

Beyond NeRF Underwater: Learning Neural Reflectance Fields for True Color Correction of Marine Imagery

Tianyi Zhang¹, Matthew Johnson-Roberson¹

Abstract— Underwater imagery often exhibits distorted coloration as a result of light-water interactions, which complicates the study of benthic environments in marine biology and geography. In this research, we propose an algorithm to restore the true color (albedo) in underwater imagery by jointly learning the effects of the medium and neural scene representations. Our approach models water effects as a combination of light attenuation with distance and backscattered light. The proposed neural scene representation is based on a neural reflectance field model, which learns albedos, normals, and volume densities of the underwater environment. We introduce a logistic regression model to separate water from the scene and apply distinct light physics during training. Our method avoids the need to estimate complex backscatter effects in water by employing several approximations, enhancing sampling efficiency and numerical stability during training. The proposed technique integrates underwater light effects into a volume rendering framework with end-to-end differentiability. Experimental results on both synthetic and real-world data demonstrate that our method effectively restores true color from underwater imagery, outperforming existing approaches in terms of color consistency. Our code and data are released at <https://github.com/tyz1030/neuralsea.git>

I. INTRODUCTION

Optical imaging is being widely used in exploring the benthic world together with modern underwater robotic systems. The visual information presented in RGB format reveals rich details about underwater ecosystems and artifacts. For example, images collected by an underwater robot can be used to assess the health of coral reefs and segment live corals from dead samples [1]. However, the colors displayed in underwater images are consistently distorted due to wavelength-dependent attenuation and veiling effects resulting from light-water interactions. Such effects alter the visual appearance of images, as well as the performance of downstream tasks such as detection, classification, or segmentation [2]. Restoring the color in underwater imagery is of great interest to communities working on marine ecology, biology, and geography, etc.

The formation of underwater color distortion has seen significant work, in which two kinds of light-water interaction are commonly studied: attenuation and scattering [3], [4]. Attenuation describes the process whereby water absorbs light at varying rates depending on the wavelength. Red light is absorbed most quickly leading to a loss of the red part of the visual spectrum in typical underwater images [5]. Underwater light scattering refers to the process by which

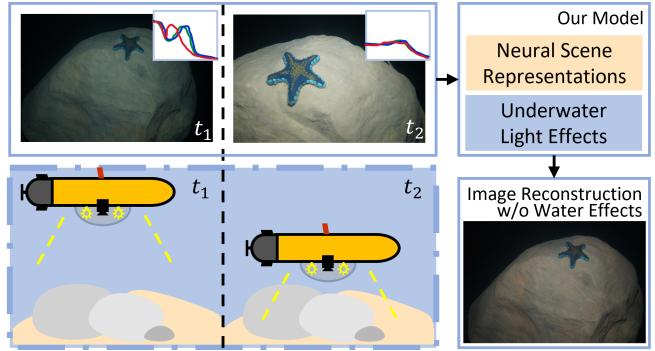


Fig. 1: Observing an underwater scene from different altitudes results in variating color distribution over the RGB channels. Such observations encode the physics of light-water interactions. Our proposed model leverages this cue to restore the true color of underwater scenes by learning water effects together with neural scene representations.

light is dispersed in various directions as it interacts with water molecules, suspended particles, and other microscopic elements within the underwater environment [3]. While in graphics multiple-scattering are typically modeled, in water photons reflected to the camera without striking the scene, i.e. backscatter, have a major impact on image formation by creating a veiling effect. Although our understanding of water optics has advanced, restoring color in underwater images is still challenging. While these effects are well-modeled, accurately estimating them from real data in uncontrolled environments remains an open problem.

Early studies on marine optics developed underwater image formation models [3], [6] and measured absorption and scattering functions from different types of water samples [5], [7]. With the above work, images can be synthesized with underwater effects [8]. However, this approach is insufficient for accurately correcting the color of real-world underwater images, as the measurements of a finite number of water optic properties cannot be reliably applied to novel field data. Recent progresses on structure-from-motion (SfM) and deep learning have inspired the development of data-driven algorithms for underwater color correction. SfM-based method [9] estimates the true color (albedo) with multiple-view geometry constraints, but is only able to generate sparse results on feature points. Deep-learning-based methods [2], [10], [11] are able to correct the color with physical information, but the result depends on prior color distributions or pre-training.

¹T. Zhang and M. Johnson-Roberson are with the Robotics Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA {tianyiz4, mjk}@andrew.cmu.edu

Combining insights from both types of methods, we developed a unified model that effectively restores the true color in underwater imagery (Fig. 1). Our proposed model optimizes the attenuation and backscatter coefficients together with a neural reflectance field [12] from a sequence of observations without any assumptions on prior color distributions. Based on the observation that water and scene are separable given volume density, we embed a logistic regression function in our neural scene representation which allows us to apply different light-transmitting physics to water and the scene, while maintaining end-to-end differentiability of our model. Our experiments demonstrate that our method is able to generate photo-realistic results with restored true color in a dense format, outperforming previous studies, particularly when the underlying albedo of the scene has a biased color distribution in the RGB space.

II. RELATED WORK

A. Underwater Image Formation Model

According to Jaffe-McGlamery model [3], [6], the formation of underwater images can be decomposed into direct signals, forward-scattering, and backscatter. Direct signals refer to the light that is reflected from the underwater scene. Backscatter refers to the phenomenon in which light enters a camera without being reflected directly from the scene. The trajectory of a photon after interacting with a particle in water is characterized by volume scattering functions (VSFs) [7]. These empirical functions are dependent on both viewing and lighting directions. Forward-scattering occurs when a photon deviates from its direct path before reaching the sensor, resulting in a blurred image. This effect can be modeled by convolution operations [3] or Gaussian blurring [8].

In this work, we face challenges of modeling VSF for robots with different camera-light configurations. To overcome this challenge, we propose several approximations for backscatter that are applicable to the cases where the camera and light source move as a rigid body. Our scene representations do not model forward scattering, as the error introduced by forward scattering is zero-mean and negligible [13].

