HydroNeRF-SLAM: NeRF SLAM for Underwater Scene Reconstruction

Robert Duquette
University of Michigan
robduq@umich.edu

Junzhe Wu
University of Michigan
junzhewu@umich.edu

Max Rucker
University of Michigan
mruck@umich.edu

***Index Terms*—underwater, NeRF, SLAM**

***Abstract*—Autonomous Underwater Vehicles (AUVs) are becoming widely used to explore and map underwater areas for further research, however, many AUVs cannot create dense reconstructions of these environments only creating sparse reconstructions of their surrounding environments. With that, this paper evaluates different NeRF algorithms' ability to recreate underwater scenes from monocular camera data in a dense rendering. The two NeRF algorithms evaluated in this paper are SeaThruNeRF, a NeRF algorithm specifically designed for underwater scenes, and NeRF-SLAM, a NeRF algorithm that utilizes deep learning visual SLAM to help estimate poses for NeRF reconstruction. Along with these two algorithms, this paper introduces the use of UGAN, a generative adversarial network used for underwater image reconstruction, as a preprocessing method for NeRF-SLAM to test if that aids the performance of NeRF compared to the model used by SeaThruNeRF. From the results, it is clear that further work needs to be done to improve NeRF reconstruction for large underwater scenes, and that current GAN methods require temporal image reconstruction to benefit NeRF rendering.**

I. INTRODUCTION

Within marine robotics, one of the biggest challenges has been the mapping of underwater environments. Whether it be sunken shipwrecks, underwater caves, or coral reefs, there is an effort to explore more of these areas. Autonomous underwater vehicles (AUVs) offer great opportunities to explore these areas safely and collect data on many unreachable areas [1].

NeRF [2], which stands for Neural Radiance Fields, is an innovative new mapping technique that involves mapping the volumetric density and color of a scene at any given 3D point using neural networks. This is different from usual techniques which usually use geometric point clouds or meshes for visual maps. With NeRF, instead of having discrete point clouds the volumetric map from the neural networks is continuous leading to a smoother visual output [2]. This has led to better results for visually analyzing and creating 3D maps. Along with this, NeRF allows for 3D maps to be viewed from angles that were not captured in the original dataset. Overall, NeRF technology has shown strong results for digitally reconstructing objects with enough training images for the neural network.

For underwater mapping, new developments have been made to apply NeRF to underwater imaging to produce more detailed maps using AUVs [3]. While many innovations are being made, there are many inherent issues caused by underwater environments. These include attenuation and backscattering from light traveling through water, simultaneous localization and mapping (SLAM) limitations, dynamic objects, etc. Even with these limitations, there is some evidence showcasing NeRF's ability to proficiently handle environments with inconsistent lighting and sparse environments [4]. With this said, underwater NeRF is still a relatively new concept that has little research behind it, and we hope to fill in that gap with our project. Our project tests state-of-the-art NeRF algorithms and evaluates their accuracy at mapping an underwater scene utilizing monocular camera information in an underwater dataset. We also propose utilizing a GAN image correction system to help aid underwater NeRF scene reconstruction.

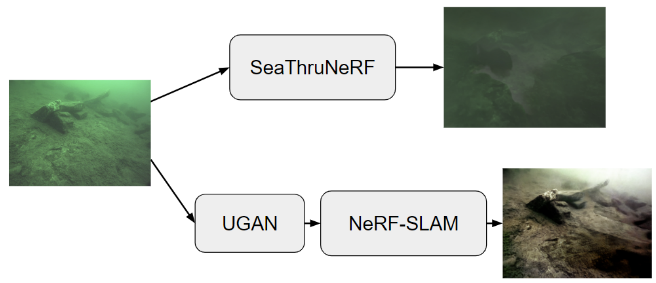

Fig. 1: Flowchart of the design process. We pass an underwater image through SeaThruNeRF for one result and compare that against our design, being UGAN along with NeRF-SLAM.

II. RELATED WORKS

A. NeRF

The primary focus of our research is two cutting-edge NeRF algorithms: NeRF-SLAM [5] and SeaThru NeRF [3]. These algorithms exhibit distinct strengths for underwater NeRF applications. NeRF-SLAM enhances NeRF rendering accuracy through pose estimates and depth maps, incorporating uncertainty from dense monocular SLAM. Conversely, SeaThru NeRF introduces a novel rendering model to mitigate medium effects such as attenuation and backscattering in underwater environments. An example is shown in Fig 2. Our research aims to test both algorithms in large-scale underwater datasets to understand their advantages and limitations.

