A Radiant Optical Porosity Model: A New Perspective on Neural Radiance Fields

Timothy Chen

***Abstract*—We introduce a new volumetric rendering scene representation based on volume density of materials, and hence reasons about the porosity of materials. We explicitly derive a volume rendering equation from this volume density model based on first principles. Importantly, we show that this rendering equation can be made identical to that of Neural Radiance Fields (NeRFs), hence equating the expressivity of this model to that of NeRFs. Additionally, we show some desirable properties of this representation. Instead of a typical L2 reconstruction loss, we formulate regression of the scene parameters as a Maximum Likelihood Estimation problem to prioritize creation of surfaces. We show a globally optimal solution exists for a class of scene parametrizations. We then evaluate several approaches to regressing these scene parameters.**

***Index Terms*—Volume Rendering, NeRFs**

I. Introduction

In this work, we derive an optical model to address the problem of novel view synthesis, while shedding insight onto the recent development that is Neural Radiance Fields (NeRFs). To perform this task, we seek to model the environment volumetrically through the association of points in space to a volume density. In other words, we are asking the question "How porous is an infinitesimal region around point x" and addressing the question of whether we are able to render images from arbitrary poses using this representation. Because this work is inspired by the gap in understanding surrounding NeRFs and the optical model it uses, we will spend some time discussing the underlying optical model of NeRFs. This discussion is also important to draw parallels and lay out the notation for this work. We then present our optical model, the associated rendering equation, and properties of this equation and how it relates to NeRFs. The goal is to empirically test the analysis presented here by comparing the performance (computation time and reconstruction quality) of our model and to NeRFs. We further provide an ablation on our model over the typical MSE loss and a well-defined Maximum Likelihood loss.

II. Neural Radiance Fields

A. Preliminaries

In this section, we introduce the mathematical preliminaries and notation used to describe NeRFs and our work. For clarity, we use bold face for vector variables and functions that output vectors, and non-bold text for scalar variables, functions that output scalars.

A NeRF is a pair of functions $(\sigma(\mathbf{p}), c(\mathbf{p}, \mathbf{d}))$. The density function, $\sigma : \mathbb{R}^3 \mapsto [0, 1]$, maps a 3D location $\mathbf{p} = (x, y, z)$ to a non-negative scalar value σ that encodes the differential probability that a point is occupied. The radiance (i.e., RGB color) function $\mathbf{c} : \mathbb{R}^3 \times \mathbb{R}^2 \mapsto \mathbb{R}^3$ maps a 3D location $\mathbf{p} = (x, y, z)$ and camera view direction $\mathbf{d} \in \{\mathbf{x} \in \mathbb{R}^3 \mid ||\mathbf{x}|| = 1\}$ (alternatively parameterized as a 2D vector of angles (θ, ϕ)) to an emitted RGB color $\mathbf{c}$ represented as a vector in $\mathbb{R}^3$. Ideally, regions of space that the representation is confident in being occupied (i.e. a solid surface) has $\sigma = \infty$, while free space is characterized by $\sigma = 0$. For this work, we restrict our focus on modelling σ, since this is what defines the geometry. This function can be modelled regardless of the form of c (e.g. Multi-layer Perceptron, spherical harmonics).

We define $\mathbf{C}(\mathbf{o}, \mathbf{d}) \in [0, 1]^3$ as the rendered pixel color in an image when taking the expected color value from the NeRF along a ray $\mathbf{r}(t; \mathbf{o}, \mathbf{d})$ with camera origin $\mathbf{o}$ and pixel orientation $\mathbf{d}$, where $\mathbf{r}(t) = \mathbf{o} + t \cdot \mathbf{d}$. Specifically, the rendering equation for a pixel is given by:

$$\mathbf{C}(\mathbf{r}; \mathbf{d}) = \int_{t_n}^{t_f} \sigma(\mathbf{r}(t)) \mathbf{c}(\mathbf{r}(t); \mathbf{d}) \exp\left[-\int_{t_n}^{t} \sigma(\mathbf{r}(\tau))\, d\tau\right] dt. \tag{1}$$

The photometric loss used to regress σ and $\mathbf{c}$ is simply the L2 norm between the rendered pixel $\mathbf{C}$ and the true pixel from the training image $\bar{\mathbf{C}}$. Vectors $\mathbf{o}$ and $\mathbf{d}$ for every pixel can be derived from the pose associated with the training image.

However, computing the integrals in Equation 2 is intractable. Instead, we use a piece-wise linear assumption on σ along sample points of t, while assuming piece-wise constant c in these intervals. We can then approximate (1) by the sum

$$\begin{aligned} \mathbf{C}(\mathbf{r}; \mathbf{d}) &= \sum_{i=0}^{N-1} \mathbf{c}(\mathbf{r}(t_i); \mathbf{d}) T_i \\ T_i &= \exp\left[-\sum_{j=0}^{i} \sigma(\mathbf{r}(t_j))\delta_j\right] - \exp\left[-\sum_{j=0}^{i+1} \sigma(\mathbf{r}(t_j))\delta_j\right]. \end{aligned} \tag{2}$$

B. Probabilistic Volumetric Model

In this section, we reference prior literature to illustrate the conventional view that NeRFs are a probabilistic quantity. The works we reference are [1], [2].