B. Neural Implicit Representations

Neural implicit representations, which encode signals as continuous functions instead of discrete samples, have been widely used in learning visual appearances and structures. NeRF [14] is a kind of neural implicit representation that learns a 3D scene in the form of a neural field of volume density and radiance. The volume rendering equations in NeRF, which is based on radiative transfer equation (RTE), are not only good for inferring the 3D geometry of objects but also have the power to model the water effects such as absorption and scattering. Based on NeRF’s framework, neural reflectance field [12] and its variants [15] model the reflectance of the scene which enables the high-quality rendering under novel lighting conditions.

For underwater scenes illuminated by light sources attached to the robot, the appearance of the scene changes due to the robot’s movement. To accommodate for these

appearance changes resulting from varying illumination conditions, it is necessary to model reflectance properties of the scene instead of radiance. Therefore, we opt to use a neural reflectance field [12] as the foundational model for 3D underwater scene representations.

C. Underwater Color Correction

Early studies on underwater color correction make assumptions on underlying color distributions, e.g. histogram equalization [16], grayworld [17], or dark-channel prior [18]. However, color balanced from the above assumptions lacks consistency when the same scene is observed from multiple views due to range-dependent water effects.

Bryson et al. [9] leverages the physical constraints from multiple-view geometry to estimate the true color of the scene. However, this method only estimates the true color of feature points and is unable to directly generate color-corrected images in a dense format.

Further progress in this field is made with deep learning approaches. WaterGAN [10] proposes to generate a synthetic dataset with ground truth depth and colors by training a GAN, then train a color correction network to restore the color together with depth estimations. FUnIEGAN [2] employs a GAN, emphasizing image quality for downstream tasks rather than adhering to physical constraints and as such is able to achieve real-time performance. GAN-based methods, such as those mentioned above, require pre-training on a dataset. These methods can exhibit biases if the underlying color distribution differs from that of the training set. In contrast, our approach does not require any pre-training on pre-collected datasets. Rather, it restores color by creating neural scene representations using a series of observations from multiple perspectives.

WaterNeRF [11] utilizes mip-NeRF [19] to model the underwater scene. Based on depth estimation from mip-NeRF, WaterNeRF learns the absorption and backscatter coefficients by optimizing the Sinkhorn loss between rendered image and histogram equalized image. Our approach diverges from WaterNeRF in that we model the scene as a reflectance field, which accounts for changes in illuminance, as opposed to a radiance field. Furthermore, we do not make any assumptions regarding the underlying color distributions.

Lastly, all the approaches mentioned above [9]–[11] use the model proposed in [20] to account for backscatter, which assumes natural and ambient light to be the major illumination source of scattering. In other words, their formulations are based on the assumption that the intensity of scattering is spatially constant, which does not hold for underwater robots equipped with light sources, taking light fall-off into consideration. In our work, we depart from [20]’s model and propose several approximations on backscatter for underwater robots.

III. METHODOLOGY

A. Neural Scene Representation

We employ neural reflectance field [12] to model the underwater scene observed by an underwater robot with

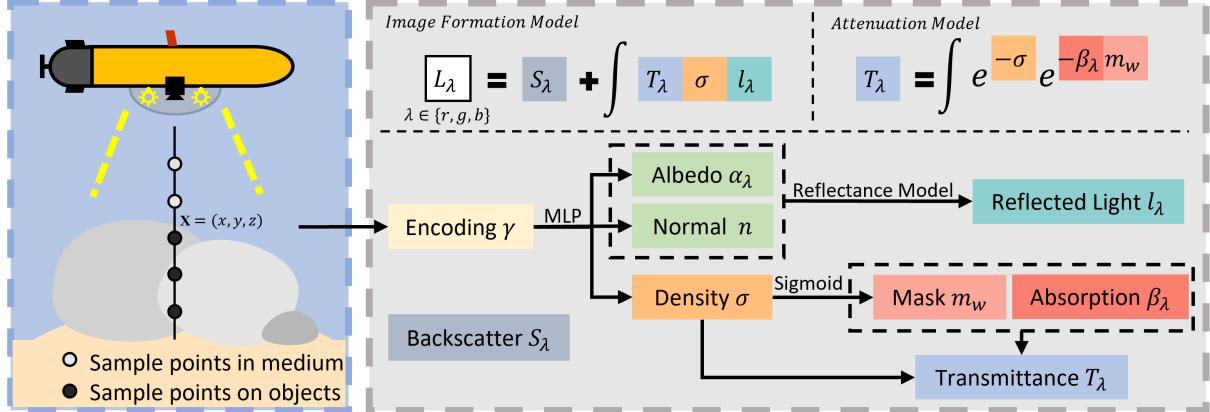


Fig. 2: Our proposed model: Sample points \mathbf{x} are first mapped into positional encoding $\gamma(\mathbf{x})$, as the input of an MLP. The output of the MLP consists of albedo α , surface normal \mathbf{n} , and volume density σ . Backscatter S_λ and attenuation coefficient β_λ are global parameters optimized along with the MLP. With α and \mathbf{n} we can calculate the reflected radiance l_λ from the scene. We apply a sigmoid function on σ to separate water from scene and calculate transmittance T_λ through the scene and water using different coefficients. With S_λ , T_λ , σ and l_λ , our rendering model predicts the pixel values in the image.

onboard lights. The continuous scene is represented as a function of 3D location $\mathbf{x} = (x, y, z)$ in the global coordinate frame. The outputs of the function are the rendering properties $(\sigma, \alpha, \mathbf{n})$, where σ is the volume density, $\alpha = (\alpha_r, \alpha_g, \alpha_b)$ is the albedo and $\mathbf{n} = (n_x, n_y, n_z)$ is the surface normal (see Fig. 2).