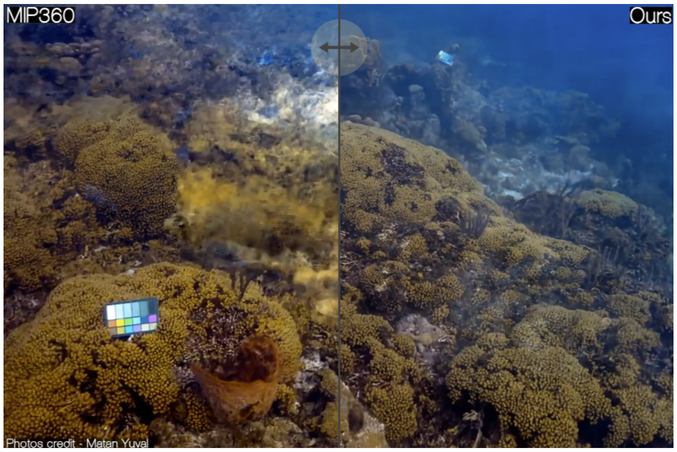

Fig. 2: MP360 results (left) compared with SeaThru-NeRF results (right), adapted from [3]

B. Dataset

1) Underwater Caves Sonar Dataset [6]: This dataset comprises data gathered by an autonomous underwater vehicle (AUV) within the intricate and unstructured confines of an underwater cave complex. The AUV used in this dataset is the Sparus AUV. It has six main sensors attached, including two inertial measurement units, two sonar sensors, one Doppler velocity log, and a downward-facing analog camera. For our purposes, we will be focusing on using the downward-facing analog camera for visuals to train and test our NeRF system. The camera used has a resolution of 384x288 pixels. While this is not the best image quality, it is sufficient for our project's purpose. An example of the image quality is seen in Figure 3.

This dataset also provides a ground truth in the form of traffic cones placed throughout the AUV's path. For our purposes. we can use the placement of these cones and their distances to evaluate how accurately the NeRF-SLAM algorithm correctly determined its position in the world frame.

Fig. 3: Camera view from Sparus AUV, adapted from [6]

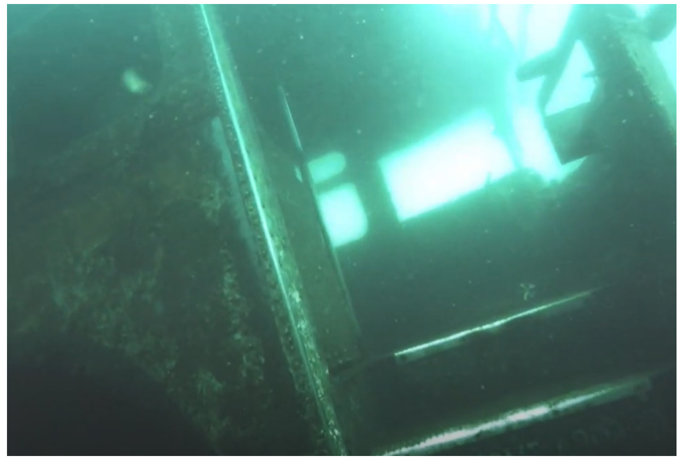

Fig. 4: Monocular image of underwater bus dataset, adapted from [7]

For this dataset, rosbag files contain both information for the camera and the sensor data. One rosbag contains pure camera data of the images with a framerate of 5 fps. It also contains information for the calibration of the camera. The other rosbag contains information on the IMU, Sonar, and DVL data along with each transformation from sensor frame to body frame. This information could be useful if we can expand upon NeRF-SLAM to utilize more sensor data in its SLAM algorithm.

2) Underwater Bus Dataset [7]: Another real-world dataset used in this paper is a monocular video taken by a diver of a bus that has sunk underwater. This dataset is on a smaller scale than the underwater caves, being a video that loops around the bus slowly. Since this dataset is smaller and loops, it will be easier for the NeRF algorithms to render since there will be more frames of specific camera angles for NeRF to be trained on. While this dataset does provide this advantage, there is a lack of ground truth to quantitatively evaluate NeRF algorithms against. A photo of the attenuation can be seen in Figure 4.

3) Underwater Cemetery Dataset [7]: Similar to the underwater bus dataset, the underwater cemetery dataset is a small-scale scene of an underwater environment. This dataset is taken from the Aqua 2 AUV that navigates over a fake cemetery underwater. This dataset has a different attenuation than the underwater bus, so it can provide an edge case for how green the hue of the video is. It also lacks ground truth similar to the underwater bus dataset. An image of the dataset can be seen in Figure 5.

4) Cube-Diorama Dataset [8]: This is a synthetic dataset rendered in Blender specifically for NeRF testing and features various indoor scenes. The rendered scene is shown in Figure 6. This dataset can be used as a baseline for testing the NeRF algorithms, as it is an extremely small scene of a room and should lead to good results. We can also use data from the simulation such as depth maps to further evaluate the results from NeRF.

Fig. 5: Monocular image of underwater cemetery dataset, adapted from [7]

III. Technical Apporach

In this section, we will introduce two methods for underwater scene reconstruction. The first method is to use COLMAP for tracking and SeaThru NeRF [9] for learning the medium parameters and removing the medium effects in the underwater environment. The second method is to preprocess the images with Underwater GAN [10] and then use NeRF-SLAM for tracking and reconstruction.