Williams and Max [1] rely on a particle model to derive (2). Namely, they assume spherical particles occluding a cylinder drawn out from the pixel area. The number of occluding particles within a cylinder extending from the pixel to some distance t away, along the optical ray, is assumed to be a

Poisson-distributed random variable. The contribution of each slice of the cylinder to the pixel follows a distribution that is the derivative of the Poisson distribution. Therefore, the rendering equation is an expectation of the color, following the distribution of the ray terminating at a distance t.

However, the assumptions used to derive this equation are very strong, and does not coincide with the practical expressivity seen in NeRFs. Pixel frustums are typically modelled as pyramids, and there is a lack of reasoning for treating the appearance of occluding particles as a Poisson random variable.

Chen et. al [2] relaxes many of these assumptions while still deriving the same rendering equation. We only give a brief summary in this work, but refer readers to [2] for an in-depth discussion. The work demonstrates that a NeRF density field can be transformed into the density of a Poisson Point Process (PPP), and the NeRF color and density fields together give rise to a "marked" PPP [3], [4]. Here we review the definition and properties of the Poisson Point Process (PPP), a stochastic process that models the distribution of a random collection of points in a continuous space. Much of this discussion is drawn from [3], to which we refer the reader for a more detailed and rigorous treatment.

However, [1], [2] both derive a particle-based model of the environment, which also happens to be probabilistic. We argue that this is not a very natural interpretation of the physical world. Although it is perfectly valid to define a probability on the binary event that something is there or not there, building downstream tasks based on a binary random variable is non-trivial since we are no longer dealing with solid geometry. This also begs the question of why do we take the expectation for the color, instead of another statistic.

This interpretation is also in conflict with certain materials, like glass and fog. Typically NeRFs assign low density to these materials. As humans, we know these materials exist but are simply optically porous. A practical concern is the ability to model the NeRF ρ. Computers cannot store solid surfaces as these regions correspond to $\rho = \infty$. Although one can simply use large numbers for ρ with inconsequential loss of rendering quality, using such large numbers can pose problems for solution methods, such as gradient descent. This is primarily because the two learnable parameters ρ and $\mathbf{c}$ can be orders of magnitude different.

Therefore, the purpose of this work is to provide a non-probabilistic representation of the environment that can potentially make downstream tasks more intuitive to formulate, resolve the incorrect interpretation associated with special materials, and address the computational un-modellability of surfaces. However, since NeRFs have shown themselves to be so powerful, we would also like to prove the expressivity of our representation is no less than that of NeRFs by also proving equivalence in their rendering equations.

III. A Porous Scene Representation

A key insight inspired by the volume integral formulation from [2] and by the fact that glass and fog are not optically dense (i.e. we have less occluding material in a given volume than the actual volume itself). An analogy can be made here that molecules of a gas are farther apart than in a solid, hence more light can pass through. We call this property of a material *optically porous*. Specifically, we define porous as the ratio of occluding volume V_o over the whole volume V_T

$$\phi = \frac{V_o}{V_T}. \tag{3}$$

In the same way [2] defines an integrable field, we can then define a field based on this property, named the *optical porosity field*.

Definition 1 (Optical Porosity Field)**.** *Consider a scalar field Λ that maps subsets $B \subset \mathbb{R}^n$ of the state space to a bounded [0, 1] real number that describes the occupied volume within B. Then, we say the field is a optical porosity field with intensity $\rho : \mathbb{R}^n \mapsto [0, 1]$ if:*

(i) The amount of occupied volume in B is given by

$$\Lambda(B) = \int_{\mathbf{x}\in B} \rho(\mathbf{x})\, d\mathbf{x}.$$

(ii) The fraction of occupancy (or bulk porosity $1 - \mathcal{P}$) is given by $\frac{\Lambda(B)}{V(B)}$, where $V(B)$ is the volume of B.

In essence, we can think of this field as modelling an amorphous sponge, where each point in space models the differential percentage of holes at that point.

A. Optical Porosity Volume Rendering

Due to space considerations, we will only present the results of the analysis, but the full analysis will be included in the final report. The rendered pixel color for our model is

$$\mathbf{C} = \sum_i \mathbf{c}(\mathbf{x}_i)\rho(\mathbf{x}_i) \prod_j^{i-1} \left[1 - \rho(\mathbf{x_j})\right], \tag{4}$$

where $\mathbf{x}_i$ are the sample points along the ray associated with a particular pixel.

Computationally, this formulation is also attractive. Immediately, the field is only between 0 and 1, which matches the range of colors exactly. This makes gradient descent solution methods typically more stable. Depending on solution methods, computers could model surfaces precisely since they no longer need to store and operate on ∞. Moreover, the bounded range makes better use of floating-point precision.