In practice, we first sample 3D points \mathbf{x} on camera rays in the global coordinate frame. We then use hash encoding γ to map the input \mathbf{x} into a higher-dimensional space [21] before feeding it into a nested multilayer perceptron (MLP):

$$(\sigma, \alpha, \mathbf{n}) = \text{MLP}(\gamma(\mathbf{x})) \quad (1)$$

B. Rendering Equations

The volume rendering equation [22], [23] maps a camera ray $\mathbf{x} = \mathbf{o} - t\omega$ into the radiance L_λ captured at location \mathbf{o} in direction ω :

$$L_\lambda(\mathbf{o}, \omega) = \int_{t=0}^d T_\lambda(\mathbf{x}) \sigma(\mathbf{x}) l_\lambda(\mathbf{x}) dt \quad (2)$$

Here T_λ is the transmittance from \mathbf{x} to \mathbf{o} , σ is the volume density, l_λ is the scattered radiance from \mathbf{x} to \mathbf{o} along the ray, and λ indicates the wavelength. In this study, the wavelength is discretized into RGB space that $\lambda \in \{r, g, b\}$ [24].

For a light beam emitted from \mathbf{x} to \mathbf{o} , the fraction of light that reaches the camera is described by the transmittance T_λ :

$$T_\lambda(\mathbf{x}) = \exp\left(-\int_{s=0}^t \sigma_\lambda(\mathbf{o} - s\omega) ds\right) \quad (3)$$

Here, σ_λ denotes the attenuation coefficient as a function of the 3D location $\mathbf{o} - s\omega$, which combines the extinction of light due to both volume-density-dependent out-scattering and wavelength-dependent absorption [3], [23]. The formulation of σ_λ will be further discussed in III-C.

The scattered radiance l_λ from the scene, as a part of the integrand in Eq. 2, is formulated as follows:

$$l_\lambda(\mathbf{x}) = \int_{S^2} f_\lambda(\mathbf{x}, \omega, \omega_i) I_\lambda(\mathbf{x}, \omega_i) d\omega_i \quad (4)$$

where S^2 represents the spherical domain around point \mathbf{x} , f_λ is the phase function that governs the distribution of light scattered at \mathbf{x} , and I_λ is the incident radiance from direction ω_i into \mathbf{x} .

In practice, we follow the assumptions in [9] that object surfaces underwater are Lambertian, which scatters light into all directions equally. Following Lambert's cosine law, the phase function for objects underwater is described as: $f_\lambda(\mathbf{x}, \omega, \omega_i) = \alpha_\lambda(\mathbf{x}) \cos(\mathbf{n}(\mathbf{x}), \omega_i)$. Here $\alpha_\lambda(\mathbf{x})$ and $\mathbf{n}(\mathbf{x})$ are the albedo and normal at \mathbf{x} estimated by the neural network. In other words, we are not modeling any specular reflection which is rare underwater.

Inferring the phase function f_λ of water volumes, i.e. VSF, is challenging and not scalable on real robots due to different light and camera configurations. To address this, we propose approximating backscatter in the image as a constant and moving away from estimating VSF (see III-D), by which the complexity of our approach is significantly reduced while still achieving accurate and realistic rendering results .

Similar to [9], we only consider direct illumination from onboard lights. While natural and ambient light also impacts the lighting in shallow water, they are out of the scope of this work. The direct illumination on point \mathbf{x} from the light source is expressed by:

$$I_\lambda(\mathbf{x}, \omega_i) = T_\lambda^i(\mathbf{x}) E_\lambda^i(\mathbf{x}) \quad (5)$$

Here i indicates the light source from direction ω_i , T_λ^i is the transmittance from the light source to \mathbf{x} (the calculation is similar to Eq. 3), and $E_\lambda^i(\mathbf{x})$ is the intensity of light source i evaluated at \mathbf{x} taking light fall-off with distance into account.

C. Unified Transmittance Model

The attenuation of light in water can be modeled with a transmittance term T_λ given attenuation coefficient σ_λ and distance t :

$$T_\lambda = \exp\left(-\int_{s=0}^t \sigma_\lambda ds\right) = \exp(-\sigma_\lambda t) \quad (6)$$

Given the emitted radiance E , the arrived radiance is $T_\lambda E$. The attenuation coefficient σ_λ for water can be decomposed into the out-scattering coefficient σ and the absorption coefficient β_λ [3]. Notably, the out-scattering coefficient σ is independent of the wavelength of the light [25], and can be represented as the volume density in rendering equations.

In the neural reflectance field, volume density is a function of spatial location \mathbf{x} , so we have:

$$\sigma_\lambda(\mathbf{x}) = \sigma(\mathbf{x}) + \beta_\lambda \quad (7)$$

where $\sigma(\mathbf{x})$ is predicted by the neural implicit functions and β_λ will be optimized as a global parameter that doesn't change with spatial locations.

On a camera ray, points in the water attenuate light through both absorption and out-scattering, as described by Eq. 7. In contrast, points on objects have no wavelength-dependent absorption effects. So for underwater scenes $\sigma_\lambda(\mathbf{x})$ can be formulated as follows:

$$\sigma_\lambda(\mathbf{x}) = \begin{cases} \sigma(\mathbf{x}) + \beta_\lambda, & \text{if } \mathbf{x} \text{ is in water} \\ \sigma(\mathbf{x}), & \text{if } \mathbf{x} \text{ is on objects} \end{cases} \quad (8)$$

When sampling points from non-transparent objects, the volume density $\sigma(\mathbf{x})$ should typically be large enough that regardless of whether \mathbf{x} is in water or on objects, $\sigma(\mathbf{x}) \approx \sigma(\mathbf{x}) + \beta_\lambda$. However, it's still important to maintain the separate attenuation coefficients in Eq. 8 during training until the prediction of $\sigma(\mathbf{x})$ has converged.