Furthermore, NeRF-SLAM may face potential challenges due to its design not specifically tailored to underwater environments. To address such challenges, we propose to modify the Cube-Diorama dataset to mimic underwater conditions. This modification involves adding attenuation and backscattering effects to simulate underwater scenes. By evaluating NeRF-SLAM's performance on this adjusted dataset, we aim to understand how underwater scenarios impact the system's rendering accuracy and overall effectiveness. This experimental approach will not only help us to identify any shortcomings of NeRF-SLAM in underwater environments but also explore the possibility of integrating the rendering models from SeaThru NeRF with NeRF-SLAM.

Fig. 6: Cube-Diorama rendered, adapted from [8]

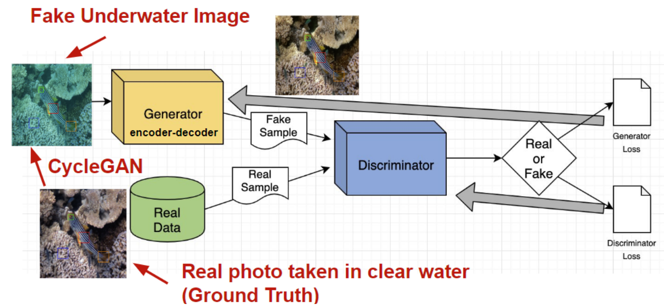

Fig. 7: Deep-learning Architecture of UGAN [10]

A. Underwater Image Enhancement

With the issues discussed previously about attenuation and backscattering, underwater imaging is likely to have a big detriment on NeRF due to the coloring distortion and lack of contrast within images. Not only would it hurt NeRF, but it would also provide issues with accurate depth mapping with DROID-SLAM due to discolorization. With this, this paper wanted to look at another potential way to aid NeRF-SLAM by using image reconstruction during preprocessing to help DROID-SLAM and NeRF correctly work in underwater settings. The method this paper looks at is using a generative adversarial network (GAN) network for underwater image restoration, specifically, Underwater GAN (UGAN) [10].

UGAN works by training a deep-learning generator and a discriminator with clear underwater images. The architecture can be seen in Figure 7. UGAN works by taking a clear underwater image and using CycleGAN to create a false underwater image. The generator then takes the fake underwater image and creates an image restored version of the underwater image. The discriminator takes the restored image and the clear water image and tries to determine which is the generated image and which is the true image. Based on whether the discriminator gets it right or not, the loss updates either the generator or the discriminator. After training, only the generator output is used for inference. UGAN has been shown to provide great imaging results compared to physics-based restoration methods and other GAN methods. Using these corrected images, we can pass them into NeRF-SLAM for SLAM and rendering use.

B. NeRF-SLAM

The core concept of NeRF-SlAM [5] involves supervising a neural radiance field with information generated from the dense monocular SLAM. NeRF-SlAM uses DROID-SLAM [11] for tracking, which is capable of generating dense depth maps and determining camera poses while also offering estimates of uncertainty for both dense depth maps and poses. Utilizing these data allows us to train a radiance field using a depth loss that considers the marginal covariances of the depth maps. Specifically, the mapping loss is defined as the weighted sum of color loss $\mathcal{L}_{\text{rgb}}$ and depth loss $\mathcal{L}_{\text{D}}$:

$$\mathcal{L}_M(T,\Theta) = \mathcal{L}_{\text{rgb}}(T,\Theta) + \lambda_D \mathcal{L}_{\text{D}}(T,\Theta)$$
$$\mathcal{L}_{\text{rgb}}(T,\Theta) = \|I - I^*(T,\Theta)\|^2$$
$$\mathcal{L}_{\text{D}}(T,\Theta) = \|D - D^*(T,\Theta)\|^2_{\Sigma_D}$$

where T is camera poses, Θ is neural network parameters, D, Σ_D are dense depth and uncertainty estimated by DROID-SLAM, and D^* is the rendered depth. Similarly, I is the ground truth image, and I^* is the rendered color image.

Compared to the conventional training equation for NeRF, this approach employs a weighted norm to compute the depth loss, denoted as $\mathcal{L}_{\text{D}}$. Intuitively, a high covariance in Σ_D signals significant uncertainty in the depth information of a particular region. Thus, this area contributes less to the overall mapping loss. Conversely, a low covariance suggests that the depth information is reliable and precise, so this area has a greater influence on the loss calculation. This differential contribution to the loss function enables more precise supervision of NeRF, enhancing the reconstruction quality of areas with lower depth uncertainty.

Real-time performance can be achieved by employing real-time DROID SLAM and radiance field training and executing them concurrently.

