Sparse-to-Dense Depth Completion Using Regularized Optimization

Eric Wagner
University of Maryland
College Park, United States
emwagner56@gmail.com
116484695

Shashank Ashoka
University of Maryland
College Park, United States
shashk09@umd.edu
122241329

Simran Kharbanda
University of Maryland
College Park, United States
simrank@umd.edu
122283671

***Abstract*—Sparse-view 3D reconstruction is challenging because limited observations make the geometry highly under-constrained. While Neural Radiance Fields (NeRF) can produce high quality reconstructions, this project focuses on the core geometric subproblem of sparse-to-dense depth completion as a regularized optimization problem. We add smoothness and edge aware regularization terms to the objective and solve for the dense depth map itself using gradient based methods, Adam and L-BFGS and evaluate performance on the NYU Depth V2 dataset. Experimental results show our optimization framework consistently improves reconstruction quality compared to nearest neighbors from the applied regularization terms. Additional studies demonstrate trade-offs in solvers between convergence speed and numerical stability.**

***Keywords*—Sparse-to-Dense Depth Completion, Neural Radiance Fields, Regularized Optimization, Edge-Aware Regularization, 3D Reconstruction, Adam, L-BFGS**

I. Introduction

Reconstructing dense 3D geometry from sparse observations is a fundamental problem in computer vision, robotics, and neural rendering. With only a sparse number of measurements available, the problem is severely under-constrained since multiple dense depth maps may satisfy the same observations. This creates unstable geometry, noisy reconstructions, and unrealistic boundaries.

Recent sparse-view Neural Radiance Field (NeRF) methods such as SimpleNeRF address this issue by introducing regularization terms that guide optimization toward geometrically plausible solutions. However, full NeRF training is computationally expensive, making it difficult to analyze the optimization behavior. To avoid this, we isolate the depth completion component of NeRF.

Specifically, we will isolate the problem to when given an RGB image and sparse depth observations, reconstruct a dense depth map over all pixels in the image. This way our optimization variable is the dense depth map itself rather than the parameters of a neural rendering model, which we express as z∈R^H×W, where z is the depth of each pixel in the image with HxW dimensions.

To improve reconstruction we add several terms to the objective function including a data fidelity term that enforces consistency with sparse depth values, a smoothness regularizer that promotes stable geometry, and an edge-aware regularization term that preserves discontinuities near object boundaries. These terms allow the project to focus on how regularization shapes the optimization and reconstruction quality. Numerical solvers such as Adam and L-BFGS are then used to solve the problem and analyze convergence behavior, runtime and reconstruction accuracy.

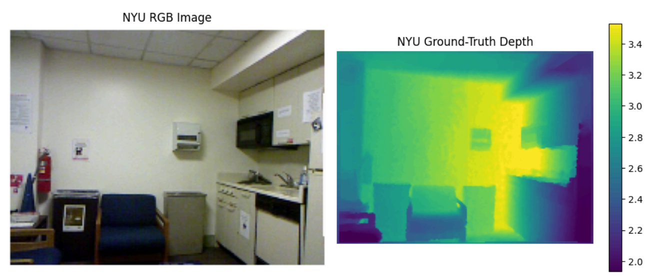

Fig.1.. Example NYU Depth V2 RGB-D sample. The left image shows the RGB input, and the right image shows the corresponding dense ground-truth depth map used for evaluation. In our experiments, we randomly retain only a small percentage of the ground-truth depth pixels to form sparse observations, while the full dense depth map is hidden from the optimizer and used only to compute RMSE and MAE.

II. Motivation

Many modern systems, including robotics platforms, autonomous vehicles, augmented and virtual reality rendering, rely on accurate depth estimation to understand the structure of a scene. However, in practical settings, depth measurements are often incomplete, noisy, or extremely sparse due to sensor limitations, restricted viewpoints, or limited numbers of input images.

Sparse-view Neural Radiance Fields (NeRF) methods have demonstrated that limited observations frequently lead to unstable geometry and reconstruction artifacts such as blurry surfaces, incorrect boundaries, or floating structures in empty regions of space. These failures occur as a product of the problem being underconstrained since with multiple geometric solutions satisfying the same sparse measurement, optimization methods may converge to physically implausible local minima.

Recent NeRF-based approaches, such as SimpleNeRF, address this issue by introducing additional regularization terms to guide optimization toward more realistic solutions.

