UW-3DGS: Unveiling Underwater Scenes and Scattering Media through 3D Gaussian Splatting

Anonymous Author(s)
Submission Id: 496

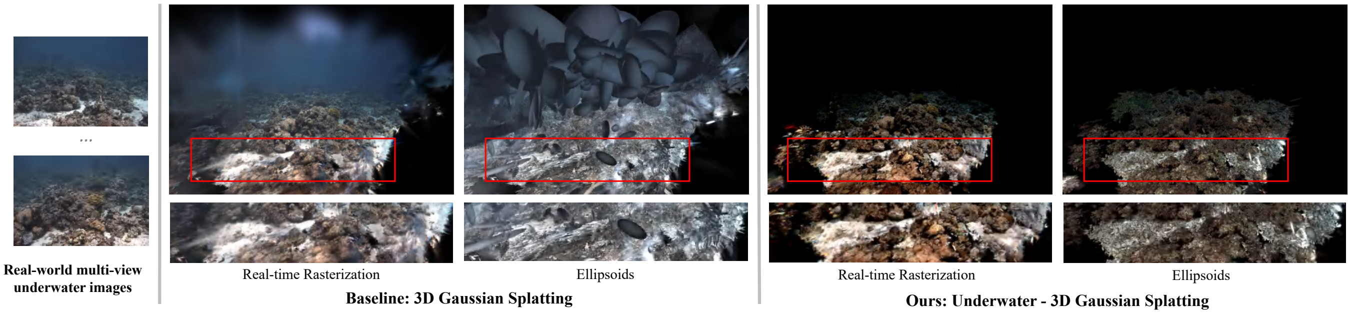

Real-world multi-view underwater images
...
Real-time Rasterization
Ellipsoids
Baseline: 3D Gaussian Splatting
Real-time Rasterization
Ellipsoids
Ours: Underwater - 3D Gaussian Splatting

Figure 1: Employing 3DGS[17] directly on underwater imagery leads to messy ellipsoidal distributions. Our novel approach, UW-3DGS, enhances this process, achieving precise and detailed reconstructions of the ocean's seabed, free from the messy ellipsoids.

ABSTRACT

Neural Radiance Fields (NeRF) have set new benchmarks in reconstructing scenes from multi-view images. However, its application to scenes with scattering media like water or fog has been underexplored. Image formation with scattering media is difficult since it requires physical attributes, such as wavelength to be inversely inferred, which makes 3D reconstruction from images affected by scattering media an ill-posed problem. Though scattering media's continuous nature aligns well with NeRF's volume rendering nature, enable outputting per-point physical attributes but at a cost on computational efficiency. In order to improve the efficiency of existing NeRF-based methods modeling scattering media, we integrate the latest in NeRF technology, 3D Gaussian Splatting (3DGS), with the image formation model in scattering media to efficiently approximate the physical attributes, and reconstruct clear 3D scenes. Our method is validated on real underwater scenes, demonstrating its ability to render novel views of underwater scenes and produce clear images as though in the absence of water. It outperforms existing methods in efficiency, quality, and simplicity, offering a more effective solution to underwater scene reconstruction. The anonymous project page is available at https://uw3dgs.github.io/.

CCS CONCEPTS

• **Computing methodologies → Volumetric models**; *Image and video acquisition*; *Hierarchical representations*.

KEYWORDS

neural radiance fields; novel view synthesis; underwater scene reconstruction.

ACM Reference Format:
Anonymous Author(s). 2024. UW-3DGS: Unveiling Underwater Scenes and Scattering Media through 3D Gaussian Splatting. In *Proceedings of the 32th ACM International Conference on Multimedia (MM '24), October 10–14, 2024, Melbourne, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/

1 INTRODUCTION

3D reconstruction is pivotal for creating the Metaverse and scene simulators, essential for advancing autonomous driving vehicles and robotics. However, 3D scanning relies on specialized equipment, making 3D reconstruction from multi-view images a crucial research field within computer vision.

Neural Radiance Fields (NeRF) [26] has recently emerged as a groundbreaking approach in this area, distinguished by its ability to reconstruct scenes with intricate geometrical details and vibrant colors from multi-view images. It conceptualizes the scene as a continuous volumetric form represented through Multilayer Perceptron Networks (MLPs). Nonetheless, NeRF and its subsequent variations [11, 22, 43] operate under the assumption that the captured multi-view images are clear and of high quality, disregarding any participating media in the air that significantly impacts light transmission between objects and the camera. Essentially, it is presumed that the color of the final pixel solely represents the radiance from the surface intersected by the light ray.

Unlike in clear-air environments, foggy weather and underwater scenes introduce media that both absorb and scatter light as it traverses through the space/volume. The intensity of these absorbing and scattering effects varies across different spatial points, influenced by several complex physical factors that are challenging to quantify. NeRF models the scene as a continuous volumetric

MM '24, October 10–14, 2024, Melbourne, Australia

ACM ISBN . . . $15.00
https://doi.org/

form, outputting radiance (color and density) for each spatial point. Utilizing the volume rendering approach, only visible points with radiance contribute to the final pixel color. The volumetric representation of NeRF is inherently suitable for scenes featuring spatially varying media. Consequently, SeaThru-NeRF [20] leverages NeRF's volumetric forms for the reconstruction and rendering of underwater scenes, incorporating the scattering media (water) for each point along the line of sight.

However, volume rendering necessitates hundreds of queries, and ray casting exhibits extreme inefficiency when querying MLPs. Consequently, SeaThru-NeRF exhibits significant inefficiency in both the training and rendering phases. This inefficiency markedly limits its real-world applicability, necessitating further post-processing steps to transform the radiance field volume into a format compatible with computer graphics, or requiring the deployment of a specifically designed renderer [47], which might lead to further degradation in rendering quality. In contrast, our methodology is built upon 3D Gaussian Splatting (3DGS) [17], offering an efficient pipeline for both training and real-time rendering. 3DGS models the scene using a collection of 3D Gaussians of variable shapes and rasterizes them using an efficient *splatting* technique. Directly applying 3DGS to model underwater scenes could result in a proliferation of floating 3D Gaussians in the water, aimed at simulating attenuation and scattering effects. Although this approach results in higher rendering quality, the emergence of these floating 3D Gaussians is highly undesirable for underwater applications, complicating the use of the reconstructed 3D scene for navigation and collision detection due to these floating entities.