We can also compare the runtime of computing (4) with (2). Our method (4) requires N multiplications. Meanwhile, evaluating (2) using *exponentiation by squaring* is $\mathcal{O}(N \log a)$ where a is the exponent. Since the density values of NeRFs can be large, we can see that the complexity of (2) can be no better than that of (4). Implemented in Pytorch on random examples, we observe anywhere from 2 to 3 times speed up using (4).

B. Expressivity by Equivalence

Although we have shown attractive properties of the optical porosity model, we need to understand the expressivity of such a model compared to the state-of-the-art (i.e. NeRFs). In fact, we can recover (2) immediately from (4) using the one-to-one transformation $(1-\rho) = \exp(-\sigma)$, proving no expressivity gap. We omit the proof here, but will include it in the final report.

IV. Solutions to 3D Volumetric Reconstruction

One possible solution to 3D reconstruction is to minimize the reconstruction loss between the observed pixel color and the *expectation* of the colors over the slices (like in [5]). Although reasonable, this objective may not yield the best geometry, critical for tasks that interact with the geometry. Specifically, there is not necessarily an incentive in the L2 reconstruction loss to prioritize the creation of surfaces, possibly leading to floaters and a fuzzy scene representation.

We also present a maximum log-likelihood loss, derived from our optical model. The full derivation will be included in the final report. The optimization (next page, 5) solves the maximum likelihood problem for our porosity model. Note that $LSE_0^+(a) = \log(1+\exp a)$ is the LogSumExp, $\mathbf{c}, \alpha$ are the model parameters for color and porosity, and s^{-1} is the inverse sigmoid. All terms in the bracket are convex if the model parameters appear affinely in some fixed local area. This is the case for interpolated fields (e.g. Plenoxels [6]) or those that use primitives (e.g. Gaussian Splatting [7]). Due to this, we know that we can find a global optimum through brute force, permuting over the selection of i and solving that particular permutation's unconstrained convex program.

V. Goals

The primary goals of this project is to (1) perform analysis of NeRFs and their solution methods, and (2) provide empirical evidence to support the analysis. Much of (1) has already been performed, though we can certainly perform more analysis. For the remainder of the class, we would like to compare the claims on computational complexity and training stability by training both a NeRF and our optical model and comparing the RGB reconstruction (PSNR of test images) as well as the qualitative geometric reconstruction on the Gaussian Splatting [7] and NeRFacto [8] framework. Additionally, we would like to ablate our optical model using the MSE loss and the MLE loss and compare geometric reconstruction as well as convergence by using the Plenoxel [6] framework to garner insight on optimality of either loss. We provide a timeline below.

- Week 1: Spin up Plenoxel-like trilinear interpolation field and train using the MSE and MLE losses.
- Week 2: Change the rendering equations of both NeRFacto and Gaussian Splatting to (4).
- Final 5 days: Collect results (PSNR, compute time, training losses, qualitative test renders) and make poster.
- March 13-14: Write report.

References

[1] P. L. Williams and N. Max, "A volume density optical model," in *Proceedings of the 1992 Workshop on Volume Visualization*. Association for Computing Machinery (ACM), 1992, p. 61–68.

[2] T. Chen, P. Culbertson, and M. Schwager, "Catnips: Collision avoidance through neural implicit probabilistic scenes," *arXiv preprint arXiv:2302.12931*, 2023.

[3] J. F. C. Kingman, *Poisson processes*, ser. Oxf. Stud. Probab. Oxford: Clarendon Press, 1993, vol. 3.

[4] S. N. Chiu, D. Stoyan, W. Kendall, and J. Mecke, "Point processes I — The Poisson point process," in *Stochastic Geometry and Its Applications*. John Wiley & Sons, Ltd, 2013, ch. 2, pp. 35–63.

[5] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *European Conference on Computer Vision (ECCV)*, 2020.

[6] Sara Fridovich-Keil and Alex Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[7] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, July 2023. [Online]. Available: https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

[8] M. Tancik, E. Weber, R. Li, B. Yi, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. McAllister, A. Kanazawa, and E. Ng, "Nerfstudio: A framework for neural radiance field development," in *SIGGRAPH*, 2023.

TABLE I: Maximum Likelihood Optimization

$$\begin{aligned} &\max_{\mathbf{c},a} \sum_{\mathbf{r}\in D} \max_{\mathbf{x}_i} \log \left[\exp\left[-||\bar{\mathbf{C}} - \mathbf{c}(\mathbf{x}_i)||_{\Sigma^{\mathbf{c}}} \right] a(\mathbf{x}_i) \prod_{j}^{i-1} 1 - a(\mathbf{x}_i) \right] \\ \equiv &\min_{\mathbf{c},\alpha} \sum_{\mathbf{r}\in D} \min_{\mathbf{x}_i} \left[||s^{-1}(\bar{\mathbf{C}}) - \mathbf{c}(\mathbf{x}_i)||_{\Sigma_i^{\mathbf{c}}} + LSE_0^{+}(-\alpha(\mathbf{x}_i)) + \sum_{j}^{i-1} \big(\alpha(\mathbf{x}_i) + LSE_0^{+}(-\alpha(\mathbf{x}_i)) \big) \right] \end{aligned} \tag{5}$$