However full NeRF pipelines are computationally expensive and render large neural network architectures, making them difficult to analyze directly from an optimization perspective.

Motivated by these challenges, our project focuses on the underlying sparse geometry reconstruction rather than the full neural rendering system by optimizing the dense depth map using regularized optimization. This simplification allows optimization variables, constraints and numerical behavior of the solver to be studied more explicitly while still capturing the core difficulty of sparse-view reconstruction.

Our approach introduces smoothness and edge-aware regularization terms motivated by the observation that naive methods, such as nearest-neighbor filling, often produce unstable geometry and large reconstruction errors near edges. We also incorporate geometric priors directly into the objective function so the optimization process can better infer missing depth values in under-contrained regions.

From an optimization standpoint, this is a fascinating problem because it highlights how objective terms and numerical solver influence convergence and reconstruction quality. Being a severely unconstrained problem, there is an additional challenge to not converge to local minima that satisfy the minimization but correspond to physically implausible geometry. The project also provides an opportunity to compare first-order optimization methods such as Adam with quasi-Newton approaches such as L-BFGS in a practical reconstruction problem. Ultimately the goal is to better understand how regularized optimization can improve spare geometric reconstruction while remaining computationally simple and more interpretable than full neural rendering pipelines.

III. Related Work

Several recent studies have attempted to improve the performance of Neural Radiance Fields when only a limited number of input images are available.

• DS-NeRF incorporates depth supervision into the NeRF training process by using sparse depth estimates obtained from multi-view geometry. This additional geometric information helps guide the model toward more accurate scene structures.

• RegNeRF addresses the sparse-view problem by introducing regularization techniques that enforce consistency across synthesized viewpoints. The method generates pseudo views and applies appearance consistency constraints to improve generalization when the number of input images is small.

• FreeNeRF proposes a frequency regularization strategy that stabilizes training by gradually allowing the network to learn higher-frequency scene details over time.

While these methods focus on different strategies to improve sparse-view reconstruction, our project approaches the problem from an optimization perspective. We formulate the NeRF training process as a multi-term optimization problem that combines reconstruction loss with several regularization terms designed to reduce artifacts commonly observed when training with sparse inputs

IV. Mathematical Formulation

A. Background: Neural Radiance Field Rendering Objective

Neural Radiance Fields (NeRF) represent a 3D scene as a continuous function parameterized by a multi-layer perceptron (MLP). Given a spatial location $x \in R^3$ and a viewing direction d, NeRF predicts the volume density σ and view-dependent color c at that location [1]:

$$F_\Theta : (x, d) \rightarrow (c, \sigma). \qquad (1)$$

For a camera ray

$$r(t) = o + td, \quad (2)$$

where o is the camera origin and d is the ray direction, NeRF renders the expected color of the ray using differentiable volume rendering:

$$C(r) = \int_{t_n}^{t_f} T(t)\sigma(r(t))c(r(t), d)dt, \quad (3)$$

where t_n and t_f are the near and far bounds of the ray. The transmittance term $T(t)$ is defined as

$$T(t) = exp\left(- \int_{t_n}^{t} \sigma(r(s))ds\right), \quad (4)$$

The network parameters Θ are optimized by minimizing the photometric reconstruction loss:

$$L_{rgb} = \sum_{r \in R} \| \hat{C}(r) - C(r) \|_2^2. \qquad (5)$$

This objective is effective when many views are available. However, in sparse-view settings, the photometric loss may not uniquely determine scene geometry.Multiple density configurations can explain the same limited set of observed images, which makes geometry recovery under-constrained.

This observation motivates our reformulation. Instead of learning geometry implicitly through volume density σ and photometric loss, we study the geometric subproblem directly: recovering a dense depth map from sparse depth observations. In our formulation, the unknown is not the NeRF network parameter Θ, but the dense depth map

$$z \in R^{H \times W}. \qquad (6)$$

B. Motivation from SimpleNeRF Regularization

SimpleNeRF addresses the sparse-view NeRF problem by regularizing NeRF training with simpler auxiliary models [2]. Its main observation is that when only a few input views are available, the standard photometric loss may allow implausible scene geometries. To reduce this ambiguity, SimpleNeRF introduces auxiliary NeRF models that produce depth estimates biased toward simpler geometric explanations. In SimpleNeRF, the color of a pixel is still computed through NeRF-style volume rendering. For sampled points along a ray, the model predicts density and color, and combines them using weights

$$w_i = exp\left(-\sum_{j=1}^{i-1} \delta_j \sigma_j\right)\left(1 - exp\left(-\delta_i \sigma_i\right)\right). \qquad (7)$$

The expected ray depth is then computed as

$$z = \sum_{i=1}^{N} w_i z_i, \qquad (8)$$

where z_i is the depth of the i-th sampled point along the ray. This is important for our project because it shows that NeRF-based methods can produce depth estimates from volume-rendering weights, and these depth estimates can be regularized during training.