Our method can reconstruct a cleaner 3D Gaussian scene with significantly fewer floating 3D Gaussians to represent the underwater environment, achieving high-quality novel view rendering. This is achieved by decoupling the 3D Gaussian rendering from the scattering media effects. Initially, a clear image devoid of water is rendered using 3D Gaussians. Then, the underwater effects are integrated using the revised image formation model [1], which accommodates the scattering media. This model employs two wavelength-dependent coefficients and one backscatter color to represent the media. We simplify these three coefficients' dependencies and optimize them as directly learnable parameters. In contrast to SeaThru-NeRF, which models the wavelength-dependent coefficients as functions of viewing angles, our approach optimizes these three coefficients independently, applying them universally across all pixels. This method is more direct and efficient for both training and rendering. To further eliminate the scattering media, we introduce a novel Self-Pruning Supervision Loss to achieve a clear reconstruction of the seabed. The experiment demonstrates that our approach can accurately learn both the 3D underwater scene and the scattering media, thereby producing high-quality renderings of underwater scenes, either with or without scattering media. The contributions of UW-3DGS are summarized as follows:

1. UW-3DGS represents the pioneering method that integrates the image formation model with 3DGS for the reconstruction and color restoration of underwater scenes.
2. UW-3DGS can achieve state-of-the-art underwater rendering quality and efficiency on real-world underwater scenes.
3. The introduction of the Self-Pruning Supervision Loss effectively eliminates floating noisy Gaussians, enabling the clear reconstruction of the seabed.

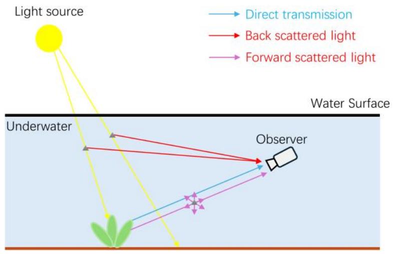


Figure 2: Light propagation underwater.

2 RELATED WORK

2.1 Neural Radiance Fields (NeRF)

NeRF [26] has achieved high-quality outcomes in 3D scene reconstruction and novel view rendering. Variants of NeRF have been developed to enhance its performance and extend its applicability across different domains. Regarding rendering efficiency, FastNeRF [12] stores a deep radiance map for each point, achieving speeds of up to 200 FPS. KiloNeRF [31] depicts the scene using thousands of tiny MLPs, reaching speeds about 2000 times faster than the original NeRF. Some approaches aim to sample points closer to surfaces to minimize ineffective sampling, thereby enhancing efficiency. For example, EfficientNeRF [13] pioneers the use of valid and pivotal sampling strategies. DONeRF [28] utilizes a depth oracle network to determine the optimal sample locations along each ray. In a similar vein, DVGO [40] and NSVF [21] create sampling masks by assessing density values.

Beyond enhancements in sampling, voxel-based representations surpass coordinate-based MLPs in efficiency. DVGO [40] and NSVF [21] encapsulate radiance field features within voxel grids, creating a continuous feature field via interpolation. To enhance data compactness, Instant-NGP [27] encodes radiance field features by accessing a feature table with hash indexes. Meanwhile, TensoRF [7] employs tensor decomposition to closely approximate the intricate details of high-order radiance field feature voxels.

To manage large-scale outdoor scenes extending to infinity, NeRF++ [49] devises a warping space for distant background points. MIP-NeRF 360 [4] employs a proposal network for sampling distant content. For generalized inference, PixelNeRF [48], IBRNet [44], and MVSNeRF [8] compute radiance by analyzing and comparing features across multi-view pixels. To make NeRF output compatible with graphics engines, Nerfactor [51] and Nvdiffrec [14] have adapted NeRF for inverse rendering. While the original NeRF encountered challenges related to efficiency, editability, and scalability, the introduction of 3DGS [17] has effectively tackled these issues through its point-based pipeline.

NeRF has been extended to applications beyond view synthesis, such as NeRF-SLAM (Simultaneous Localization and Mapping) [32], GS-SLAM [45], and robotics for object grasping [18]. It is also utilized in 2D image tasks, including image denoising [30], de-blurring [23], super-resolution [42], and low-light enhancement [24]. Our

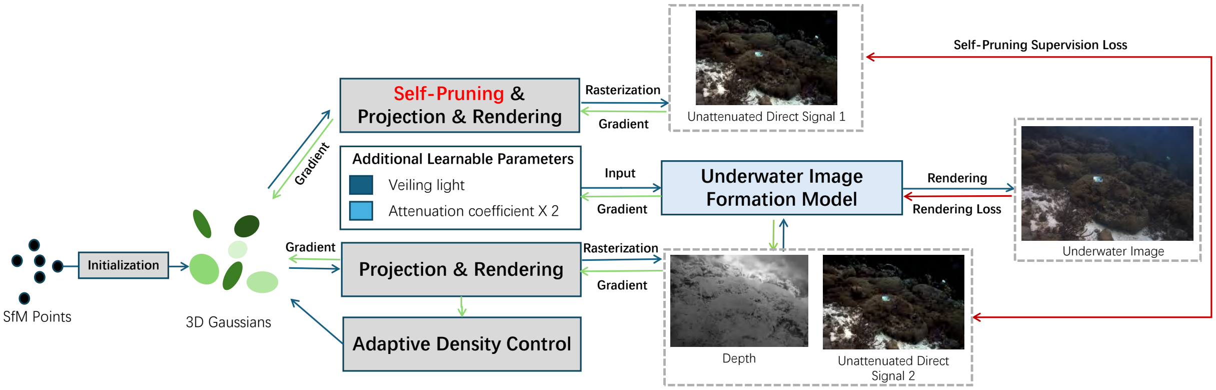

Self-Pruning & Projection & Rendering
Rasterization
Gradient
Unattenuated Direct Signal 1
Self-Pruning Supervision Loss
Gradient
Additional Learnable Parameters
■ Veiling light
■ Attenuation coefficient X 2
Input
Gradient
Underwater Image Formation Model
Rendering
Rendering Loss
Underwater Image
Initialization
SfM Points
3D Gaussians
Gradient
Projection & Rendering
Rasterization
Gradient
Adaptive Density Control
Depth
Unattenuated Direct Signal 2

Figure 3: Overall pipeline. UW-3DGS initializes 3D Gaussians using sparse point clouds reconstructed by the Structure-from-Motion (SfM) software, COLMAP [36]. The unattenuated signals, or direct radiance from objects, are directly rasterized from the 3D Gaussians, with depth information also obtainable. Additional learnable parameters, such as veiling light and attenuation coefficients, are initialized and incorporated into the image formation model alongside depth and unattenuated signals to render underwater images. Furthermore, a Self-Pruning branch is introduced to generate clearer unattenuated signals, leading to a Self-Pruning Supervision loss that enhances the cleanliness of the 3D Gaussians.

work is particularly related to underwater NeRF methods. SeaThru-NeRF [20] integrates scattering media with the volume rendering approach, achieving high reconstruction quality. Similarly, Water-NeRF [37] enhances the original NeRF with a physics-based image formation model for underwater scene learning. WaterHE-NeRF [52] employs histogram equalization as pseudo-ground truth for supervision. Yet, both WaterNeRF and WaterHE-NeRF have limited their experiments to water tanks, not extending to real-world ocean environments. Our method, in contrast, is tested in the most challenging real-world oceanic conditions, showcasing its robustness and applicability. Additionally, their MLP-based network backbone is notably slow in training and rendering.