C. SeaThru-NeRF

The idea behind SeaThru-NeRF is to extend the NeRF rendering process by considering the effects of light-scattering media such as water or fog [3]. Traditionally, NeRFs do not account for these effects, which can cause many undesirable artifacts and distortions such as floating blobs, . SeaThru-NeRF considers both solid objects and the medium in the scene. Given the camera pose (x, y, z) and the viewing direction (θ, ϕ) for each image, a standard NeRF algorithm will learn the object density $\sigma_i^{obj} \in \mathbb{R}^3$ and the object color $c_i^{obj} \in \mathbb{R}^3$. SeaThru-NeRF additionally learns the backscattering density $\sigma^{bs} \in \mathbb{R}^3$, the attenuation density $\sigma^{attn} \in \mathbb{R}^3$, and the medium color $c^{med} \in \mathbb{R}^3$ in addition to the other parameters learned by traditional NeRFs. The final pixel color is calculated in two parts as follows:

$$\hat{\mathbf{C}} = \sum_{i=1}^{N} \hat{\mathbf{C}}_{\mathbf{i}}^{\mathbf{obj}}(\mathbf{r}) + \sum_{i=1}^{N} \hat{\mathbf{C}}_{\mathbf{i}}^{\mathbf{med}}(\mathbf{r})$$

The two color contributions from the object and the medium are calculated as follows:

$$\hat{\mathbf{C}}_{\mathbf{i}}^{\mathbf{obj}}(\mathbf{r}) = T_i^{obj} \cdot exp\left(-\sigma^{attn} t_i\right) \cdot \left(1 - exp\left(-\sigma_i^{obj}\delta_i\right)\right) \cdot \mathbf{c}_i^{med}$$

$$\hat{\mathbf{C}}_{\mathbf{i}}^{\mathbf{med}}(\mathbf{r}) = T_i^{obj} \cdot exp\left(-\sigma^{bs} t_i\right) \cdot \left(1 - exp\left(-\sigma^{bs}\delta_i\right)\right) \cdot \mathbf{c}^{med}$$

$$\text{where } T_i^{obj} = exp\left(-\sum_{j=0}^{i-1} \sigma_j^{obj}\delta_j\right)$$

Fig. 8: Cube-Diorama rendered in an underwater setting

These additional parameters allow SeaThruNeRF to calculate the contributions of both the medium and the object on the final pixel color. The distinction between the medium effects and the physical objects in the scene allows rendering of the scene in multiple ways. The scene can be rendered with all lighting effects (RGB), clear as if with no medium (J), the direct attenuated light signal, only the backscattering signals, a depth map, and more. This allows the medium to be removed to render the objects as if they were in clear air, which can be very helpful for analyzing the scene.

D. Underwater Synthetic Dataset

We used available functions in Blender to create an underwater version of the Cube-Diorama Dataset as seen in Figure 8. This underwater rendering was done using a volumetric shader to create the blue hue along with a volumetric absorption to create the attention effect of light as distance increases. We also implemented floating particles into the scene to give the render dynamic objects to test how well NeRF handles these. Overall, these shaders do qualitatively create an underwater setting, and this should provide a good simulation test case for evaluating the different NeRF algorithms. An example image of the underwater simulated environment can be seen in Figure 8.

IV. Results

We conduct a comprehensive evaluation of Underwater GAN, NeRF-SLAM, and SeaThru NeRF across various datasets. These results are vital for assessing the potential limitations and scalability of these algorithms in large-scale underwater environments.

A. Underwater GAN

For this paper, we trained UGAN for 300 epochs on an NVIDIA RTX 4090 and the training data included 6128 clearwater images and their respective generated underwater images. Running the four datasets through UGAN gives the

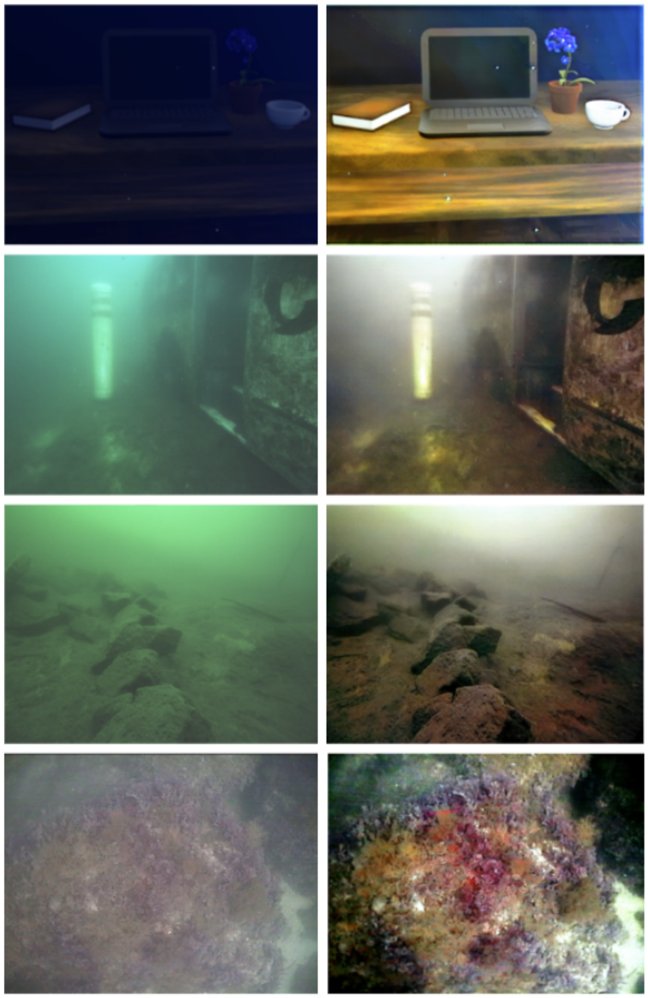

Fig. 9: UGAN results from each dataset.