SimpleNeRF uses sparse depth supervision from SfM through the loss

$$L_{sd} = \| z_f - \hat{z} \|^2 + \| z_{ap} - \hat{z} \|^2 + \| z_{av} - \hat{z} \|^2, \qquad (9)$$

where $\hat{z}$ is sparse depth from SfM, z_f is the fine NeRF depth, and z_{ap}, z_{av} are depths predicted by the augmented NeRF models. This term motivates the data-fidelity part of our objective: where sparse depth is available, the recovered dense depth should agree with it.

SimpleNeRF also introduces consistency losses between the main and augmented models. For example, its coarse-fine consistency loss is

$$L_{cfc} = 1_{\{m_{cfc}=1\}} \odot \| z_c - \tau\left(z_f\right) \|^2 + 1_{\{m_{cfc}=-1\}} \odot \| \tau\left(z_c\right) - z \qquad (10)$$

where z_c and z_f are coarse and fine depth estimates, m_{cfc} is a reliability mask, and $\tau(\cdot)$ is the stop-gradient operator.

The connection between SimpleNeRF and our formulation is conceptual rather than architectural. SimpleNeRF uses auxiliary neural models to regularize depth implicitly; our method expresses the same type of geometric guidance as explicit regularization terms in a depth-completion objective.

Specifically, SimpleNeRF's sparse depth supervision motivates our sparse-depth fidelity term,

$$\alpha \, \| \, M \odot \left(z - z_\Omega\right) \|_2^2. \qquad (11)$$

SimpleNeRF's preference for simpler geometry, achieved through reduced positional encoding, motivates our smoothness regularization term,

$$\beta \, \| \, \nabla z \|_2^2. \qquad (12)$$

SimpleNeRF's view-independent radiance augmentation encourages simpler Lambertian explanations in regions where view-dependent color can create shape-radiance ambiguity. In our formulation, we encode an analogous image-guided geometric prior using the edge-aware regularizer,

$$R_{edge}(z, I) = \sum_{i,j} exp\left(-\kappa \, \| \, \nabla I_{ij} \, \|\right) \| \, \nabla z_{ij} \|_1. \qquad (13)$$

Finally, SimpleNeRF's coarse-fine consistency motivates our coarse-consistency term,

$$\delta \, \| \, z - z_c \|_2^2, \qquad (14)$$

which stabilizes the optimized depth map relative to a coarse interpolation estimate.

C. Proposed Optimization Formulation

The goal of our project is to solve **sparse-to-dense depth completion**. This problem has been studied in computer vision because sparse depth measurements, obtained from sensors such as LiDAR, SLAM, or SfM, are often easier to obtain than dense depth maps. Prior work such as Sparse-to-Dense depth prediction studies the task of estimating a dense depth image from sparse depth samples and a single RGB image [3]. Related depth-completion work also combines raw depth observations with image-derived geometric cues such as surface normals or occlusion boundaries to solve for missing depth values [4].

Let

$$I \in R^{H \times W \times 3} \qquad (16)$$

denote an RGB image, and let

$$z_\Omega \qquad (17)$$

denote sparse observed depth values. The observed sparse depth locations are represented by a binary mask

$$M \in \{0, 1\}^{H \times W}, \qquad (18)$$

where $M_{ij} = 1$ if depth is observed at pixel (i, j), and $M_{ij} = 0$ otherwise. The optimization variable is the dense depth map

$$z \in R^{H\times W}. \quad (19)$$

Unlike NeRF or SimpleNeRF, where the optimization variables are neural network parameters, our variable is the depth value at every pixel.