2.2 Light Propagation in Scattering Media

Rendering realistic images in scattering media necessitates a physically accurate simulation of light propagation. Light interacts with particles within the media, making the computation of global illumination challenging. Early methods, as noted by Blinn et al. [6], primarily focused on single scattering suitable for optically thin media. However, thick media demand accounting for multiple scatterings, which can be simulated using finite element methods, Monte Carlo techniques, and point collocation. Although Monte Carlo methods offer versatility, they typically require extended computation times. To enhance efficiency and reduce noise, Jensen et al. [16] proposed the utilization of photon maps. Novak et al. [29] provided a comprehensive review of Monte Carlo methods for the simulation of volumetric light transport.

For underwater light propagation, the SeaThru models [1–3] have investigated medium parameters that exhibit strong wavelength dependency in underwater scenes. In the realm of monocular methods in computer vision, Yang et al. [46] have provided a comprehensive overview of underwater image restoration and enhancement. With the advent of deep learning, Sharma et al. [38] extensively discussed single-image defogging techniques based on Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Generative Adversarial Networks (GANs).

3D reconstruction within scattering media presents an ill-posed problem. Dehaze-NeRF [9] extends the volume rendering approach of NeRF to accommodate a haze image formation model. But previous methods [9, 20, 37, 52] applying NeRF to scattering scenarios have adopted the volume rendering approach, which results in slow training and rendering. Our method, built upon Gaussian Splatting, offers a much more efficient and simpler solution.

3 PRELIMINARIES

3.1 3D Gaussian Splatting (3DGS)

Our framework utilizes 3D Gaussian Splatting (3DGS) [17], representing the scene through a set of anisotropic Gaussians, $\mathcal{G}$, rendered efficiently via a splatting-style rasterization that effectively bypasses empty spaces. Analogous to NeRF's radiance $(\mathbf{c}, \sigma)$, each Gaussian $\mathcal{G}^i$ encapsulates color c^i and opacity α^i. The Gaussian's position and its ellipsoidal shape are defined by the mean $\boldsymbol{\mu}_W^i$ and covariance Σ_W^i in world space. To capture view-dependent radiance effects, color attributes are modeled using spherical harmonics.

During the rasterization process, 3D Gaussians, denoted by $\mathcal{G}(\boldsymbol{\mu}_W, \Sigma_W)$, are transformed into 2D Gaussians on the image plane, symbolized by $\mathcal{G}(\boldsymbol{\mu}_I, \Sigma_I)$, through the application of a projective transformation:

$$\boldsymbol{\mu}_I = \pi(\boldsymbol{T}_{CW} \cdot \boldsymbol{\mu}_W)\,, \Sigma_I = \mathbf{J}\mathbf{W}\Sigma_W\mathbf{W}^T\mathbf{J}^T\,, \qquad (1)$$

where π is the projection operation and $\boldsymbol{T}_{CW} \in \boldsymbol{SE}(3)$ is the view's camera pose. $\mathbf{J}$ is the Jacobian of the affine approximation of the projective transformation and $\mathbf{W}$ is the viewing transformation of $\boldsymbol{T}_{CW}$. The covariance matrix Σ_W of a 3D Gaussian is described as a scaling matrix S and rotation matrix R:

$$\Sigma_W = RSS^{\mathsf{T}}R^{\mathsf{T}}. \qquad (2)$$

The pixel's color, denoted by $\hat{C}$, is rendered through a process of splatting and blending across N Gaussians, as captured by the

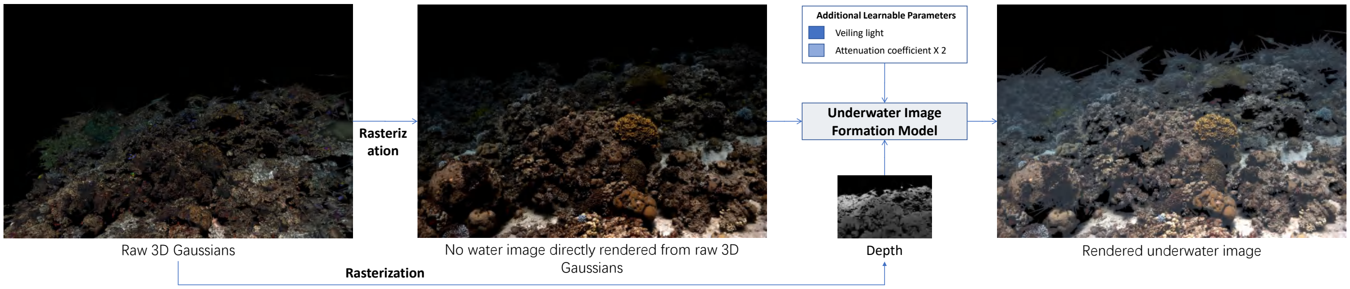

Raw 3D Gaussians
Rasteriz
ation
No water image directly rendered from raw 3D Gaussians
Additional Learnable Parameters
Veiling light
Attenuation coefficient X 2
Underwater Image Formation Model
Depth
Rendered underwater image
Rasterization

Figure 4: Visualizing the transformation: applying the image formation model to raw 3D Gaussian from "Ours w/ default Self-Pruning".

following equation:

$$\hat{C} = \sum_{i \in N} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j). \tag{3}$$

This rendering equation effectively accumulates the contributions of each Gaussian's color c_i, modulated by its opacity α_i, and adjusts for the accumulated transmittance from preceding Gaussians. This rendering formulation is fully differentiable, enabling the use of gradient descent techniques, such as Adam [19], to iteratively refine both the optical and geometric parameters of the scene for enhanced scene representation and image fidelity.

3.2 Underwater Image Formation Model

Figure 2 illustrates the trajectory of light as it travels from underwater towards the observation camera. The underwater image formation model considers each RGB color channel, denoted by c. The intensity of a pixel, $I_c(x)$, at position x is determined by two components: the attenuated direct signal D_c and the backscattered light B_c [15, 34]:

$$I_c(x) = D_c(x) + B_c(x). \tag{4}$$

D_c represents the direct radiance from the object (indicated by the blue arrow in Figure 2), which undergoes attenuation due to absorption and scattering as light travels and encounters particles. This attenuation is a function of both distance and wavelength, with longer wavelengths, such as red, attenuating more rapidly than shorter wavelengths. B_c, the backscattered light (also referred to as veiling light or path radiance, shown by the red arrow in Figure 2), solely diminishes the color and contrast of the light without transmitting any radiance from the object. The forward-scattered light F (depicted by the purple arrow in Figure 2) represents light reflected from the object and deviating from the observer's line of sight. The impact of image degradation by F is considerably less significant than that by backscattered light B_c [10, 33, 35, 41]; hence, it is typically disregarded in analyses.