results seen in Figure 9. Overall, UGAN does a great job at correcting the image color and contrast, however, has some notable issues with the output. One key issue is the resolution of the output is only 512x512 pixels, while the input images from each dataset vary. This means the images are resized to a lower resolution causing pixelation in the output images. Another key issue is that UGAN does not consider temporal information, meaning that color correction for each image within a video could be slightly different than the previous image in the sequence. This causes some slight lighting and color changes throughout the output image sequence. Even with these issues, overall the videos are much clearer. These image sequences will be passed into NeRF-SLAM and evaluated against the NeRF-SLAM without UGAN image correction.

B. NeRF-SLAM

This section delves into the evaluation of NeRF-SLAM's reconstruction capabilities across various datasets. The results are shown in columns d and e of Fig. 10.

We tested NeRF-SLAM using an RTX 3080TI and found that its tracking module, DROID-SLAM, requires over 11GB of VRAM. Therefore, for larger underwater datasets, we use a server equipped with an NVIDIA Quadro RTX 8000, which offers 48GB of VRAM, for training.

1) Cube-Diorama Dataset: Results in 1d of Fig. 10 shows the NeRF-SLAM rendered image obtained from the original Cube-Diorama dataset. The fidelity of the reconstruction to the actual environment (1a of Fig. 10) is notably high, showcasing NeRF-SLAM's ability to achieve high-quality reconstructions in standard NeRF settings.

The rendered image from the underwater synthetic version of the Cube-Diorama dataset is illustrated in 1e. The rendered image appears blurry, with noticeable translucence effects observed on the laptop at certain viewing angles. Furthermore, object edges are indistinct.

2) Underwater Datasets: 3d and 3d showcase the rendered image derived from the Underwater Caves Sonar Dataset. Despite the inherent blurriness due to the low resolution of the training images, the rendered image can display the seafloor details relatively well compared with ground truth (Figure **??**). However, the medium effect results in low-contrast and blurry reconstructions.

5d and 6d display the results for the Underwater Cemetery Dataset. The seafloor in 5d appears clear, whereas the image in 6d is very blurry, making it impossible to identify the stones. A similar issue occurs in 7d, where numerous floaters obscure the bus. These issues arise primarily from two factors. First, the DROID-SLAM neural network is trained with in-air datasets. So, it produces poor depth maps and covariance estimates in underwater settings and thus has a negative effect on NeRF training. Second, the instant-NGP module in NeRF-SLAM struggles with reconstructing large scenes. While the camera in 5d is near the central area, the cameras in 6d and 7d are positioned far from the start, causing NeRF to miss detailed features in these distant areas.

3) Preprocess with Underwater GAN: The combination of UGAN and NeRF-SLAM sometimes delivers the best results among all tested methods, as shown in 2e and 5e. UGAN effectively restores the original color of the scene and mitigates medium effects. This shows the potential of preprocessing underwater datasets prior to NeRF training. However, in 3e, 4e, 6e, and 7e, despite color restoration, the rendered images lack detail. For instance, the red cone in 4e is much harder to identify compared to 4d. This issue stems from the inconsistent output of Underwater GAN. This problem can be seen in 3d and 4e. In 3d, the bottom right corner is darker due to attenuation, whereas in 4e, both the top right and bottom right corners appear dark.

C. SeaThru-NeRF

This section details the results of the SeaThru-NeRF-lite algorithm for various datasets, including strengths and weaknesses of the algorithm. The discussion will focus on the Cube-Diorama dataset [8], our underwater synthetic dataset, and the Underwater Caves Vision and Sonar dataset [6]. Figure 11