To apply Eq. 8, we need to differentiate water from the rest of the scene. We experimentally observe that the value of $\sigma(\mathbf{x})$ for objects is at least 10 times greater than that in clear water. This observation also aligns with the measurements by Jerlov [26]. Assuming that there are no highly transparent objects in the scene other than water, we define the following logistic regression functions using the sigmoid function:

$$\begin{aligned} m_o(\mathbf{x}) &= \text{sigmoid}(a(\sigma(\mathbf{x}) - b)) \\ m_w(\mathbf{x}) &= 1 - m_o(\mathbf{x}) \end{aligned} \quad (9)$$

where m_o and m_w indicate the probabilities of the query point \mathbf{x} being on non-transparent objects and water, respectively. Specifically, a controls the steepness of the sigmoid function, and a higher value of a results in higher confidence in prediction, but it may also increase the risk of vanishing gradient. b determines the density threshold used to distinguish water from objects. With m_o and m_w , we can express $\sigma_\lambda(\mathbf{x})$ in the following form:

$$\begin{aligned} \sigma_\lambda(\mathbf{x}) &= m_w(\mathbf{x})(\sigma(\mathbf{x}) + \beta_\lambda) + m_o(\mathbf{x})\sigma(\mathbf{x}) \\ &= \sigma(\mathbf{x}) + m_w(\mathbf{x})\beta_\lambda \end{aligned} \quad (10)$$

In other words, m_o and m_w can be considered as masks on sample points, exposing those in the water and objects to distinct light-transmitting physics.

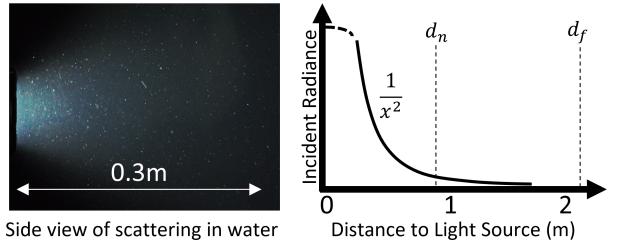


Fig. 3: A side view of scattering generated from an LED light (left) reflects the intensity distribution of incident radiance. We observe significant light fall-off with the distance from the light source. The plot on the right sketches a typical light fall-off curve. d_n and d_f indicates the typically positions of near and far bounding planes. When the distance is close to the dimensions of the lighting component, we need to precisely calibrate the lighting and imaging components to approximate the curve. The rest of the curve can be approximated with the inverse-square law.

D. Approximating Water Effects

The backscatter effects in water can be described using a VSF. However, in learning neural scene representations from real underwater data, we encounter difficulties in modeling VSFs. Firstly, backscatter from the closer regions of the field of view has a greater impact in imaging (Fig. 3). We need a precise imaging system model to accurately infer the VSF in this area. This requires detailed information about the dimensions and poses of the camera and light source. However, calibrating such a system complicates the deployment of our algorithm on real robots and is hard to scale across different robot platforms. Secondly, estimating the VSF along the ray prevents us from using bounding planes, which could significantly enhance the sampling efficiency and avoid overfitting by constraining the viewing frustum from multiple views. To address the issues mentioned above, we propose several approximations to avoid modeling VSFs:

1) *Backscatter as a constant* : The backscatter captured in the image can be approximated as a constant S_λ , as the majority of backscatter comes from the region close to the light source, which is not affected when the images are taken from different depths and perspectives (see Fig. 3).

2) *Co-centered camera and light source*: Points are only sampled between the near and far bounding planes, and their distances to the camera are sufficiently large compared to the typical dimensions of imaging system components. Therefore, we model the light source as a single point light source that is co-centered with the camera, similar to [12]. We use the inverse-square law to calculate the incident radiance $E_\lambda(\mathbf{x})$.

We design a loss function that enforces the model to output $\sigma(\mathbf{x}) = 0$ if \mathbf{x} is in water (see III-F). With this constraint, we are able to avoid double counting backscatter with both S_λ and Eq. 2 since the integrand in Eq. 2 will have zero values for \mathbf{x} in the water. Additionally, constraining $\sigma(\mathbf{x}) = 0$ for \mathbf{x} in water allows us to calculate the attenuation between the near bounding plane and the camera without sampling

points. As a parameter to be optimized in training, β_λ will approach $\sigma_\lambda(\mathbf{x})$ when $\sigma(\mathbf{x})$ approaches 0 according to Eq. 7. Then the transmittance between the near bounding plane and the camera will be $T_\lambda^n = \exp(-\beta_\lambda d_n)$ according to Eq. 6, and Eq. 2 can be written as:

$$L_\lambda(\mathbf{o}, \omega) = S_\lambda + T_\lambda^n \int_{t=d_n}^{d_f} T_\lambda(\mathbf{x}) \sigma(\mathbf{x}) l_\lambda(\mathbf{x}) dt \quad (11)$$

Here d_n and d_f are the distances from the camera to near and far bounding planes respectively.

E. Ray Marching

We numerically estimate Eq. 11 by ray marching. Rays are sampled from the center of the camera and pass through uniformly sampled points on the image plane in training. Points are then sampled along the ray between the near and far bounding planes. The rendering equation is discretized as follows:

$$\begin{aligned} L_\lambda(\mathbf{o}, \omega) &= S_\lambda + T_\lambda^n \sum_{i=0}^N T_\lambda(x_i) \Phi_\lambda(x_i) l_\lambda(x_i) \\ T_\lambda(x_i) &= \exp\left(-\sum_{j=0}^i \sigma_\lambda(x_j) \delta_j\right) \\ \Phi_\lambda(x_i) &= \frac{\sigma(x_i)}{\sigma_\lambda(x_i)} (1 - \exp(-\sigma_\lambda(x_i) \delta_i)) \\ l_\lambda(x_i) &= T_\lambda^n T_\lambda(x_i) E_\lambda(x_i) \alpha_\lambda \cos(\mathbf{n}(x_i), \omega) \end{aligned} \quad (12)$$

where δ_i denotes the step size at sample point x_i . It is worth noticing that transmittance terms T_λ^n and $T_\lambda(x_i)$ are used in both the calculation of the incident radiance l_λ and the sensed radiance L_λ according to approximation III-D.2.