The contemporary underwater image formation model utilizes the wide-band attenuation coefficient β^D to encapsulate the cumulative effect of absorption and scattering:

$$I_c = \overbrace{\underbrace{J_c}_{\text{albedo color}} \cdot \underbrace{(e^{-\beta^D \cdot z})}_{\text{attenuation}}}^{\textit{direct}} + \overbrace{\underbrace{B_c^\infty}_{\text{veiling light}} \cdot \underbrace{(1 - e^{-\beta^D \cdot z})}_{\text{accumulation}}}^{\textit{backscatter}} \tag{5}$$

where z is the range along the line of sight, J_c is the unattenuated radiance, and B_c^∞ is the wide-band veiling light, respectively,

Akkaynak et al. [1] demonstrated that the wide-band attenuation coefficients employed in Equation (5) vary in their dependencies. Consequently, a revised image formation model is introduced:

$$I_c = J_c \cdot (e^{-\beta^D(\mathbf{v}_D) \cdot z}) + B_c^\infty \cdot (1 - e^{-\beta^B(\mathbf{v}_B) \cdot z}) \tag{6}$$

where the vectors $\mathbf{v}_D$ and $\mathbf{v}_B$ denote the dependencies of β^D and β^B on factors such as range, sensor response, ambient light, etc. While Equation (5) offers a simplified perspective by equating β^D to β^B, it sacrifices accuracy.

On the other hand, reducing the number of unknown parameters in Equation (6) facilitates its solution. Commonly, the values of B_c^∞ are presumed uniform; however, Bekerman et al. [5] revealed non-uniformity in these values, attributable to various influences. SeaThru [2] determined that β^B could be considered constant, whereas β^D is influenced by distance and reflectance.

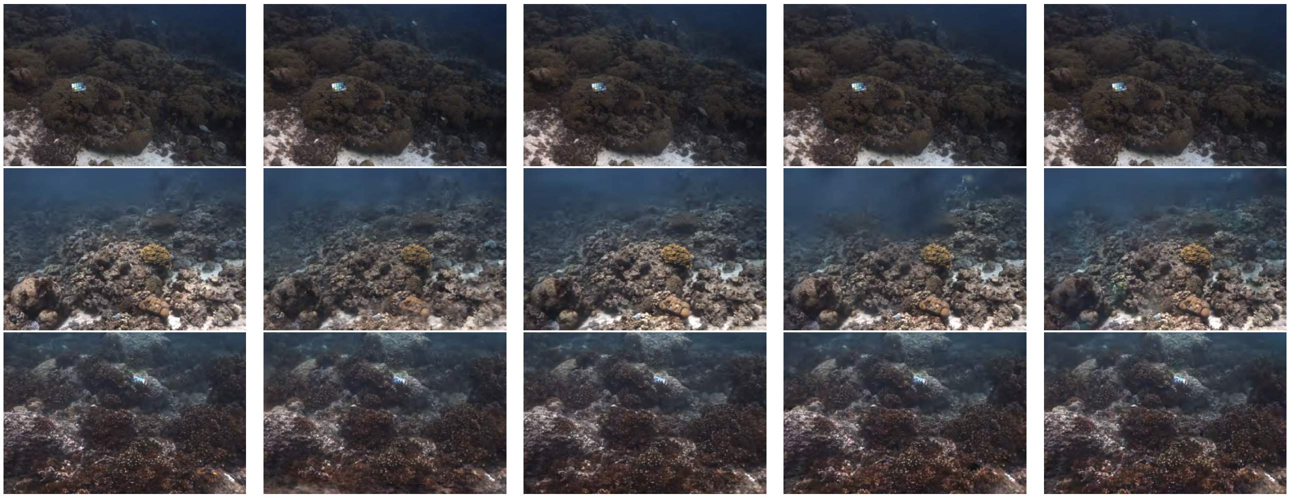
4 APPROACH

Our UW-3DGS representation builds upon 3DGS [17], targeting efficient reconstruction of underwater scenes from multi-view images captured in real-world conditions. It aims to disentangle the scattering media and objects and achieve high-quality rendering of novel views with or without water.

4.1 Motivation

3DGS has achieved high-quality 3D scene reconstruction from images in environments without scattering media. However, applying 3DGS directly to underwater images may introduce numerous noisy floating Gaussians, attempting to model the scattering media inherent in such environments. These floating Gaussians obscure the true colors and geometries of underwater objects, including essential details like seabed topography and vegetation crucial for marine engineering.

Consequently, a manual post-processing step becomes necessary to identify and eliminate these media-influenced Gaussians within the current 3DGS framework for marine applications. Yet, this intervention risks compromising the final 3DGS quality, as those Gaussians are integral to both the learning and image rendering processes. In our approach, by incorporating the scattering effects directly into 3DGS's rendering pipeline during training by image formation model and leveraging the Self-Pruning Supervision Loss,



(a) GT (b) TensoRF (c) SeaThru-NeRF (d) 3DGS (e) Ours w/o Self-Pruning

Figure 5: Visualization of rendered underwater images. Images from the top to bottom rows are from scenes: Caribbean Sea, Red Sea, and Panama.

UW-3DGS not only circumvents the need for such post-processing but also ensures the production of higher-quality reconstruction and rendering of underwater scenes, irrespective of the presence of water.

4.2 Method Design

As the rendering pipeline shown in Figure 4, the colors $\hat{C}$ rendered by 3DGS are utilized as the unattenuated signals J_c, resulting in the formation of a No Water Image (NWI) $\hat{I}_{\text{NW}}$. We then apply the image formation model outlined in Equation (6) to simulate the absorbing and scattering effects present in underwater environments. More precisely, we reformulate Equation (6) as follows:

$$I_c = \hat{I}_{\text{NW}} \cdot (e^{-\beta^D(\mathbf{v}_D)\cdot z}) + B_c^{\infty} \cdot (1 - e^{-\beta^B(\mathbf{v}_B)\cdot z}). \tag{7}$$

This design strategy is intended to compel the 3DGS to focus solely on learning the underwater objects while effectively ignoring the scattering media. Unlike the image rendering model utilized in SeaThru-NeRF [20], which incorporates scattering parameters directly into the volume rendering, our methodology, leveraging Equation (7), does not impede the efficiency of 3DGS's splatting-style rasterization. Consequently, our approach offers significantly enhanced efficiency compared to previous MLP-based approaches.

However, six unknown parameters exist in Equation (7): the attenuation coefficients β^B and β^D, the dependencies of these attenuation coefficients $\mathbf{v}_D$ and $\mathbf{v}_B$, the depth z, and the veiling light B_c^{∞}. The veiling light B_c^{∞}, representing backscatter colors at infinity, can be identified by selecting colors from regions that extend to infinity. Identifying these regions, however, necessitates additional steps to accurately locate areas that represent infinity. Therefore, we assume B_c^{∞} is constant and set it as a learnable parameter $\hat{B_c^{\infty}} \in \mathbb{R}^3$. $\mathbf{v}_D$ and $\mathbf{v}_B$ represent the dependencies of attenuation coefficients β^B and β^D. To simplify the problem, we directly treat β^B and β^D as learnable parameters, $\hat{\beta^B} \in \mathbb{R}^3$ and $\hat{\beta^D} \in \mathbb{R}^3$, for each color channel. Consequently, $\mathbf{v}_D$ and $\mathbf{v}_B$ can be omitted. The depth z can be inferred from 3DGS by substituting the colors of each Gaussian with depth values for rasterization through alpha blending.

$$\hat{z} = \sum_{i \in N} z_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \tag{8}$$

where z_i is the distance to the mean of each Gaussian i along the camera ray.

