Neural Underwater Scene Representation

Yunkai Tang[1,2†] Chengxuan Zhu[3†] Renjie Wan[4*] Chao Xu[3] Boxin Shi[1,2*]

[1]National Engineering Research Center of Visual Technology, School of Computer Science, Peking University

[2]National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University

[3]National Key Lab of General AI, School of Intelligence Science and Technology, Peking University

[4]Department of Computer Science, Hong Kong Baptist University

tangyunkai@stu.pku.edu.cn, peterzhu@pku.edu.cn, renjiewan@hkbu.edu.hk,
xuchao@cis.pku.edu.cn, shiboxin@pku.edu.cn

https://freebutuselesssoul.github.io/uwnerf

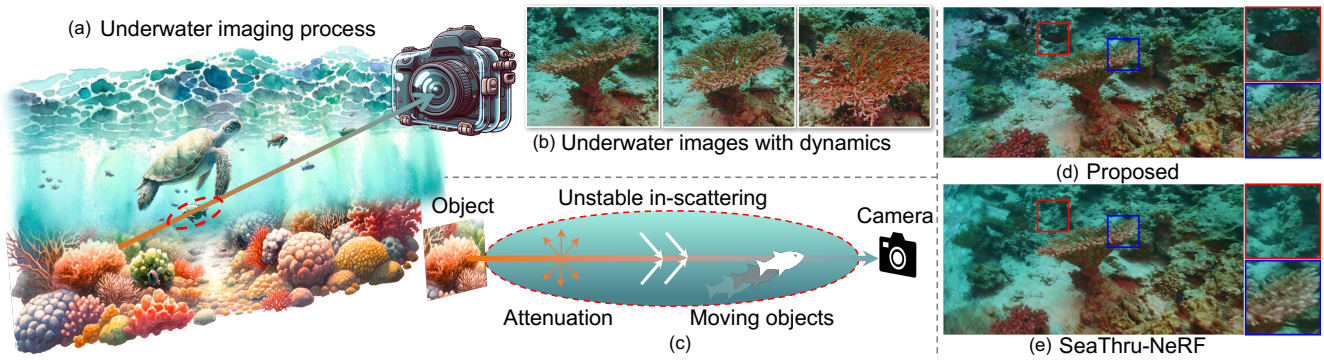


Figure 1. The underwater image capturing process is shown in (a), where an underwater camera captures 3 exemplar images in (b), demonstrating the attenuation, unstable in-scattering and moving objects during light transport as depicted in (c). We then show a comparison of underwater scene modeling in (d) and (e), between the proposed method and SeaThru-NeRF [15].

Abstract

Among the numerous efforts towards digitally recovering the physical world, Neural Radiance Fields (NeRFs) have proved effective in most cases. However, underwater scene introduces unique challenges due to the absorbing water medium, the local change in lighting and the dynamic contents in the scene. We aim at developing a neural underwater scene representation for these challenges, modeling the complex process of attenuation, unstable in-scattering and moving objects during light transport. The proposed method can reconstruct the scenes from both established datasets and in-the-wild videos with outstanding fidelity.

1. Introduction

Neural Radiance Fields (NeRF) achieves promising performance in representing the scenes above the surface of the water [23, 25]. *Can NeRF still model the scenes under water properly?*

The answer is negative, due to the widespread "dynamic" phenomena underwater, while vanilla NeRF [23] is tailored for static scenes only. First, an object becomes more and more difficult to be observed as its distance to the camera increases, due to the absorbing property of water. This leads to the foremost dynamic factor in underwater scenes, that is ① **distance-dependent visibility**. Next, the scattering effect of water and the changeable lighting condition introduce another dynamic factor. The illumination observed from different views are thus time-varying, leading to ② **unstable illumination**. Moreover, the underwater ecosystem, teeming with marine life, leads to the third dynamic. The ③ **moving objects** such as marine plants and animals challenge the static assumptions inherent to standard NeRF models. The synergy of the aforementioned dynamic factors creates an intricate and multifaceted environment that is difficult for the current NeRF models to comprehend and represent accurately. Thus, finding a way to manage such an environment is crucial for the neural representation of underwater scenes.

Existing underwater image processing methods [1, 7, 14] aim at removing the effects of water blending, and simplify the underwater imaging model with priors, such as the dark

†Equal contribution.
*Corresponding authors.

channel prior [14], haze line prior [7], *etc.* However, most of them deal with a single image, and may not help on alleviating the curse made by dynamic factors, given that each image is just one static slice in time of a dynamic process. A more reasonable approach would be to factor in dynamics when constructing NeRF models [11, 27, 33], utilizing NeRF variants designed for dynamic objects. However, those methods designed for scenes above the surface of the water assume dynamic scenes with constant illumination and minimal attenuation. If they are applied directly to underwater scenes, challenges ① and ② can quickly invalidate their assumptions about static properties, thereby compromising the performance in scene representation.