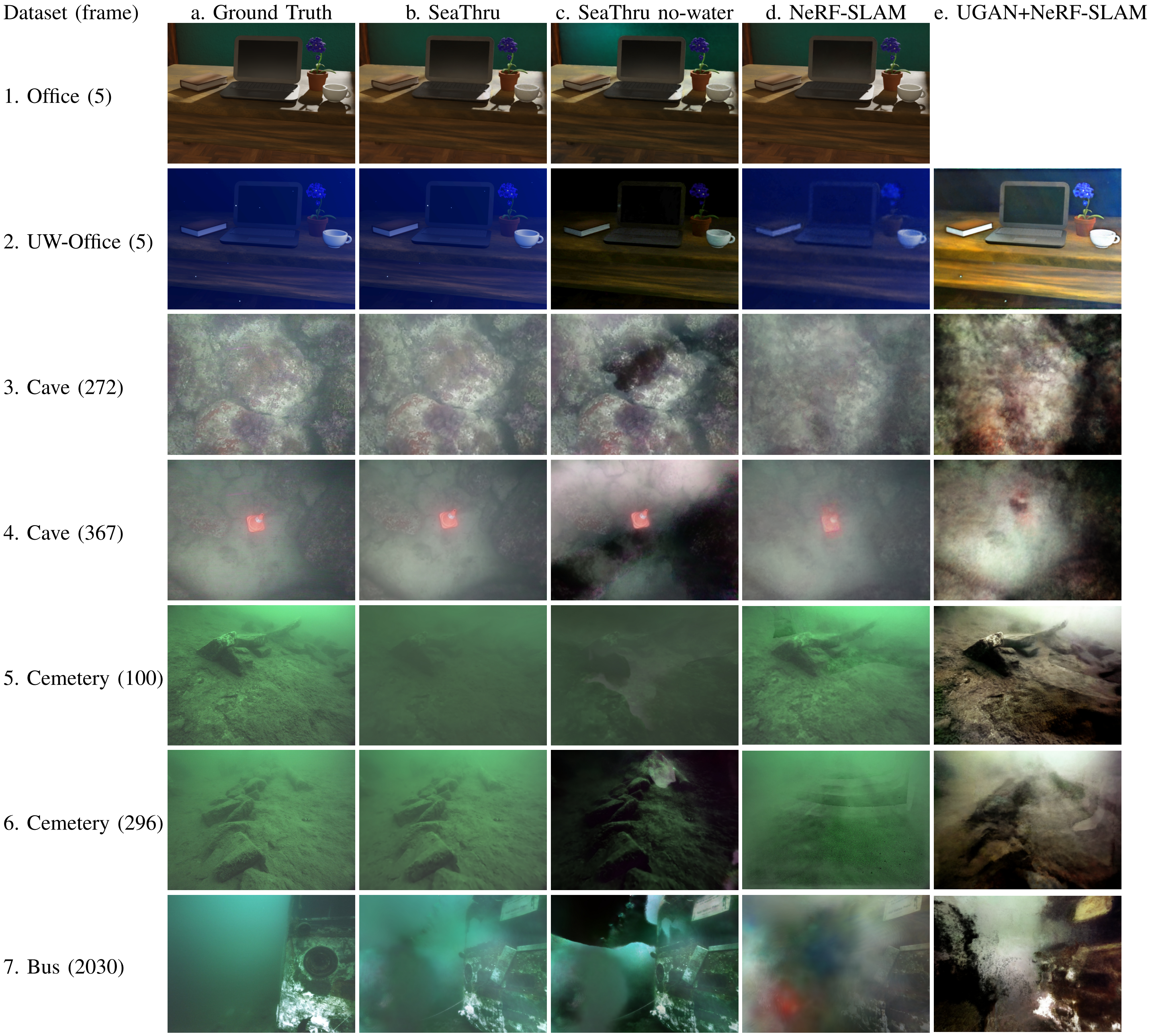

Fig. 10: NeRF reconstruction results from different methods on (row 1) original Cube-Diorama Dataset office scene [8], (row 2) underwater synthetic office scene, (row 3,4) Underwater Caves Sonar Dataset [6], (row 5,6) Underwater Cemetery Dataset [7], and (row 7) Underwater Bus Dataset [7]. Brightness in 2a-2d is increased by 400% to improve readability. UGAN is not applicable to the in-air office scene, so there are no results for 1e.

compares a) the ground truth testing image, b) the RGB render with the medium, c) the no-medium render showing only the object, and d) the estimated depth for various frames from the datasets.

The SeaThru-NeRF scenes were trained in nerfstudio [12] using the SeaThru-NeRF-lite model. Unfortunately due to computing constraints we were only able to run the smaller of the two provided models which used less GPU memory - approximately 7GB instead of 23Gb. The models trained on a Nvidia GeForce GTX 1650 GPU for 30,000 iterations with a batch size of 2048 rays.

1) Cube-Diorama Dataset: As seen in figure 11, SeaThru-NeRF was able to reconstruct the office scene quite well. The ordinary office scene has no medium, so it is expected that the RGB and no-medium renders are the same. However, comparing the RGB (1b) and no-medium (1c) renderings of the office scene from figure 11, some of the coloration is different, especially visible on the back wall and table.