Finally, after simplifying the complex modeling of underwater light transmission, there are a total of additional three learnable parameters: $\hat{\beta^B}$, $\hat{\beta^D}$, $\hat{B_c^{\infty}}$. The final rendering equation is composed as follows:

$$I_c = \hat{I}_{\text{NW}} \cdot (e^{-\hat{\beta^D}\cdot \hat{z}}) + \hat{B_c^{\infty}} \cdot (1 - e^{-\hat{\beta^B}\cdot \hat{z}}). \tag{9}$$

Though simplifications have been made to the underwater image formation model, our experiments demonstrate the model's effectiveness and efficiency when integrated with the 3DGS rendering pipeline.

4.3 Loss Function

4.3.1 Rendering Loss. We utilize the original rendering loss from 3DGS [17], which includes an L1 loss $\mathcal{L}_1$ and a D-SSIM term $\mathcal{L}_{\text{D-SSIM}}$ between the ground truth (GT) image and the predicted training image I_c:

$$\mathcal{L}_{\mathbf{IMG}} = (1 - \lambda)\mathcal{L}_1 + \lambda \mathcal{L}_{\text{D-SSIM}} \tag{10}$$

4.3.2 Self-Pruning Supervision Loss. While the rendering Equation (9) compels the color prediction $\hat{C}$ directly from 3DGS as unattenuated signals, there are still floating Gaussians present that simulate underwater effects caused by water media. This occurrence is likely due to the extreme simplification of our image formation model, which may not accurately reflect the true non-uniformity of underwater scenes. To enhance the clarity of the learned 3D Gaussians,

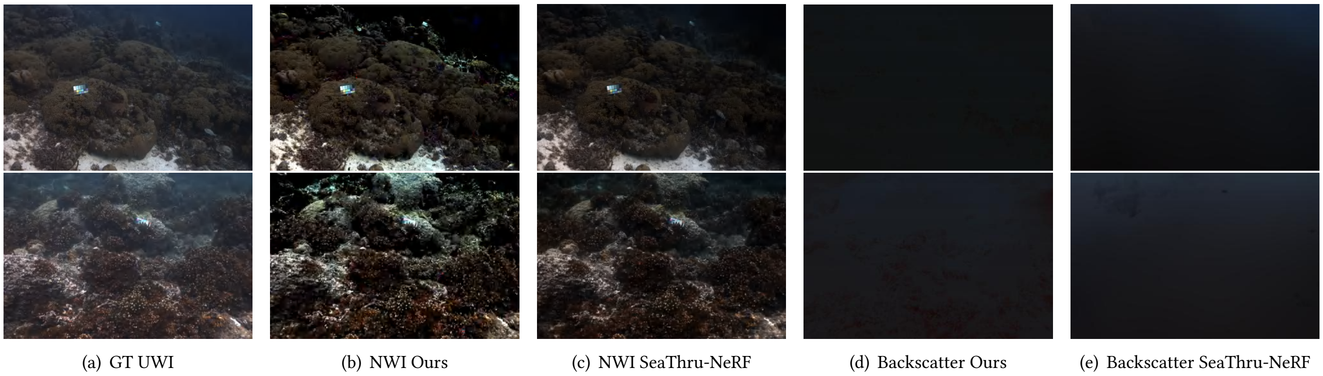


Figure 6: Visual comparisons of NWI on the SeaThru-NeRF dataset. Ours denote "Ours w/o Self-Pruning".

we introduce a Self-Pruning Supervision Loss designed to eliminate these floating Gaussians.

Specifically, we prune the original Gaussians whose scale S exceeds a predefined threshold S_τ. This pruning strategy is motivated by our observation that Gaussians representing water media typically exhibit an excessively large scale. The pruned, or cleaned, Gaussians $\mathcal{G}_{\text{Clean}}$ are then rasterized into a new No Water Image (NWI) $\hat{I}^{\text{Clean}}_{\text{NW}}$. Typically, $\hat{I}^{\text{Clean}}_{\text{NW}}$ appears much clearer than $\hat{I}_{\text{NW}}$ in terms of mitigating scattering effects attributable to water media. Hence, we propose a Self-Pruning Supervision Loss $\mathcal{L}_{\text{SPSL}}$ aimed at reducing the presence of floating Gaussians simulating water media within the learned 3DGS. This is achieved by enforcing consistency between $\hat{I}^{\text{Clean}}_{\text{NW}}$ and $\hat{I}_{\text{NW}}$:

$$\mathcal{L}_{\text{SPSL}} = ||\hat{I}_{\text{NW}} - \hat{I}^{\text{Clean}}_{\text{NW}}||_1. \tag{11}$$

The $\mathcal{L}_{\text{SPSL}}$ can be applied in different training stages. Introduction at a later training stage can facilitate the learning of more detailed content in the scene.

4.3.3 Final Loss. The final loss function is $\mathcal{L} = \mathcal{L}_{\text{SPSL}} + \mathcal{L}_{\textbf{IMG}}$.

5 IMPLEMENTATION DETAILS

The implementation of 3DGS adheres to the default configurations detailed in [17], utilizing PyTorch and CUDA for development. We specify the number of training iterations as 40,000. The pruning scale threshold S_τ is determined to be five times the median of the Gaussians' scales and the order of spherical harmonics is set to three. Default learning rates for opacity, scaling, and rotation parameters are 0.05, 0.005, and 0.001, respectively. The 3D Gaussians undergo densification starting from the 500th iteration and continue until the conclusion of training. Opacity is reset every 3,000 iterations, with a default rate of 0.01 for forced densification. Learning rates for $\hat{\beta^B}$, $\hat{\beta^D}$, and $\hat{B^\infty_c}$ are all set to 0.001. The optimizer utilized is Adam [19]. The Self-Pruning Supervision Loss is integrated into the training process flexibly—either from the beginning, never, or during the last 10,000 iterations, depending on the desired outcome. All models are trained using a single NVIDIA Tesla V100 (32 GB) GPU.