SeaThru-NeRF [15] has been proposed recently, trying to model the distance-dependent attenuation from multiple viewpoints. However, it ignores ② changing illumination and ③ dynamic objects in the scene, and also their intertwined relationship, resulting in blurry and floater artifacts, as shown in Fig. 1. Besides, SeaThru-NeRF [15] conditions the water body on viewing direction, adding to the complexity of the processing pipeline. Their main goal, which is to remove the water effects and "see through", also slightly deviates from our purpose of modeling underwater scene.

Faced with the aforementioned issues, in this paper we extend neural radiance fields to handle *scenes under water*. We propose to treat the water as an object with semi-transparent property, and optimize the water parameters jointly with the objects in the scene. Instead of omitting the "empty" space between the scene and the camera in vanilla NeRF setting [18, 23, 35], or using a straight-forward image blending model in 2D space [1, 7], the proposed model solves the challenge of ① distance-dependent attenuation through a 3D formulation that also takes the scattering water medium into consideration. Besides, we design an illumination field operating in the logarithmic space and a self-adaptive tone mapper module that model the ② unstable illumination in the scene. The ③ moving objects are optimized after the first stage where static part and water body is reconstructed, enabling a higher efficiency, since only the moving objects in front of the static counterparts contribute to the rendered result.

In brief, our contributions can be summarized as follows:

- a simple physics-based model to simulate the distance-related attenuation of the water medium;
- an illumination field and a self-adaptive tone mapper module that mimic the unstable illumination observed underwater, preventing the system from naively modeling illumination as motion; and
- a separated reconstruction scheme, composed of a static branch aiming at the still structures, and a dynamic branch for moving objects, leading to a more robust optimization.

With these features integrated, the proposed method not only outperforms the state-of-the-art methods quantitatively in scene representation, but also produces more realistic results in qualitative evaluation. The proposed method also enables the editing of underwater scenes, such as draining the water and transferring the water effect to non-underwater scenes.

2. Related work

Neural radiance fields for dynamic scenes. Neural Radiance Fields [23] (NeRF) emerged as a significant development for novel view synthesis, by constructing an implicit, neural network-based scene representation. Though NeRF is originally designed for static scenes, the community has sought for numerous methods to model the inconsistency across frames.

Nerfies [26] and HyperNeRF [27] apply a deformation field to map the observation to a canonical scene representation, and optimizes a per-frame latent. They can handle unstructured videos, but are limited to object-centric poses. NeRF-in-the-wild [21] also uses a per-frame latent, enabling handling of diverse illumination and appearance in the input, while requiring hundreds of images to robustly optimize the latent space.

Unlike these frame-based methods that optimize a latent for each view [21, 26, 27], time-based NeRFs try to encode spatio-temporally varying scene volumetrically. They take time step, position, and viewing direction as the input of the neural network, greatly increasing the capability as well as complexity. Several insightful ideas have been proposed to blend time into the network. NSFF [16] designs the neural scene flow fields that can handle complex and fast motion, DynIBaR [17] gathers the warped frames from temporally nearby frames to perform realistic rendering, MonoNeRF [30] learns a velocity field to further imitate the temporal consistency. It is also observed that optical flow provides useful hints for reconstruction [11, 19, 30].

Underwater imaging. Analyzing the images taken underwater has become a hot topic for computer vision community for the past decades. Removing the effects of water requires a physics-based modeling of the scene. Light propagation in a medium is characterized by radiative transfer equation [8], but the complete computation requires Monte Carlo simulation, which is prohibitively expensive for real-time rendering. Many physics-based methods use certain priors to simplify the problem, such as dark channel prior [14], white balance [3], or haze line prior [6], and then separate the back-scatter and transmission part. Some alleviate the problem for just one water type with a fixed attenuation coefficient [10, 20]. Some also try removing the degradation from images through data-driven optimization process [13]. However, most efforts mentioned only try to remove the water effects from 2D images, which is nevertheless an ill-posed problem. A more physics-grounded model with respect to light propagation is proposed [1, 2]

for underwater image restoration, but still requires known depth information. To this point, an inherent 3D representation underwater for a physics-based simulation is desired.

Neural radiance fields for scattering medium. A growing number of works have found the straight-forward modeling in vanilla NeRF unable to recreate real-world phenomena, such as reflection [12, 31, 38], occlusions [21, 39], and attenuation when propagating in the media [9, 15, 29]. Previous works introduced physics-grounded model into NeRF [15, 29], but still naively rely on a combination of water scattering and opaque object surface, which prevents them from considering the time-varying effects as discussed before. We attribute their concession to the over-complicated physical model. In this paper, the water properties are simplified, and the previously ignored changing illumination and moving objects are considered, which is suitable for underwater structure restoration, accessible scene editing and realistic graphics rendering.

