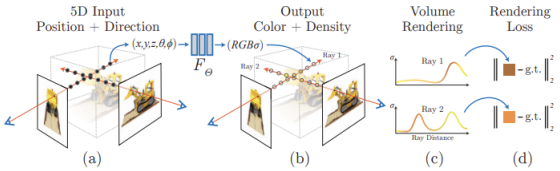
However, to make it work, one needs several tricks.

A. Contributions

The method proposed in this paper comes with several technical contributions. These contributions allowed their neural radiance field to outperform state-of-the-art view synthesis methods.

1. *Rendering continuous scenes with complex geometry:*

A scene representation is approximated using an MLP $F_\theta : (\boldsymbol{x}, \boldsymbol{d}) \rightarrow (\boldsymbol{c}, \sigma)$ where the weights θ are optimized to map each inputs to the corresponding volume density and emitted color. To enforce the representation to be multi-



view consistent, the network is constrained to predict the volume density σ with respect to the location $\boldsymbol{x}$. The RGB color $\boldsymbol{c}$ is predicted using $\boldsymbol{x}$ and $\boldsymbol{d}$.

2. *Volume rendered techniques used to construct a differentiable rendering procedure:*

The volume density $\sigma(x, y, z)$ can be seen as the differential probability of a ray that terminates at an infinitesimal particle at position (x, y, z). The approximation of the expected color $C(\mathbf{r})$ of camera ray $\mathbf{r}(t)$, which is computed from an origin $\mathbf{o}$ and a direction $\mathbf{d}$, within the bound $[t_n, t_f]$ is done by using quadrature with a stratified sampling approach.

$$C(\boldsymbol{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt$$

where

$$T(t) = exp\left(-\int_{t_n}^{t} \sigma(\mathbf{r}(s))ds\right)$$

is the probability that the ray doesn't hit any other particle in a range from t_n to t, also denoted as the accumulated transmittance along the ray.

3. *Hierarchical sampling*

In order to enhance the efficacity of the rendering strategy, samples are allocated proportionally with respect to their expected effect on the final rendering. This is done using two networks; a "coarse" network which is evaluated by sampling N_c locations using stratified sampling, this network will act as a piecewise-constant PDF which will be used to sample N_f additional samples using inverse transform sampling. A "fine" network is evaluated at the union of the first and second set of samples which gives the final rendered color of the ray.

4. *Effectively optimize neural radiance fields using positional encoding:*

The representation of high-frequency variation in color and geometry is not efficient when the network works with the input $xyz\theta\sigma$. Instead, the network is composed along with a function $\lambda : p \in \mathbb{R} \rightarrow \mathbb{R}^{2L}$, L being a hyperparameter, that is not learned.

$$\lambda(p) = (\sin(2^0\pi p), \cos(2^0\pi p), \ldots, \sin(2^{L-1}\pi p), \cos(2^{L-1}\pi p))$$

This function allows the MLP to approximate a higher frequency function with less difficulty by appyling λ separately to each components of $\boldsymbol{x}$ and $\boldsymbol{d}$.

B. Critical discussion of the advantages and shortcomings of the contributions

First, the abstract of the paper explains quite well its content, without going into too much details, and does not oversell its content. It is also important to note that this paper comes with the source code, which is convenient since the experiments are reproducible, allowing the reader to re-implement the paper.

Second, the authors provide an ablation study to validate their design choices, which is interesting. However, the choice of restricting the network to predict the volume density only by $\boldsymbol{x}$, while the RBG color is predicted using $\boldsymbol{x}$ and $\boldsymbol{d}$, seems more to be taken by intuition and from an arbitrary choice than by empirical experiments or theoretical proofs. Adding this experiment to the ablation study would be more scientific and robust.

Finally, concerning the model itself, as the paper states it, NeRF is very time consuming. Indeed, the optimisation for a single scene takes, according to the paper, around one or two days on a NVIDIA V100 GPU. This restricts the use of NeRF to non real-time problems while *old* methods are quicker but give less good results. Since then, new advances on the domain has emerged, such as Instant NeRF by Nvidia [2] that takes only few seconds to train and miliseconds to render.

[1] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *CoRR*, abs/2003.08934, 2020.

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