We solve the following constrained regularized optimization problem:

$$\min_{z\geq 0} \alpha \| M \odot \left(z - z_{\Omega}\right)\|_2^2 + \beta \| \nabla z\|_2^2 + \gamma R_{edge}(z, I) + \delta \| z - z_c\|_2^2. \quad (20)$$

Here, $\odot$ denotes element-wise multiplication, z_c is a coarse depth estimate obtained from nearest-neighbor interpolation, and $\alpha, \beta, \gamma, \delta \geq 0$ control the relative strength of the objective terms

D. Interpretation of Objective Terms

1) Sparse-depth fidelity

The first term is

$$\alpha \| M \odot \left(z - z_{\Omega}\right)\|_2^2. \quad (21)$$

This term is motivated by sparse depth supervision used in depth-completion and sparse-view reconstruction methods. In Sparse-to-Dense depth prediction, sparse depth samples are explicitly used to improve dense depth prediction from RGB images [3]. Similarly, in SimpleNeRF, sparse SfM depth is used as geometric supervision for NeRF-predicted depth [2]. Our formulation expresses this idea directly as a squared-error penalty between the optimized dense depth and sparse observed depth.

2) Smoothness regularization

The second term is

$$\beta \| \nabla z\|_2^2. \quad (22)$$

This term penalizes large spatial depth gradients and encourages neighboring pixels to have similar depth values. Since only a small fraction of depth pixels are observed, the reconstruction problem is under-constrained. Total variation regularization was introduced for image denoising by Rudin, Osher, and Fatemi and has since become a standard tool in imaging inverse problems [5]. Our L_2smoothness term is smoother than total variation but serves a similar purpose reducing unstable local variation in unobserved regions.

3) Edge-aware regularization

The third term is

$$\gamma R_{edge}(z, I), \quad (23)$$

where

$$R_{edge}(z, I) = \sum_{i,j} exp\left(- \kappa \| \nabla I_{ij} \|\right) \| \nabla z_{ij}\|_1. \quad (24)$$

This is an **edge-aware smoothness penalty**. It uses RGB image gradients to decide where depth should be smooth and where depth discontinuities should be allowed. If the RGB gradient $\| \nabla I_{ij} \|$ is small, then

$$exp\left(- \kappa \| \nabla I_{ij} \|\right)$$

is close to 1, so depth smoothness is strongly encouraged. If the RGB gradient is large, the exponential weight becomes small, so the optimizer allows a depth discontinuity near a likely object boundary.

Edge-aware smoothness losses are widely used in depth-estimation literature because image reconstruction or sparse supervision alone can produce poor-quality depth unless additional geometric consistency terms are imposed. For example, Godard et al. show that reconstruction loss alone can lead to poor depth and add consistency-based regularization for improved robustness [6]. Depth-completion methods also use image-derived boundaries or normals to guide missing depth reconstruction [4]. Our formulation adopts this principle explicitly by weighting the depth-gradient penalty using RGB image gradients.

4) Coarse-consistency term

The fourth term is

$$\delta \| z - z_c\|_2^2. \quad (25)$$

Here,

$$z_c = NNFill\left(z_{\Omega}\right) \quad (26)$$

is a coarse depth estimate obtained by nearest-neighbor interpolation of the sparse depth observations. This term weakly anchors the optimized solution to an initial coarse reconstruction

The purpose of this term is not to make the solution identical to nearest-neighbor interpolation. Instead, it provides a stabilizing prior that prevents the optimizer from drifting too far in poorly observed regions. This is conceptually related to consistency regularization in

SimpleNeRF, where different depth estimates are encouraged to agree when sparse-view training is under-constrained [2].

5) Nonnegative-depth constraint

The constraint in (20) is

$$z \geq 0. \qquad (27)$$

This constraint enforces physical validity: depth values cannot be negative. In implementation, we enforce the feasible set using a projection step after solver updates:

$$z \leftarrow max(z, 0). \quad (28)$$

This is a simple projected-optimization approach. In our checked NYU Depth V2 run, the constraint was not active: the minimum predicted depth was 1.9279 for Adam and 1.9359 for L-BFGS, with zero pixels clamped to zero.

V. Numerical Solvers

1) Adam

Adam is a first-order adaptive gradient method that updates variables using estimates of the first and second moments of the gradient [7]. It is computationally efficient, has low memory requirements, and is commonly used for large optimization problems in machine learning.