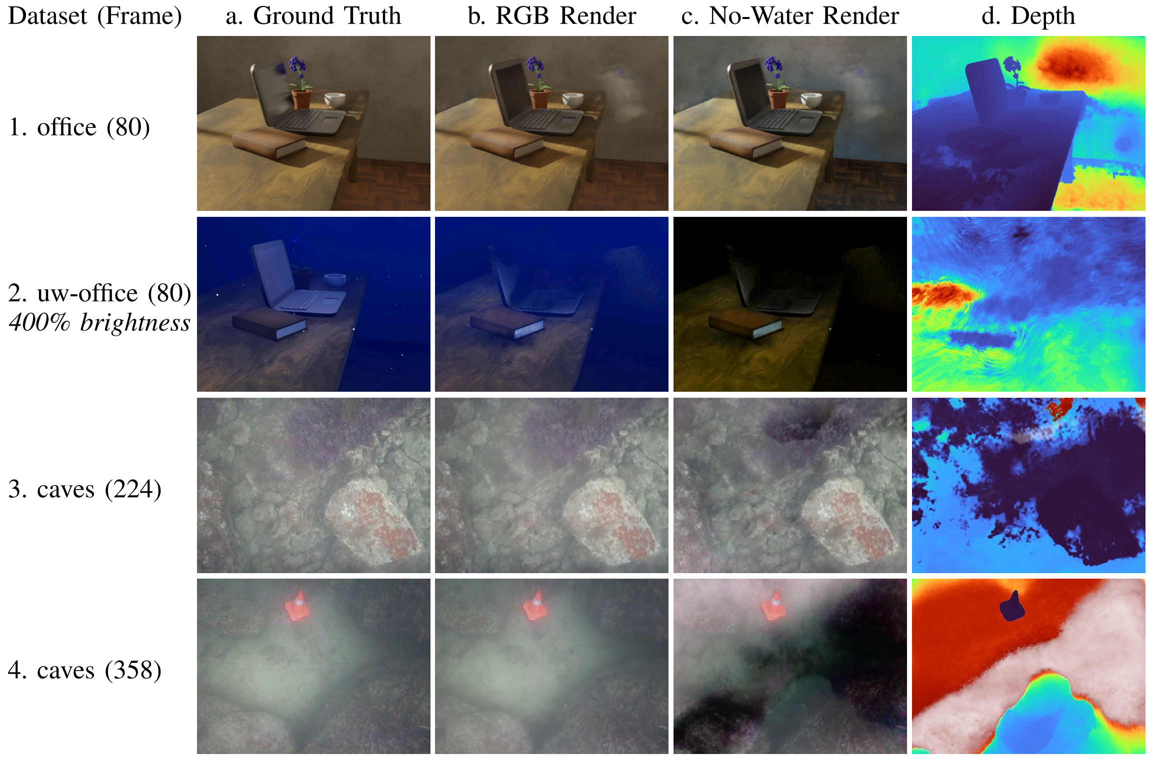

Fig. 11: SeaThr-NeRF reconstruction results of novel views from different datasets. Each frame is a novel view taken from the test dataset. Row 1 is from the original Cube-Diorama Dataset office scene [8], row 2 is the underwater synthetic office scene, and rows 3 and 4 are from the Underwater Caves Sonar and Vision dataset [6].

Additionally, the model did not capture the reflections off the laptop screen, and introduced extra lighting on the back wall. This may be due to using the smaller SeaThru-NeRF-lite model, which may not be fully capturing the scene as well as the larger model. However these are smaller details, and SeaThru-NeRF does a very good job of recreating the overall structure of the scene.

2) Underwater Synthetic Cube-Diorama Dataset: As seen in figure 11 row 2, SeaThru-NeRF struggled recreating the underwater scene. The closer details are visible in the renderings, but details further from the camera are not captured by the NeRF model. SeaThru-NeRF notably is able to remove the synthetically added medium as seen in 2c. Looking at the depth map, however, shows that SeaThru-NeRF did not learn the overall 3D structure of the scene.

One nuance of SeaThru-NeRF [3] and NeRFs in general to note is the assumption that a sparse recreation of the camera poses for each image are known before training [2]. The NeRF model can adjust the camera poses as necessary during training to minimize the loss function, but is unable to find these starting camera poses on its own. Therefore a structure-from-motion algorithm, such as COLMAP, is a necessary preprocessing step before the NeRF model can be trained. This can present challenges when this algorithm fails. This happened several times during training, including on the Synthetic Underwater Cube-Diorama Dataset. In order to train the model, the camera poses from the original non-underwater dataset were used. On other datasets, like the Caves, Bus, and Cemetery datasets, COLMAP only found poses for some of the images, resulting in trained models that didn't have information about parts of the scene.

3) Underwater Caves Vision and Sonar Dataset: SeaThru-NeRF did quite well at recreating scenes from the Underwater Caves dataset. As seen in figure 11 rows 3 and 4, the RGB renderings look very good and retain a lot of the details from the test images. Textures are preserved pretty well, such as the red on the rock and the purple plant in frame 224. In frame 358, the cone is very clear in the image. Overall the RGB renders look very good. However, looking at the no-medium render, some of the seafloor was incorrectly modeled as medium by SeaThru-NeRF. Additionally, the model does not seem to have learned the depth super well, as the rocks in frame 224 are not visible on the depth map, and in frame 358 the cone seems to be either very far or very near to the camera.

