Milestone Report: Optimizing 3D Object Reconstruction

A Comparative Study of Differentiable Volumetric Rendering and Signed Distance Functions

Sina MohammadiNiyaki, Ali Nikan
Department of Computing Science
Simon Fraser University
Burnaby, Canada
{sma231, ana106}@sfu.ca

I. Progress Overview

The primary goal of our project is to conduct a systematic comparison of two advanced techniques for 3D object reconstruction: Differentiable Volumetric Rendering (DVR) [1] and Signed Distance Functions (DeepSDF) [2]. Our original plan was ambitious, aiming to set up and train both DVR and DeepSDF on DTU and ShapeNet datasets [3], [4], followed by evaluating their performance in terms of reconstruction quality, computational efficiency, and scalability. At this midpoint, we have encountered several challenges and made necessary adjustments to our initial plan.

A. Completed Tasks

We began with an attempt to configure the DeepSDF environment using Anaconda and later Ubuntu via the Windows Subsystem for Linux (WSL). This effort encountered substantial difficulties, including dependency conflicts, CMake configuration errors, and GPU compatibility problems related to OpenGL and Pangolin. After considerable troubleshooting without success, we decided to temporarily prioritize DVR to keep project progress on track.

Our DVR setup initially involved adjustments to the provided `environment.yaml` file. After mistakenly downgrading to Python 3.7 to match the PyTorch and Torchvision version in the environment.yaml, we recognized that Python 3.8 provided better compatibility with our updated PyTorch and torchvision versions since we could not install the author's versions. Specifically, PyTorch version 1.7.1 and torchvision 0.8.2, paired with CUDA 11, were necessary to fully support our GPU (RTX 3070). Addressing additional issues related to the Python Imaging Library (PIL) led to a significant milestone: the successful execution of the DVR demo using pre-trained weights [1].

B. Tasks Remaining

Despite successful execution of the DVR demo, several critical tasks remain. First, we must resolve ongoing DVR data loader issues with the DTU dataset, caused by incorrect placeholder paths within configuration files. We had to create a configuration file to train the model on the DTU dataset since there was no configuration file for the DTU dataset present [1]. Following that, we will train the DVR model on the DTU dataset. Then, we will revisit the DeepSDF environment setup to enable direct methodological comparisons.

After training, our plan includes a quantitative evaluation of reconstruction quality using statistical metrics, such as Chamfer Distance and Intersection over Union (IoU) [1], [2]. Additionally, we aim to assess computational efficiency by analyzing training time, inference speed, and memory consumption [1], [2].

II. Issues Encountered and Adjustments Made

The complexity involved in configuring research-oriented software was significantly underestimated. Often, these research-based implementations lag in maintaining compatibility with evolving software and hardware, which resulted in extensive troubleshooting and delays.

We encountered unexpected technical issues, notably GPU compatibility problems with CUDA, PyTorch, and OpenGL during the DeepSDF setup. Furthermore, integrating the DVR model with the DTU dataset led to unforeseen data loader errors. Resolving these issues requires substantial additional time and introduced tasks that were not initially planned, including detailed documentation of troubleshooting efforts.

Given these challenges, adjustments were necessary for our original project timeline. We prioritized DVR training on the DTU dataset due to computational constraints, deferring potential training on the larger ShapeNet dataset. We also recognized the need to adjust our expectations with respect to the feasibility of developing a hybrid DVR-DeepSDF model.

III. Updated Timeline and Responsibilities

Our revised timeline, outlined in Table I, considers the adjustments required due to encountered challenges:

IV. Next Steps

Our immediate next step is to resolve the existing DVR data loader issues for the DTU dataset. The dataset integration encountered problems primarily due to incorrect placeholder paths, causing data loading errors. Successfully

Date	Task	Responsible
Mar 5–10	Resolve DTU data loader issues (DVR), revisit DeepSDF setup	Both
Mar 11–17	Train DVR on DTU dataset	Sina
Mar 18–24	Evaluate reconstruction metrics (Chamfer, IoU)	Ali
Mar 25–31	Implement optimizations (Sparse grids for DVR; Hash-encodings for DeepSDF)	Both
Apr 1–2	Explore hybrid DVR-DeepSDF model feasibility	Both
Apr 3–6	Finalize benchmarking, prepare presentation	Both
Apr 7–11	Submit final project report	Both

TABLE I: Updated Timeline and Team Responsibilities

addressing this issue will enable us to proceed with DVR model training, which is essential for achieving our primary goal of comparing reconstruction methods.

Once the DVR model is successfully trained on the DTU dataset, we will quantitatively evaluate its reconstruction quality using metrics such as Chamfer Distance and Intersection over Union (IoU). These metrics provide concrete measures to objectively assess how accurately the DVR model reconstructs 3D shapes. Additionally, we will evaluate computational efficiency by measuring training time, inference speed, and memory consumption, which are crucial for understanding real-world applicability.

If DVR training and evaluation proceed smoothly, our next priority will be to revisit and complete the DeepSDF environment setup. Despite earlier challenges, having both methods operational is vital to our comparative study. Due to computational constraints and time limitations, training models on the larger ShapeNet dataset remains secondary; however, it will still be considered if sufficient time and resources become available.

Furthermore, as we move toward later stages of the project, we aim to implement specific optimization strategies. Sparse voxel grids will be explored to potentially reduce DVR's memory footprint, while hash-based encodings will be considered for DeepSDF to accelerate training times and enhance scalability.

Finally, we will evaluate the feasibility of developing a hybrid DVR-DeepSDF model, combining the strengths of both methods. However, given the unexpected complexities encountered thus far, we will carefully reassess this goal based on our available resources and remaining timeline, ensuring we focus our efforts effectively.

V. Conclusion

Despite encountering significant challenges related to environment setup, dependency management, and dataset integration, our project has made meaningful progress. Setting up complex research-oriented software frameworks proved considerably more difficult than anticipated, primarily due to compatibility issues between older software components and modern hardware. Nevertheless, successfully executing the DVR demo with pre-trained models was a critical milestone, demonstrating our capability to overcome technical hurdles and providing a stable platform for subsequent steps. Moving forward, we have strategically adjusted our goals, focusing first on addressing immediate technical challenges with the DVR data loader, and then on training and evaluating the DVR model. We remain optimistic that, through careful prioritization, realistic planning, and detailed documentation of encountered issues, we will achieve substantial comparative analysis between DVR and DeepSDF methodologies, ultimately delivering meaningful results by the conclusion of this term.

References

[1] M. Niemeyer, L. M. Mescheder, M. Oechsle, and A. Geiger, "Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision," *arXiv preprint*, vol. abs/1912.07372, 2019. [Online]. Available: http://arxiv.org/abs/1912.07372

[2] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [Online]. Available: https://arxiv.org/abs/1901.05103

[3] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "Shapenet: An information-rich 3d model repository," *arXiv preprint*, 2015. [Online]. Available: https://arxiv.org/abs/1512.03012

[4] R. Jensen, A. Aanæs, H. W. Jensen, and R. Larsen, "Large-scale multi-view stereo dataset," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 406–413, 2014. [Online]. Available: https://roboimagedata.compute.dtu.dk/?page_id=36