6 EXPERIMENT SETTING

6.1 Dataset

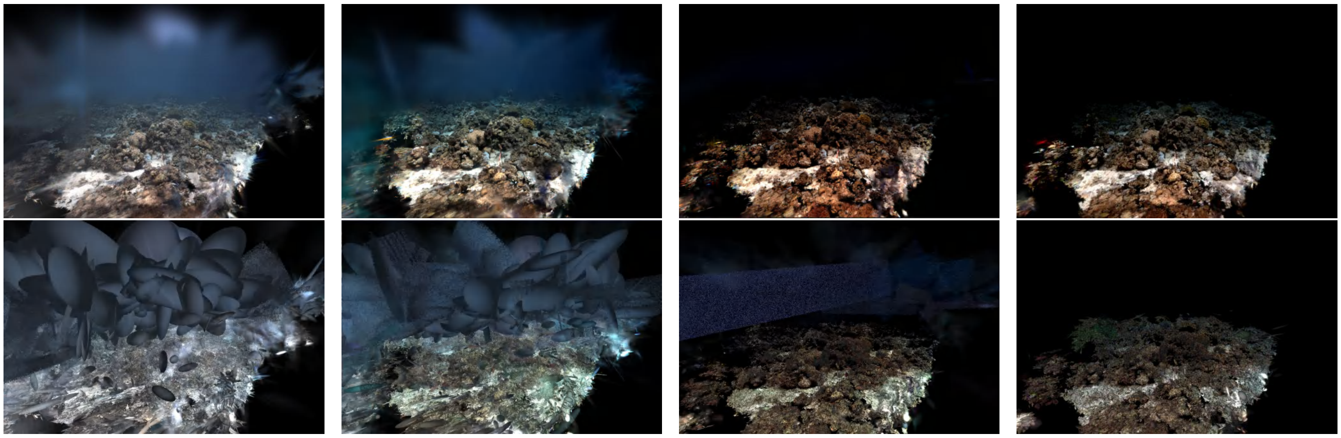
We train and evaluate our model on the SeaThru-NeRF dataset [20] and the UWBundle dataset [39]. The SeaThru-NeRF dataset comprises images from three real-world underwater scenes, captured facing forward in different locations: the Pacific Sea (in Panama with 20 images), the Red Sea (in Israel with 20 images), and the Caribbean Sea (in Curacao with 18 images). We employ white-balanced images as the ground truth for training and evaluation, following the same procedure as [20]. Camera parameters for the dataset images are estimated using the Structure from Motion (SfM) software, COLMAP [36]. To ensure a fair comparison, we adopt the experimental setup of SeaThru-NeRF [20], which uses the standard training and testing split method from LLFF [25]: test images are selected every 8th image, with the remaining used as training images. The UWBundle dataset consists of 36 images depicting a synthetic rock platform submerged in a water tank under laboratory conditions. These images were taken from various angles by an underwater camera following a lawnmower pattern, a common approach in underwater surveys.

6.2 Baseline Method

Our method is evaluated against various NeRF-based approaches with code publicly available, categorizing them based on their consideration of media effects during rendering. For methods that assume clear media, we draw comparisons with 3DGS [17] and TensoRF [7]. In contrast, for methods that account for media effects, we compare our approach with SeaThru-NeRF [20].

6.3 Evaluation Rubics

The quality of image rendering is assessed using three metrics: PSNR, SSIM, and LPIPS [50]. Evaluations are conducted from the following perspectives: (1) Quality of Novel Underwater Image (UWI) Rendering: This evaluation compares the rendered UWI quality against the ground truth (GT) UWI. To achieve this, we designate our approach as "Ours w/o Self-Pruning," which excludes the $\mathcal{L}_{\text{SPSL}}$, allowing the model to concentrate solely on UWI rendering without addressing the removal of floating Gaussians that



(a) 3DGS (b) Ours w/o Self-Pruning (c) Ours w/ delayed Self-Pruning (d) Ours w/ default Self-Pruning

Figure 7: Visualization of underwater scene 3D reconstruction: The top row displays splatting results, while the second row showcases the ellipsoids.

simulate water media. (2) NWI Rendering Quality: This assessment focuses on the color restoration performance. We introduce two variations of our method: "Ours w/ default Self-Pruning" and "Ours w/ delayed Self-Pruning." The former applies $\mathcal{L}_{\text{SPSL}}$ from the beginning to effectively eliminate floating Gaussians, while the latter introduces $\mathcal{L}_{\text{SPSL}}$ in the final training phase, striking a balance between UWI and NWI rendering quality. (3) Training and Rendering Efficiency Comparison: High-efficiency training is essential for facilitating real-world applications. This aspect evaluates how our method and others fare in terms of speed. (4) Density of Noisy Floating Gaussians: This comparison assesses the level of disentanglement between water media and the objects emitting radiance within the scene, indicated by the density of noisy floating Gaussians simulating scattering effects.

7 EXPERIMENT RESULTS

All experimental results are obtained through our rerunning.

7.1 Quantitative Comparison

7.1.1 Rendering Quality Comparison: Table 1 presents the quantitative results of the novel view rendering quality of UWI on the SeaThru-NeRF dataset. "Ours w/o Self-Pruning" achieves the highest rendering quality and the lowest training time. This proves that integrating the image formation model (Equation (9)) into the 3DGS's rendering pipeline enhances the rendering quality of underwater scenes and reconstruction efficiency, thereby demonstrating the efficacy of our approach. The results of "Ours w/ default or delayed Self-Pruning" will be discussed in the section of the ablation study and limitations.

7.1.2 Training Time Comparison: It can be observed that the training time for SeaThru-NeRF is substantial due to its network backbone being purely based on MLPs. Our method is more than *100 times* more efficient than SeaThru-NeRF when training on a single NVIDIA Tesla V100 GPU. "Ours w/o Self-Pruning" and "Ours w/ delayed Self-Pruning" are also more efficient than 3DGS on training.

Table 1: Quantitative comparisons of UWI rendering quality and training time on SeaThru-NeRF dataset [20].

Method	Scene Specific Training Time (Minutes) ↓	PSNR↑	SSIM↑	LPIPS↓
3DGS [17]	32	26.113	0.86	**0.216**
TensoRF [7]	61	24.307	0.787	0.285
SeaThru-NeRF [20]	3272	25.768	0.806	-
Ours w/o Self-Pruning	**28**	**26.604**	**0.868**	0.221
Ours w/ default Self-Pruning	45	17.488	0.559	0.375
Ours w/ delayed Self-Pruning	31	21.520	0.730	0.346

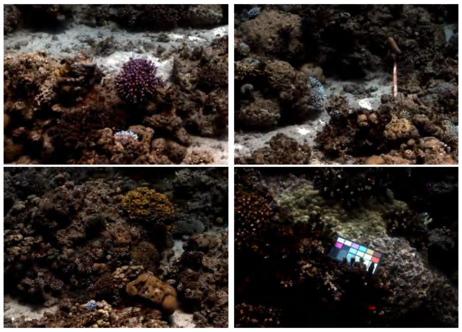


Figure 8: NWI of seabed. Direct rendering of 3D Gaussians from "Ours w/ default Self-Pruning".