The opacity Φ_λ corresponds to the term $1 - \exp(\sigma(x_i) \delta_i)$ in NeRF and its variants. In NeRF, the volume density σ governs both the emission and attenuation of the radiance, making it sufficient to model objects in the air, haze, and even transparent glowing gas [27]. In our study, we need to model the wavelength-dependent attenuation, which requires both the volume density σ and the attenuation coefficient σ_λ to play a role together in Φ_λ . However, if σ_λ in the denominator approaches 0 in training, the model will encounter numerical issues. To avoid this, we take advantage of our proposition in III-D that enforces $\sigma(x_i) = 0$ if x_i is in the water, so $\Phi_\lambda(x_i) = 0 = 1 - \exp(-\sigma(x_i) \delta_i)$. When x_i falls on objects, $\sigma_\lambda(x_i) = \sigma(x_i)$ according to Eq. 10, so $\Phi_\lambda(x_i) = 1 - \exp(-\sigma(x_i) \delta_i)$. We then simplify $\Phi_\lambda(x_i)$ into the following form, which is identical to the opacity term in NeRF [14]:

$$\Phi_\lambda(x_i) = 1 - \exp(-\sigma(x_i) \delta_i) \quad (13)$$

F. Loss Function

We use L_2 loss to optimize the rendered radiance with captured pixel values from the raw image, which has linear color. As a result, the L_2 loss will be dominated by errors in the brighter parts of the image, and the darker parts will have low rendering quality. To achieve better visual results, we apply a stronger penalization on errors in the darker parts of the image by tone-mapping ψ on both the model output and

raw pixel values before passing them into the loss function as suggested by [13]:

$$\mathcal{L} = \sum_\lambda \sum_{r \in R} \|\psi(\hat{L}_\lambda(r)) - \psi(L_\lambda(r))\|_2^2 \quad (14)$$

Here R is the sampled ray batch, \hat{L} is the raw pixel value and L is the radiance predicted from the model. We use the gamma correction proposed in [28] as our ψ function to map the linear color to sRGB space.

As proposed in III-D, we want to constrain the volume density $\sigma(\mathbf{x}) = 0$ for \mathbf{x} in the water. We first set $\sigma(\mathbf{x}) = 0$ for \mathbf{x} in water by multiplying $m_o(\mathbf{x})$. This gives us the refined volume density $\bar{\sigma}(\mathbf{x})$:

$$\bar{\sigma}(\mathbf{x}) = m_o(\mathbf{x}) \sigma(\mathbf{x}) \quad (15)$$

Then we are able to calculate the refined radiance $\bar{L}_\lambda(r)$ with equations in III-E using $\bar{\sigma}(\mathbf{x})$ in the place of $\sigma(\mathbf{x})$. The refined loss is calculated similarly to Eq. 14:

$$\bar{\mathcal{L}} = \sum_\lambda \sum_{r \in R} \|\psi(\hat{L}_\lambda(r)) - \psi(\bar{L}_\lambda(r))\|_2^2 \quad (16)$$

The total loss is $\mathcal{L}_{total} = \mathcal{L} + \bar{\mathcal{L}}$. By optimizing \mathcal{L}_{total} , we are encouraging the model to generate the same results with $\sigma(\mathbf{x})$ and $\bar{\sigma}(\mathbf{x})$. So the prediction of $\sigma(\mathbf{x})$ from network will converge to $\bar{\sigma}(\mathbf{x})$, where for \mathbf{x} in the water, $\sigma(\mathbf{x}) = 0$.

G. Re-rendering with True Color

To re-render the image with true color, we just need to remove the backscatter S_λ , wavelength-dependent absorption β_λ and volume density $\sigma(\mathbf{x})$ for \mathbf{x} in water. We only need to use $\bar{\sigma}(\mathbf{x})$ in calculating transmittance T and opacity Φ . The rendering equation in III-E becomes the following:

$$\begin{aligned} L_\lambda(\mathbf{o}, \omega) &= \sum_{i=0}^N T(x_i) \Phi(x_i) l_\lambda(x_i) \\ T(x_i) &= \exp\left(-\sum_{j=0}^i \bar{\sigma}(x_j) \delta_j\right) \\ \Phi(x_i) &= 1 - \exp(\bar{\sigma}(x_i) \delta_i) \\ l_\lambda(x_i) &= T(x_i) E_\lambda(x_i) \alpha_\lambda \cos(\mathbf{n}(x_i), \omega) \end{aligned} \quad (17)$$

IV. EXPERIMENTS

A. Dataset

We collect our underwater data in a water tank with 1.3m water depth. Our imaging system consists of a Sony ILCE-7M3 camera with a 40mm prime lens and LED lights. The maximum distance between lights and the camera does not exceed 20cm and their centerlines are parallel to each other. The imaging system is housed in a waterproof case and fully submerged when collecting data. The images are captured using 1/250s exposure time, f/5.6 aperture, and ISO 1600. The raw image files with 14-bit pixel values in HDR space are decoded, denoised, and scaled into 8-bit images with linear values using RawPy [29]. We placed artificial decorations with various colors on the bottom of the tank together with a Macbeth ColorChecker [30]. We use the manufacturer's (X-rite) software to balance the image color as ground truth, which is only used for comparison