For our variable z, the Adam update can be written as

$$z_{t+1} = z_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}}, \qquad (29)$$

where η is the learning rate, $\hat{m}_t$ is the bias-corrected first-moment estimate of the gradient, $\hat{v}_t$ is the bias-corrected second-moment estimate, and ϵ is a small constant for numerical stability.

In our implementation, z is treated as a PyTorch optimization variable. After each Adam update, the nonnegative-depth constraint is enforced by projection.

2) L-BFGS

L-BFGS is a limited-memory quasi-Newton method designed for large-scale optimization problems. Instead of storing the full Hessian matrix, it approximates curvature information using a limited history of previous gradients and updates.

For our objective $f(z)$, the L-BFGS update can be written abstractly as

$$z_{t+1} = z_t - \eta H_t \nabla f\left(z_t\right), \qquad (30)$$

where H_t is the limited-memory approximation to the inverse Hessian, $\nabla f\left(z_t\right)$ is the gradient of the objective in (20), and η is chosen through line search.

L-BFGS is suitable for our problem because the objective contains several smooth quadratic terms and a structured edge-aware regularizer. In our experiments, L-BFGS achieved similar RMSE to Adam while reducing runtime.

F. Summary of the Optimization Design

$$\min_{z \geq 0} \; \underbrace{\alpha \| M \odot (z - z_\Omega) \|_2^2}_{\text{sparse-depth fidelity}} + \underbrace{\beta \| \nabla z \|_2^2}_{\text{smoothness}} + \underbrace{\gamma R_{\text{edge}}(z, I)}_{\text{edge-aware prior}} + \underbrace{\delta \| z - z_c \|_2^2}_{\text{coarse consistency}}$$

(31)

Together, these terms convert sparse depth completion into a well-defined constrained optimization problem that can be solved and analyzed using numerical solvers.

VI. Numerical Results

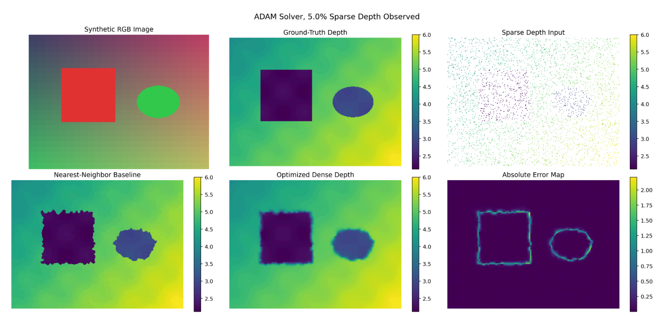

Fig. 2. Sparse-depth reconstruction example. With only 5% observed depth, nearest-neighbor interpolation gives a rough dense estimate with jagged boundaries. The proposed regularized optimization recovers a smoother dense depth map, with most residual error concentrated near object boundaries.

Table 3.1

Observed Depth	NN RMSE	Optimized RMSE	NN MAE	Optimized MAE	Runtime
1.0%	0.1281	0.0884	0.0681	0.0552	3.67s
5.0%	0.0730	0.0540	0.0331	0.0267	3.54s
10.0%	0.0621	0.0433	0.0243	0.0189	3.52s

Solver
Adam
L-BFGS

We evaluate sparse-to-dense depth completion on one NYU RGB-D sample by retaining only **1%, 5%, and 10%** of the ground-truth depth pixels as sparse observations. The remaining depth pixels are hidden from the optimizer and used only for evaluation. Compared with nearest-neighbor interpolation, the proposed optimization consistently reduces both RMSE and MAE across all sparsity levels. As the percentage of observed depth increases, the reconstruction error decreases, showing that additional sparse depth anchors make the optimization problem better constrained.

TABLE 3.2

Variant	RMSE	MAE	Runtime
Data only	0.2271 ± 0.1259	0.0753 ± 0.0380	2.37 ± 0.03s
Edge only	0.1838 ± 0.1067	0.0630 ± 0.0340	2.39 ± 0.03s
L2 only	0.1837 ± 0.1057	0.0826 ± 0.0430	2.37 ± 0.02s
L2 + Edge	0.1613 ± 0.0940	0.0676 ± 0.0342	2.43 ± 0.04s
Full objective	0.1530 ± 0.0880	0.0615 ± 0.0300	2.38 ± 0.03s

VARIANTS AT **5%** OBSERVED DEPTH OVER **N=5000** SAMPLES. THE **DATA-ONLY** MODEL IS WEAK BECAUSE SPARSE MEASUREMENTS CONSTRAIN ONLY A SMALL FRACTION OF PIXELS. ADDING **SMOOTHNESS** AND **EDGE-AWARE REGULARIZATION** REDUCES RECONSTRUCTION ERROR BY PROPAGATING STRUCTURE INTO UNOBSERVED REGIONS WHILE PRESERVING LIKELY DEPTH BOUNDARIES.