7.2 Visual Comparisons

7.2.1 UWI Visual Comparisons. Figure 5 showcases the underwater image (UWI) rendering results of 3DGS, TensoRF, SeaThru-NeRF, and "Ours w/o Self-Pruning". Although their quantitative performance is closely matched, discernible differences in rendering characteristics can still be observed through visual comparison. Both TensoRF and SeaThru-NeRF demonstrate superior performance in capturing distant content. However, TensoRF exhibits diminished

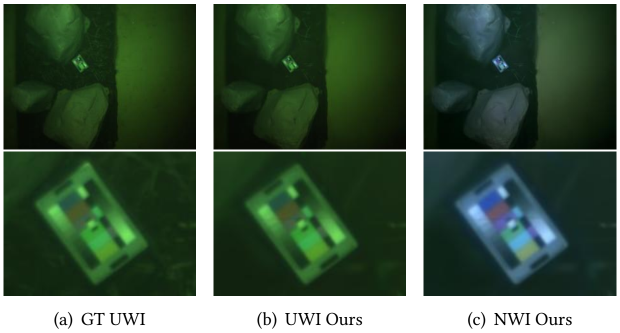


Figure 9: Visualization of rendered UWI and NWI in the UWBundle Dataset. Ours denotes "Ours w/o Self-Pruning".

quality for content closer to the near plane of the scene, likely due to the configuration of its bounding box, which plays a crucial role in voxel-grid-based scene representations. SeaThru-NeRF exhibits inferior details in scene content. Floating Gaussians obscure the rendering of distant content in 3DGS. Our method demonstrates the best overall performance. Figure 9 (b) presents the rendering results of our method in the UWBundle-RR scene from the UWBundle dataset [39]. SeaThru-NeRF cannot converge in this dataset due to the challenging image distributions in lawnmower patterns, while Our method can render high-quality novel UWI.

7.2.2 NWI Visual Comparisons. The lack of GT NWI (No-Water Images) complicates the evaluation of the no-water rendering outcomes. Figure 6 illustrates NWI visual comparisons between "Ours w/o Self-Pruning" and SeaThru-NeRF. Our NWIs in Figure 6 (b) are derived from rasterizing pruned 3D Gaussians, which requires a post-pruning process. Both methods effectively separate seabed objects and backscattering colors, demonstrating comparable performance in NWI rendering. More results of our NWI renderings are introduced in Section 7.2.3.

Figure 9 (c) also displays NWI rendering outcomes on the UWBundle dataset [39], demonstrating that our method effectively restores the unattenuated colors. SeaThru-NeRF failed to converge in this dataset.

7.2.3 Seabed Reconstruction. We compare the seabed 3D reconstruction with 3DGS [17], results are shown in Figure 7 where the top row displays the splatting results, and the second row presents the raw 3D Gaussians/ellipsoids. While raw 3D Gaussians/ellipsoids may not directly reflect the quality of 3D reconstruction, they can serve as indicators of the cleanliness and overall quality of the reconstruction. Unlike the messy results from 3DGS [17] in Figure 7 (a), results from "Ours w/ delayed or default Self-Pruning" (Figure 7 (c) and (d)) are capable of reconstructing the seabed with fine 3D Gaussians that capture intricate details and exhibit minimal noisy floating 3D Gaussians. As a result, our method proves to be more effective than 3DGS [17] in the 3D reconstruction of seabeds in real-world underwater scenes. Close-up views of the reconstructed seabed from "Ours w/ default Self-Pruning" are shown in Figure 8, which further shows our high-quality reconstruction.

7.3 Ablation Study

7.3.1 Image Formation Model. The image formation model is designed to replicate the absorbing and scattering effects characteristic of underwater environments. Figure 4 demonstrates the model's mechanism to incorporate underwater effects into the rendering results of raw 3D Gaussians. Figures 6, 7, 8, and 9 validate the model's ability to restore colors. Figure 6 (d) and (e) exemplify the precise rendering of backscatter colors. Collectively, these results underscore the effectiveness of the image formation model in simulating realistic underwater imaging conditions.

7.3.2 Self-Pruning Supervision Loss. The loss function is designed to eliminate floating noisy Gaussians. As illustrated in Figure 7, without the loss ("Ours w/o Self-Pruning") or introducing this loss later ("Ours w/ delayed Self-Pruning") in the training process results in a higher presence of noisy Gaussians. "Ours w/ default Self-Pruning" achieves the best/cleanest NWI rendering results compared to the other variants. These observations substantiate the effectiveness of the loss function in enhancing the clarity and quality of the seabed 3D reconstruction.

7.4 Limitations and Future Research

(1) Our method relies on white-balanced images provided by the SeaThru-NeRF dataset [20]. Its performance may degrade when applied directly to raw underwater images under low-light conditions. (2) The current approach necessitates camera poses extracted by SfM software, suggesting that future research on SLAM holds significant potential. (3) The determination of the pruning factor is manually selected, indicating a need for a more automated or adaptive approach. (4) The UWI rendering quality of "Ours w/ delayed or default Self-Pruning" in Table 1 is affected due to the incomplete depth maps caused by no floating 3D Gaussians in the space. Completing the depth maps may be a future improvement. (5) Applying the Self-Pruning Supervision Loss will increase the training time due to the additional rendering branch.

8 CONCLUSIONS

In this work, we enhance the 3DGS framework with an image formation model to facilitate high-quality rendering, color restoration, and reconstruction of underwater scenes. Furthermore, we introduce the Self-Pruning Supervision Loss to refine the 3D Gaussians, ensuring they are free from floating noisy 3D Gaussians, thus enabling clean 3D reconstruction of seabed landscapes. Our experiments convincingly demonstrate the efficacy of both the integration of the image formation model with 3DGS and the Self-Pruning Supervision Loss to remove noisy floating 3D Gaussians. Our approach sets a new benchmark in performance over existing methodologies for underwater novel view rendering and achieves remarkable clarity in 3D seabed reconstruction from real-world underwater images in the SeaThru-NeRF dataset.

ACKNOWLEDGMENTS

The research was supported by the XXX anonymous Grant.

REFERENCES

[1] Derya Akkaynak and Tali Treibitz. 2018. A revised underwater image formation model. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6723–6732.

[2] Derya Akkaynak and Tali Treibitz. 2019. Sea-thru: A method for removing water from underwater images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1682–1691.

[3] Derya Akkaynak, Tali Treibitz, Tom Shlesinger, Yossi Loya, Raz Tamir, and David Iluz. 2017. What is the space of attenuation coefficients in underwater computer vision?. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4931–4940.

[4] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*. 5470–5479.

[5] Yael Bekerman, Shai Avidan, and Tali Treibitz. 2020. Unveiling Optical Properties in Underwater Images. In *2020 IEEE International Conference on Computational Photography (ICCP)*. 1–12. https://doi.org/10.1109/ICCP48838.2020.9105267

[6] James F Blinn. 1982. Light reflection functions for simulation of clouds and dusty surfaces. *Acm Siggraph Computer Graphics* 16, 3 (1982), 21–29.

[7] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. 2022. TensoRF: Tensorial Radiance Fields. In *ECCV*.

[8] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. 2021. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *ICCV*. 14124–14133.

[9] Wei-Ting Chen, Wang Yifan, Sy-Yen Kuo, and Gordon Wetzstein. 2023. Dehazenerf: Multiple image haze removal and 3d shape reconstruction using neural radiance fields. *arXiv preprint arXiv:2303.11364* (2023).

[10] Paul Drews, Erickson Nascimento, Filipe Moraes, Silvia Botelho, and Mario Campos. 2013. Transmission estimation in underwater single images. In *Proceedings of the IEEE international conference on computer vision workshops*. 825–830.