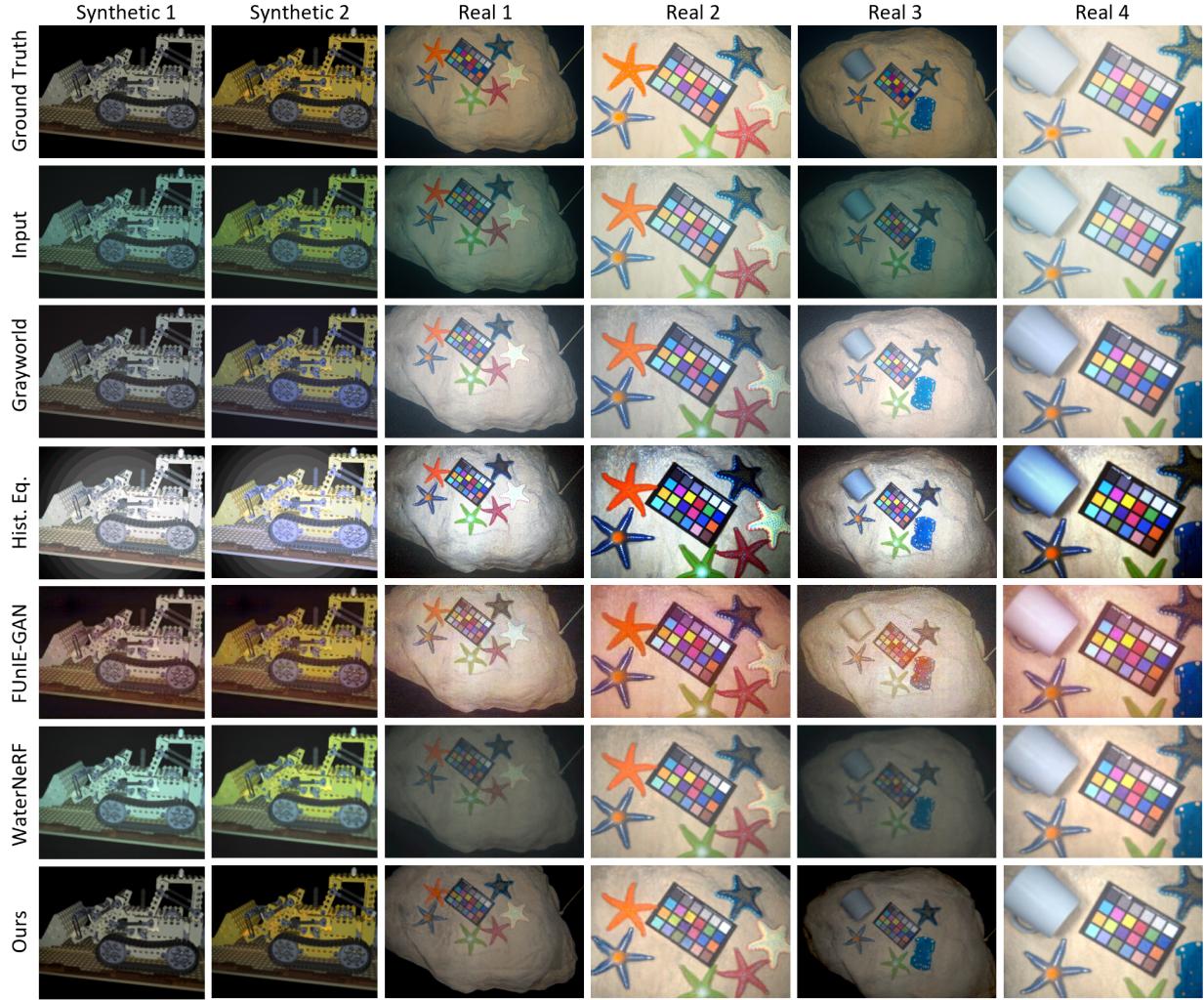


Fig. 4: Visualizations of color restoration. For good visualization quality, real images are visualized in sRGB space.

purposes and does not play a role in our proposed algorithm. We acquire camera poses from COLMAP [31] with post-processed JPEG images to ensure high feature quality.

We also build our synthetic data based on implementations from [8], [32] and measurements from [7], [26]. The ground truth color is obtained by rendering images without any water effects. Although synthetic data may not be sufficient in reflecting complex underwater lighting effects, it is useful in demonstrating that our method is able to decompose different underwater image formation components from each other. In addition, we are able to obtain absolute ground truth from the synthetic dataset by rendering with the same illumination setup, whereas calibrating the color in a real-world image by the ColorChecker could change the brightness level in the image. Both synthetic and real-world data are displayed in the second row of Fig. 4.

B. Implementations

Our code is developed using PyTorch3D Library [33]. We use hash encoding proposed in Instant-NGP [21] for positional encoding. We choose $a = 3$ and $b = 3$ empirically for our sigmoid function in Eq. 9. Our neural implicit function

consists 3 sub-MLPs predicting σ , α , and \mathbf{n} respectively similar to S^3 -NeRF [15]. We use LeakyReLU as activation functions between consecutive linear layers and SoftPlus as the final layer in predicting σ and α to guarantee non-negative outputs.

The model is trained on an Nvidia RTX 4090 GPU with 24GB memory. In each training iteration, we sample 1000 rays from one image and 100 points on each ray. The model is trained for 50k epochs for each scene.

C. Comparisons

We compare our results on both synthetic and real-world data with grayworld algorithm [17], histogram equalization [16], FUnIE-GAN [2] and WaterNeRF [11] (we use open-sourced Sinkhorn loss implementation from GeomLoss Library [34]). The color restoration results are shown in Fig. 4. Grayworld algorithm and histogram equalizing algorithm only correct color well on Synthetic 1 data sequence, in which the object's albedo is dominated by low-saturation colors. Under such circumstances, the grayworld and histogram-equalizing assumptions align well with the underlying color distribution of the scene, so they are able to generate good

TABLE I: MSE in CIELAB Space ↓ (pixel values ranges 0-255)

	Synthetic 1			Synthetic 2			Real 1			Real 2			Real 3			Real 4		
	L	A	B	L	A	B	L	A	B	L	A	B	L	A	B	L	A	B
Grayworld	110.3	4.664	19.72	104.4	22.14	91.36	96.65	10.70	82.34	79.05	15.13	109.59	91.44	10.72	77.05	91.51	12.40	34.01
Hist. Eq.	124.1	5.939	23.72	124.6	21.47	63.45	78.51	15.69	87.02	108.6	43.64	110.3	73.43	20.84	85.68	113.1	39.67	69.64
FUnIE-GAN [2]	108.4	75.49	29.99	106.2	61.18	36.13	62.90	33.81	61.46	83.95	97.36	49.45	87.62	24.91	76.29	96.89	89.23	54.03
WaterNeRF [11]	120.3	55.65	10.26	117.6	60.17	13.72	88.68	20.86	77.59	79.61	15.42	31.05	90.80	13.51	82.91	84.72	24.79	28.46
Ours	49.73	1.146	2.390	42.36	4.076	9.012	60.50	9.678	42.49	78.44	19.18	30.40	73.02	10.29	56.85	84.86	13.56	22.05