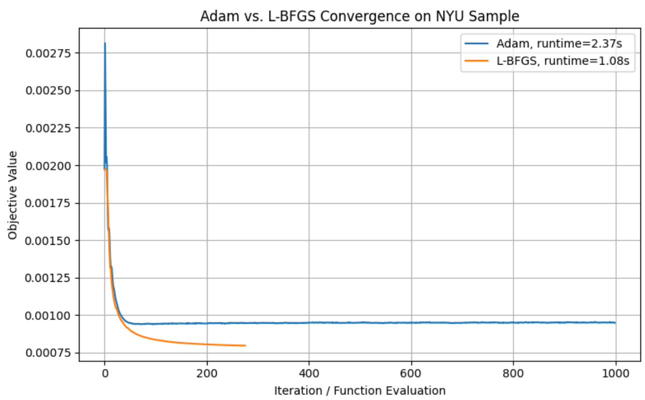

Fig. 3. We compare Adam and L-BFGS on the same 5% sparse-depth reconstruction objective. Both methods rapidly decrease the objective in the first few iterations, confirming that the proposed loss is well-conditioned enough for practical numerical optimization. L-BFGS reaches a lower objective value with fewer function evaluations and lower runtime, while Adam converges more gradually and plateaus after the initial descent. This supports the solver comparison in Table III, where L-BFGS provides similar reconstruction accuracy with substantially faster optimization.

TABLE 3.3

Solver	RMSE	MAE	Runtime
Adam	0.1451 ± 0.0750	0.0591 ± 0.0271	2.41 ± 0.03s
L-BFGS	0.1477 ± 0.0767	0.0576 ± 0.0270	0.76 ± 0.01s

Adam and L-BFGS achieve comparable RMSE and MAE at 5% observed depth, showing that the proposed objective is stable across different optimization methods. L-BFGS reduces runtime from 2.41s to 0.76s, giving a 3.2× speedup while preserving reconstruction accuracy. This demonstrates that the proposed formulation is both effective and computationally efficient to optimize.

VII. CHALLENGES ENCOUNTERED

During the development of the project, several technical and practical challenges were encountered while designing an efficient depth reconstruction and optimization framework. One of the primary difficulties was the attempt to integrate a NeRF-inspired architecture into the pipeline. Although Neural Radiance Fields provide highly detailed 3D scene representations, the volumetric rendering process required extensive computation and memory resources. Training the model involved repeated ray sampling and iterative rendering operations, which significantly increased execution time and GPU utilization. Due to these limitations, the original approach became impractical for rapid experimentation and real-time testing within the available computational constraints.

Another major challenge was handling sparse depth information. The dataset contained only a very small percentage of valid depth measurements, typically between 1% and 5% of the total pixels. Traditional interpolation methods such as nearest-neighbor filling produced inconsistent outputs with visible holes, blurred boundaries, and reconstruction artifacts. These issues became particularly severe around object edges and textured regions where accurate depth continuity was important. Since the sparse input lacked sufficient structural information, generating dense and geometrically consistent depth maps proved difficult.

Balancing the optimization objective was also a challenging aspect of the project. The reconstruction framework relied on multiple loss components, including reconstruction loss, smoothness regularization, and edge-aware consistency constraints. Adjusting these losses required careful experimentation because emphasizing one objective often negatively affected another. Increasing smoothness terms reduced noise but oversmoothed important scene details, while stronger edge-aware constraints preserved boundaries but occasionally introduced instability during optimization. Determining appropriate weights for each component required repeated tuning and validation.

Optimizer selection introduced another trade-off between convergence stability and training efficiency. The Adam optimizer produced stable and reliable convergence across most experiments, but the optimization process was relatively slow for large-scale reconstruction tasks. On the other hand, the L-BFGS optimizer achieved faster convergence in some cases, but it was highly sensitive to initialization conditions and parameter settings. Minor changes in hyperparameters sometimes caused divergence or unstable outputs, making the optimization process difficult to control consistently.