V. Conclusion & Further Work

A. Image Reconstruction for NeRF

While UGAN provided better image quality for NeRF use, the inconsistency with the temporal information seemed to detract from the SLAM and NeRF capabilities of NeRF-SLAM. While it seemed like the additional contrast for the depth detection would help, the inconsistency of the lighting and contrast makes it harder for DROID-SLAM to correctly predict the depth of each image. This led to a lot of inconsistent results for NeRF, where some render angles appear clean with others being extremely distorted. With this, further work could be done to implement a GAN approach to image reconstruction that considers temporal information to keep image reconstruction consistent across an entire video captured. Along with this, creating a GAN network that can account for different image resolutions could help keep image quality consistent across different datasets.

B. NeRF-SLAM

Although NeRF-SLAM performs well in standard NeRF settings and small underwater scenes, significant improvements are required for general underwater applications. Additionally, its runtime performance does not meet the criteria for a 'real-time system'. Using an Nvidia RTX 3080ti, we achieve only 3 fps for tracking and fewer than 10 fps for 720p rendering.

C. SeaThru-NeRF

SeaThru-NeRF performs quite well in small underwater scenes. The medium model allows the NeRF to effectively learn underwater scenes. Despite the good performance in smalls scenes, transfering to larger scenes, especially with no are of focus in the scene proved difficult. One of the biggest bottlenecks for SeaThru-NeRF was the structure-from-motion algorithm used to provide initial camera poses. This often fully or partially failed, which meant SeaThru-NeRF did not have that data to train on. Further work on feature extraction and SFM for scenes with light scattering media could help to make SeaThru-NeRF more robust in general underwater scenes. Especially in the application of AUVs, perhaps assumptions of frames being sequential could be leveraged in the offline recreation of the scene from AUV video data.

REFERENCES

[1] B. Joshi, M. Xanthidis, M. Roznere, N. J. Burgdorfer, P. Mordohai, A. Q. Li, and I. Rekleitis, "Underwater exploration and mapping," in *2022 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)*, p. 1–7, Sept. 2022.

[2] 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 *Computer Vision – ECCV 2020* (A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, eds.), Lecture Notes in Computer Science, (Cham), pp. 405–421, Springer International Publishing, 2020.

[3] D. Levy, A. Peleg, N. Pearl, D. Rosenbaum, D. Akkaynak, S. Korman, and T. Treibitz, "SeaThru-NeRF: Neural Radiance Fields in Scattering Media," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 56–65, 2023.

[4] B. Parui, Y. Devada, and K. Surender, "Vision based 3d mapping-from traditional to nerf based approaches," in *2023 2nd International Conference on Paradigm Shifts in Communications Embedded Systems, Machine Learning and Signal Processing (PCEMS)*, p. 1–6, Apr. 2023.

[5] A. Rosinol, J. J. Leonard, and L. Carlone, "NeRF-SLAM: Real-Time Dense Monocular SLAM with Neural Radiance Fields," Oct. 2022.

[6] "Underwater caves sonar data set - Angelos Mallios, Eduard Vidal, Ricard Campos, Marc Carreras, 2017." https://journals-sagepub-com.proxy.lib.umich.edu/doi/10.1177/0278364917732838.

[7] S. Rahman, A. Q. Li, and I. Rekleitis, "Svin2: An underwater slam system using sonar, visual, inertial, and depth sensor," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, p. 1861–1868, Nov. 2019. arXiv:1810.03200 [cs].

[8] J. Abou-Chakra, F. Dayoub, and N. Sünderhauf, "Implicit object mapping with noisy data," *arXiv preprint arXiv:2204.10516*, 2022.

[9] A. V. Sethuraman, M. S. Ramanagopal, and K. A. Skinner, "WaterNeRF: Neural Radiance Fields for Underwater Scenes," in *OCEANS 2023 - MTS/IEEE U.S. Gulf Coast*, pp. 1–7, Sept. 2023.

[10] C. Fabbri, M. J. Islam, and J. Sattar, "Enhancing Underwater Imagery Using Generative Adversarial Networks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7159–7165, May 2018.

[11] Z. Teed and J. Deng, "DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras," Feb. 2022.

[12] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, J. Kerr, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. McAllister, and A. Kanazawa, "Nerfstudio: A modular framework for neural radiance field development," in *ACM SIGGRAPH 2023 Conference Proceedings*, SIGGRAPH '23, 2023.

APPENDIX

A. Code

Code for UGAN is available at https://github.com/Zxl19990529/UGAN-pytorch.

Code for NeRF-SLAM is available at https://github.com/tccoin/NeRF-SLAM.

Code for SeaThru-NeRF is available at https://github.com/nerfstudio-project/nerfstudio.