TABLE II: Angular Error in sRGB Space ↓ (radians)

	Syn. 1	Syn. 2	Real 1	Real 2	Real 3	Real 4
Grayworld	0.0724	0.2186	0.1381	0.0962	0.1351	0.0475
Hist. Eq.	0.0758	0.2482	0.1421	0.1916	0.1352	0.1931
FUnIE-GAN [2]	0.1107	0.1166	0.1221	0.1655	0.2056	0.1597
WaterNeRF [11]	0.1403	0.1748	0.1303	0.0596	0.1408	0.0567
Ours	0.0361	0.0458	0.0837	0.0591	0.1136	0.0412

results. However, when we change the body color of the bulldozer to bright yellow (Synthetic 2), grayworld algorithm and histogram equalization are getting downgraded as their assumptions fail. We can observe the same in real images 1-4, where the albedo of the scene is dominated by a sand-color rock. Grayworld and histogram equalization algorithms both tend to balance it into grey color. We also observe that both predictions from grayworld and histogram equalization algorithm unpredictably add more veiling light effects into the raw image, as shown in the synthetic data for histogram equalization and real data for grayworld algorithm.

As one of the latest GAN-based methods, FUnIE-GAN is pre-trained on annotated underwater images. In our experiments, we find FUnIE-GAN overshooting in the red channel as shown in Fig. 4, implying that color distributions in their training data are less red than ours. In other words, instead of naive assumptions such as histogram equalization, GAN-based methods learn a color distribution from pre-collected datasets. The inherent color distribution in the data for pretraining can deviate from observations as well. Overall, the results from FUnIE-GAN reflect the fact that methods relying on pretraining will have problems when generalized to scenes with different underlying color distributions.

WaterNeRF tackles the problem by applying the physical constraints from Jaffe-McGlamery model while approaching the histogram-equalized image. We acknowledge that it's not a fair comparison since WaterNeRF works for any kind of illumination while our algorithm and data are only for situations where the light source moves with the camera as a rigid body. We observe that when the histogram-equalized image is flawed, e.g. with our synthetic data, the performance of WaterNeRF can be significantly downgraded. We also find our method outperforms WaterNeRF in color consistency on real data. For example, comparing Real 1 and 2 images in Fig. 4, which is from the same image sequence, our method restores the color of the rock with better consistency since we model the albedo and light reflection of the scene while WaterNeRF models the scene with constant radiance, which fails when the light source moves. In general, from the comparisons, our method restores color in both synthetic and

real-world data with the most consistent performance.

We present two metrics for quantitative evaluation: mean-squared error (MSE) of each CIELAB channel (Table I) and mean angular error [11] in the sRGB space (Table II). CIELAB is designed to approximate human vision in a uniform space [35] and sRGB is the standard colorspace in which the image is presented. For the synthetic dataset, we use the ground truth from the renderer and calculate both metrics directly. For real data, since color corrected with calibration software changes the brightness in images, we scale the images for comparison to have the same brightness before we calculate the MSE of each LAB channel.

As revealed by MSE (Table I), our method performs the best on synthetic data on all LAB channels. While among the 4 real images evaluated, our method performs best on Real 1 and Real 3 images, which exhibit heavier water effects. However, on Real 2 and Real 4 images with less distortion, our method performs slightly weaker or comparably to other approaches. This suggests that our method maintains consistency as water effects increase with altitude, while other approaches may experience greater degradation.

Besides evaluating LAB channels separately, angular error (Table II) reflects the color similarity in the entire RGB space. Results show that our method performs the best across all data, which is consistent with the visualizations in Fig. 4.

Nevertheless, it's important to note that the error in pixel values is not only from the deviation of color but also the structural quality of image reconstruction. For example, grayworld-corrected images will retain all the features while images reconstructed with our method are subject to loss of details due to errors in pose estimation, refraction in water, lens effects, and approximations, etc.

V. DISCUSSIONS

This worked is directly applicable to underwater imagery collected when dominant light sources move with the camera as a rigid body, such as in deep water, ice-covered water, or cave water. However, it may fail in the following scenarios:

- When the light source is a combination of onboard strobes (point light sources), natural light and ambient light, our model is inadequate for accurately representing water effects from mixed light sources.
- In the presence of highly turbid and layered water, scattering effects vary more significantly with depth, and the robot will have to observe the scene at a closer range (breaking approximation III-D.1). Modeling backscatter as a constant could potentially lead to failure.

- When the baseline between the camera and onboard light source is long, creating shadows in the observed scene, our model, which assumes co-centered light and camera, cannot accurately represent shadows (breaking approximation III-D.2). This issue also arises with robots equipped with multiple cameras or light sources.

VI. CONCLUSIONS

This work proposes a unified framework that learns underwater neural scene representations together with water effects. We demonstrate that our method is able to restore the true color of the underwater scene with a sequence of observations from different ranges and perspectives. By approximating the backscatter and simplifying the ray tracing, we avoid estimating VSF, which is numerically unstable and requires precise calibration of lighting and imaging system. Additionally, our proposed method generates dense results with end-to-end differentiability and does not rely on any pre-training or assumptions on prior color distributions.

Future work will extend our model to address the issues discussed in V. Our long-term goal is to achieve true color correction for all types of underwater lighting conditions.