[11] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. 2022. Plenoxels: Radiance Fields Without Neural Networks. In *CVPR*. 5501–5510.

[12] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. 2021. Fastnerf: High-fidelity neural rendering at 200fps. In *CVPR*. 14346–14355.

[13] Tao Hu, Shu Liu, Yilun Chen, Tiancheng Shen, and Jiaya Jia. 2022. Efficientnerf efficient neural radiance fields. In *CVPR*. 12902–12911.

[14] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 2013. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence* 36, 7 (2013), 1325–1339.

[15] Jules S Jaffe. 1990. Computer modeling and the design of optimal underwater imaging systems. *IEEE Journal of Oceanic Engineering* 15, 2 (1990), 101–111.

[16] Henrik Wann Jensen and Per H Christensen. 2023. Efficient simulation of light transport in scenes with participating media using photon maps. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 301–310.

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics* 42, 4 (2023).

[18] Justin Kerr, Letian Fu, Huang Huang, Yahav Avigal, Matthew Tancik, Jeffrey Ichnowski, Angjoo Kanazawa, and Ken Goldberg. 2023. Evo-nerf: Evolving nerf for sequential robot grasping of transparent objects. In *Conference on Robot Learning*. PMLR, 353–367.

[19] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint:1412.6980* (2014).

[20] Deborah Levy, Amit Peleg, Naama Pearl, Dan Rosenbaum, Derya Akkaynak, Simon Korman, and Tali Treibitz. 2023. SeaThru-NeRF: Neural Radiance Fields in Scattering Media. In *CVPR*. 56–65.

[21] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. 2020. Neural sparse voxel fields. *Advances in Neural Information Processing Systems* 33 (2020), 15651–15663.

[22] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. 2019. Neural Volumes: Learning Dynamic Renderable Volumes from Images. *ACM Transactions on Graphics.* 38, 4 (2019).

[23] Li Ma, Xiaoyu Li, Jing Liao, Qi Zhang, Xuan Wang, Jue Wang, and Pedro V Sander. 2022. Deblur-nerf: Neural radiance fields from blurry images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12861–12870.

[24] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. 2022. Nerf in the dark: High dynamic range view synthesis from noisy raw images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16190–16199.

[25] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. 2019. Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines. *ACM Transactions on Graphics* 38, 4, Article 29 (2019), 14 pages. https://doi.org/10.1145/3306346.3322980

[26] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*. 405–421.

[27] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics* 41, 4 (2022), 102:1–102:15.

[28] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H. Mueller, Chakravarty R. Alla Chaitanya, Anton S. Kaplanyan, and Markus Steinberger. 2021. DONeRF: Towards Real-Time Rendering of Compact Neural Radiance Fields using Depth Oracle Networks. *Computer Graphics Forum* 40, 4 (2021). https://doi.org/10.1111/cgf.14340

[29] Jan Novák, Iliyan Georgiev, Johannes Hanika, and Wojciech Jarosz. 2018. Monte Carlo methods for volumetric light transport simulation. In *Computer graphics forum*, Vol. 37. Wiley Online Library, 551–576.

[30] Naama Pearl, Tali Treibitz, and Simon Korman. 2022. Nan: Noise-aware nerfs for burst-denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12672–12681.

[31] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. 2021. KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs. *arXiv preprint:2103.13744* (2021).

[32] Antoni Rosinol, John J Leonard, and Luca Carlone. 2023. Nerf-slam: Real-time dense monocular slam with neural radiance fields. In *IROS*. 3437–3444.

[33] Yoav Y Schechner and Nir Karpel. 2004. Clear underwater vision. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, Vol. 1. IEEE, I–I.

[34] Yoav Y Schechner and Nir Karpel. 2005. Recovery of underwater visibility and structure by polarization analysis. *IEEE Journal of oceanic engineering* 30, 3 (2005), 570–587.

[35] Yoav Y Schechner, Srinivasa G Narasimhan, and Shree K Nayar. 2001. Instant dehazing of images using polarization. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Vol. 1. IEEE, I–I.

[36] Johannes Lutz Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *CVPR*. 4104–4113.

[37] Advaith Venkatramanan Sethuraman, Manikandasriram Srinivasan Ramanagopal, and Katherine A Skinner. 2023. Waternerf: Neural radiance fields for underwater scenes. In *OCEANS 2023-MTS/IEEE US Gulf Coast*. IEEE, 1–7.

[38] Neeraj Sharma, Vijay Kumar, and Sunil Kumar Singla. 2021. Single image defogging using deep learning techniques: past, present and future. *Archives of Computational Methods in Engineering* 28 (2021), 4449–4469.

[39] Katherine A. Skinner, Eduardo Iscar Ruland, and M. Johnson-Roberson. 2017. Automatic Color Correction for 3D Reconstruction of Underwater Scenes. In *IEEE International Conference on Robotics and Automation.*

[40] Cheng Sun, Min Sun, and Hwann-Tzong Chen. 2022. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*. 5459–5469.

[41] Tali Treibitz and Yoav Y Schechner. 2008. Active polarization descattering. *IEEE transactions on pattern analysis and machine intelligence* 31, 3 (2008), 385–399.

[42] Chen Wang, Xian Wu, Yuan-Chen Guo, Song-Hai Zhang, Yu-Wing Tai, and Shi-Min Hu. 2022. Nerf-sr: High quality neural radiance fields using supersampling. In *Proceedings of the 30th ACM International Conference on Multimedia*. 6445–6454.

[43] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. 2021. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. *Advances in Neural Information Processing Systems* 34 (2021), 27171–27183.

[44] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. 2021. Ibrnet: Learning multi-view image-based rendering. In *CVPR*. 4690–4699.

[45] Chi Yan, Delin Qu, Dong Wang, Dan Xu, Zhigang Wang, Bin Zhao, and Xuelong Li. 2023. Gs-slam: Dense visual slam with 3d gaussian splatting. *arXiv preprint arXiv:2311.11700* (2023).

[46] Miao Yang, Jintong Hu, Chongyi Li, Gustavo Rohde, Yixiang Du, and Ke Hu. 2019. An in-depth survey of underwater image enhancement and restoration. *IEEE Access* 7 (2019), 123638–123657.

[47] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. 2021. Plenoctrees for real-time rendering of neural radiance fields. In *ICCV*. 5752–5761.

[48] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. 2021. pixelnerf: Neural radiance fields from one or few images. In *CVPR*. 4578–4587.

[49] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. 2020. NeRF++: Analyzing and Improving Neural Radiance Fields. *arXiv preprint:2010.07492* (2020).

[50] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*. 586–595.

[51] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. 2021. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Transactions on Graphics* 40, 6 (2021), 1–18.

[52] Jingchun Zhou, Tianyu Liang, Zongxin He, Dehuan Zhang, Weishi Zhang, Xianping Fu, and Chongyi Li. 2023. WaterHE-NeRF: Water-ray Tracing Neural Radiance Fields for Underwater Scene Reconstruction. *arXiv preprint arXiv:2312.06946* (2023).