The transition from synthetic datasets to real-world scenes presented additional challenges. Models that performed well on synthetic environments often struggled when evaluated on real captured data. Real-world depth images contained sensor noise, incomplete measurements, illumination inconsistencies, and scaling variations that were not present in synthetic training conditions. These factors reduced reconstruction quality and required additional preprocessing, normalization, and filtering steps before the optimization pipeline could be applied effectively.

VIII. LESSONS LEARNED

One of the most important lessons learned from this project was that highly complex approaches such as full

NeRF-based volumetric rendering are not always practical under limited computational resources. While NeRF provides high-quality scene representation capabilities, simplified optimization-based methods offer a more balanced solution in terms of computational cost, scalability, and reconstruction performance for this project.

The project also demonstrated the importance of carefully designing multi-loss optimization functions. Reconstruction quality depended heavily on the interaction between data fidelity, smoothness constraints, and edge-preserving regularization. Properly balancing these objectives was critical for generating visually consistent and geometrically accurate depth outputs.

Another key lesson involved the significance of hyperparameter tuning. Parameters such as learning rates, optimizer settings, regularization weights, and initialization methods had a substantial impact on convergence stability and reconstruction accuracy. Even small parameter adjustments produced noticeable differences in performance, emphasizing the need for systematic experimentation and validation.

The experiments further highlighted that real-world computer vision tasks require extensive preprocessing. Unlike synthetic datasets, real sensor data contains noise, missing values, and environmental inconsistencies that directly affect model behavior. Effective preprocessing and normalization became essential steps for improving reconstruction quality and optimization stability.

Finally, the project reinforced the fundamental trade-off between accuracy, latency, and scalability. Methods that improved reconstruction accuracy often introduced higher computational overhead and longer inference times, whereas lightweight methods provided faster execution at the cost of reduced precision. Understanding and managing this balance became one of the central insights gained throughout the project.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our course instructor, Richard La, for his continuous guidance, valuable feedback, and support throughout the development of this project. His insights and encouragement greatly helped us understand the practical and theoretical aspects of optimization and computer vision systems.

We also thank the College of Computer, Mathematical, and Natural Sciences at University of Maryland for providing the academic environment, resources, and opportunities necessary for successfully completing this work.

In addition, we acknowledge the researchers and authors whose published papers, datasets, and open-source contributions served as important references during the course of this project. Their work in the areas of computer vision, depth estimation, neural rendering, and optimization significantly contributed to our understanding and implementation of the proposed system.

Finally, we would like to thank our teammates and peers for their collaboration, discussions, and continuous support throughout the project. Their teamwork and collective effort played an important role in the successful completion of this work.

REFERENCES

[1] 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 Proc. European Conference on Computer Vision (ECCV), 2020.

[2] N. Somraj, A. Karanayil, and R. Soundararajan, “SimpleNeRF: Regularizing sparse input neural radiance fields with simpler solutions,” in Proc. SIGGRAPH Asia Conference Papers, 2023.

[3] F. Ma and S. Karaman, “Sparse-to-dense: Depth prediction from sparse depth samples and a single image,” in Proc. IEEE International Conference on Robotics and Automation (ICRA), 2018.

[4] Y. Zhang and T. Funkhouser, “Deep depth completion of a single RGB-D image,” in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[5] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” Physica D: Nonlinear Phenomena, vol. 60, no. 1–4, pp. 259–268, 1992.

[6] C. Godard, O. Mac Aodha, and G. J. Brostow, “Unsupervised monocular depth estimation with left-right consistency,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[7] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. International Conference on Learning Representations (ICLR), 2015.

[8] D. C. Liu and J. Nocedal, “On the limited memory BFGS method for large scale optimization,” Mathematical Programming, vol. 45, pp. 503–528, 1989.

[9] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, “A limited memory algorithm for bound constrained optimization,” SIAM Journal on Scientific Computing, vol. 16, no. 5, pp. 1190–1208, 1995.

[10] MIT VisionBook, “Computer Vision: Models, Learning, and Inference,” Massachusetts Institute of Technology, 2023.

[11] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from RGBD images,” in Proc. European Conference on Computer Vision (ECCV), 2012.

[12] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge, U.K.: Cambridge University Press, 2004.

[13] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2014.

[14] N. Silberman and R. Fergus, “NYU Depth Dataset V2,” New York University, 2012.