REFERENCES

- [1] T. Manderson, J. Li, N. Dudek, D. Meger, and G. Dudek, “Robotic coral reef health assessment using automated image analysis,” *Journal of Field Robotics*, vol. 34, no. 1, pp. 170–187, 2017.
- [2] M. J. Islam, Y. Xia, and J. Sattar, “Fast underwater image enhancement for improved visual perception,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3227–3234, 2020.
- [3] J. Jaffe, “Computer modeling and the design of optimal underwater imaging systems,” *IEEE Journal of Oceanic Engineering*, vol. 15, no. 2, pp. 101–111, 1990.
- [4] C. D. Mobley, *Light and water: radiative transfer in natural waters*. Academic press, 1994.
- [5] R. M. Pope and E. S. Fry, “Absorption spectrum (380–700 nm) of pure water. ii. integrating cavity measurements,” *Appl. Opt.*, vol. 36, no. 33, pp. 8710–8723, 1997.
- [6] B. L. McGlamery, “A Computer Model For Underwater Camera Systems,” in *Ocean Optics VI*, S. Q. Duntley, Ed., International Society for Optics and Photonics, vol. 0208, SPIE, 1980, pp. 221 –231.
- [7] T. J. Petzold, “Volume scattering functions for selected ocean waters,” Scripps Institution of Oceanography La Jolla Ca Visibility Lab, Tech. Rep., 1972.
- [8] Y. Song, D. Nakath, M. She, F. Elibol, and K. Köser, “Deep sea robotic imaging simulator,” in *Pattern Recognition. ICPR International Workshops and Challenges*, A. Del Bimbo, R. Cucchiara, S. Sclaroff, G. M. Farinella, T. Mei, M. Bertini, H. J. Escalante, and R. Vezzani, Eds., Cham: Springer International Publishing, 2021, pp. 375–389.
- [9] M. Bryson, M. Johnson-Roberson, O. Pizarro, and S. B. Williams, “True color correction of autonomous underwater vehicle imagery,” *Journal of Field Robotics*, vol. 33, no. 6, pp. 853–874, 2016.
- [10] J. Li, K. A. Skinner, R. M. Eustice, and M. Johnson-Roberson, “Watergan: Unsupervised generative network to enable real-time color correction of monocular underwater images,” *IEEE Robotics and Automation Letters*, vol. 3, pp. 387–394, 2017.
- [11] A. V. Sethuraman, M. S. Ramanagopal, and K. A. Skinner, “Waternerf: Neural radiance fields for underwater scenes,” *ArXiv*, vol. abs/2209.13091, 2022.
- [12] S. Bi, Z. Xu, P. Srinivasan, B. Mildenhall, K. Sunkavalli, M. Hašan, Y. Hold-Geoffroy, D. Kriegman, and R. Ramamoorthi, *Neural reflectance fields for appearance acquisition*, 2020.
- [13] B. Mildenhall, P. Hedman, R. Martin-Brualla, P. P. Srinivasan, and J. T. Barron, “NeRF in the dark: High dynamic range view synthesis from noisy raw images,” *CVPR*, 2022.
- [14] 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 *ECCV*, 2020.
- [15] W. Yang, G. Chen, C. Chen, Z. Chen, and K.-Y. K. Wong, “S³-nerf: Neural reflectance field from shading and shadow under a single viewpoint,” in *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.
- [16] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, “Adaptive histogram equalization and its variations,” *Computer vision, graphics, and image processing*, vol. 39, no. 3, pp. 355–368, 1987.
- [17] G. Buchsbaum, “A spatial processor model for object colour perception,” *Journal of the Franklin institute*, vol. 310, no. 1, pp. 1–26, 1980.
- [18] K. He, J. Sun, and X. Tang, “Single image haze removal using dark channel prior,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1956–1963.
- [19] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, “Mip-nerf 360: Unbounded anti-aliased neural radiance fields,” *CVPR*, 2022.
- [20] Y. Schechner and N. Karpel, “Recovery of underwater visibility and structure by polarization analysis,” *IEEE Journal of Oceanic Engineering*, vol. 30, no. 3, pp. 570–587, 2005.
- [21] T. Müller, A. Evans, C. Schied, and A. Keller, “Instant neural graphics primitives with a multiresolution hash encoding,” *arXiv:2201.05989*, Jan. 2022.
- [22] M. Pharr, W. Jakob, and G. Humphreys, *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.
- [23] J. Fong, M. Wrenninge, C. Kulla, and R. Habel, “Production volume rendering: Siggraph 2017 course,” in *ACM SIGGRAPH 2017 Courses*, 2017, pp. 1–79.
- [24] D. Akkaynak, T. Treibitz, T. Shlesinger, Y. Loya, R. Tamir, and D. Iluz, “What is the space of attenuation coefficients in underwater computer vision?” In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 568–577.
- [25] D. Akkaynak and T. Treibitz, “A revised underwater image formation model,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6723–6732.
- [26] N. G. Jerlov, *Marine optics*. Elsevier, 1976.
- [27] N. Max, “Optical models for direct volume rendering,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, no. 2, pp. 99–108, 1995.
- [28] Adobe Systems Incorporated, *Inverting the color component transfer function*, <https://www.adobe.com/digitalimag/pdfs/AdobeRGB1998.pdf>.
- [29] M. Riechert, *Rawpy*, <https://pypi.org/project/rawpy/>, 2023.
- [30] C. S. McCamy, H. Marcus, and J. G. Davidson, “A colorrendition chart,” *J. Appl. Photogr. Eng.*, vol. 2, no. 3, pp. 95–99, 1976.
- [31] J. L. Schönberger and J.-M. Frahm, “Structure-from-motion revisited,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [32] L. Mossberg, *Monte Carlo Ray Tracer*, <https://github.com/linusmossberg/monte-carlo-ray-tracer>, 2022.

- [33] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari, “Accelerating 3d deep learning with pytorch3d,” *arXiv:2007.08501*, 2020.
- [34] J. Feydy, T. Séjourné, F.-X. Vialard, S.-i. Amari, A. Trouve, and G. Peyré, “Interpolating between optimal transport and mmd using sinkhorn divergences,” in *The 22nd International Conference on Artificial Intelligence and Statistics*, 2019, pp. 2681–2690.
- [35] “Image technology colour management-architecture, profile format and data structure-part 1: Based on icc.1:2010,” ISO, Tech. Rep. 15076-1, 2010.