3. Methods

3.1. Problem formulation

The scene representation of Neural Radiance Fields (NeRFs) is essentially a multi-layer perceptron (MLP) $f : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$ that maps the position $\mathbf{x}$ and viewing direction $\mathbf{d}$ to the point's color $\mathbf{c}$ and density σ. To render the color of a ray hitting the camera requires accumulating the points along the ray parameterized by $\mathbf{r}(t) = \mathbf{o} + t \cdot \mathbf{d}$, where $\mathbf{o}$ is the position of the camera and $t > 0$. The color $\mathbf{C}(\mathbf{r})$ is formulated as

$$\mathbf{C}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(t)\mathbf{c}(t)dt, \tag{1}$$

where t_n and t_f are the near and far bounds of rendering, $\sigma(t)$ and $\mathbf{c}(t)$ refer to the density and color at $\mathbf{r}(t)$. $T(t)$ denotes the accumulated transmittance from t_n to t, namely $T(t) = \exp(-\int_{t_n}^{t} \sigma(s)ds)$.

The vanilla NeRF employs a single branch to represent the whole scene. However, due to the moving objects underwater, we consider modeling the underwater scene by separating the whole scene into a static branch $f_{\text{sta}} : (\mathbf{x}) \rightarrow (\mathbf{c}_{\text{sta}}, \sigma_{\text{sta}})$ and a dynamic branch $f_{\text{dyn}} : (\mathbf{x}, t) \rightarrow (\mathbf{c}_{\text{dyn}}, \sigma_{\text{dyn}})$. The two branches following the settings in DynamicNeRF [11] can be optimized by the reconstruction error, given by

$$\mathcal{L}_{\text{recon}} = \sum_{s \in \{\text{sta,dyn}\}} \sum_{\mathbf{r} \in \mathbf{M}_s} \|\hat{\mathbf{C}}_s(\mathbf{r}) - \mathbf{C}_s(\mathbf{r})\|_2, \tag{2}$$

where $\mathbf{C}_{\text{sta}}(\mathbf{r})$ and $\mathbf{C}_{\text{dyn}}(\mathbf{r})$ are the rendered color of the static branch and the dynamic branch respectively, and $\hat{\mathbf{C}}$ is the color of ground truth; $\mathbf{M}_{\text{sta}}$ and $\mathbf{M}_{\text{dyn}}$ are the binary mask for static area and moving objects, estimated by off-the-shelf optical flow models [34]. Via Eq. (2), the static branch and dynamic branch are progressively optimized without affecting each other.

However, solely relying on Eq. (2) does not yield satisfactory results, with the outcomes exhibiting various artifacts. This primarily stems from its limitations in addressing the other two dynamics discussed in Sec. 1, specifically the distance-dependent visibility and unstable illumination.

In the model of vanilla NeRF [23], to effectively calculate Eq. (1) by sampling and summation, NeRF model is expected to skip the "empty" space that contributes less to the rendered color [5, 23, 35]. However, in underwater scenes, the visibility of objects is considerably influenced by the water medium all over the space. Skipping the water medium can lead to degeneration in the rendered results, by encouraging NeRF to falsely punish the distance-dependent visibility caused by water on the color change of objects. We propose and justify a volume rendering model that can characterize the distance-dependent visibility, detailed in Sec. 3.2. In addition, we find the previously proposed sampling strategy [5, 23] unable to focus on the water medium and the objects simultaneously, and tailor a progressive sampling strategy for underwater scenes in Sec. 3.3.

Moreover, Eq. (2) focuses solely on moving objects, neglecting the challenges posed by unstable illumination that varies across different viewpoints. The varying illumination only changes the exposure locally, but does not alter the location or actual color of objects. Eq. (2) is thus not sufficient since illumination can change in static areas too. Modeling both the changing illumination and moving objects with dynamic branch is under-constrained To effectively disentangle the multiple dynamics under water, in Sec. 3.4 we propose an illumination branch $f_{\text{I}} : (\mathbf{x}, \mathbf{d}, t) \rightarrow \lambda$ to model the locally varying illumination, where $t \in [0, 1]$ is the capturing time step of the image.

3.2. Modeling distance-dependent visibility

As the first challenge comes from the absorbing property of water, we address this issue by considering water as semi-transparent that should be considered during volume rendering. Then, Eq. (1) can be rephrased as follows:

$$\mathbf{T}(t) = \exp\left(-\int_{t_n}^{t} (\boldsymbol{\sigma}_{\text{w}}(s) + \sigma_{\text{obj}}(s))ds\right), \tag{3}$$

where the subscripts of "w" and "obj" denote the water and the objects, respectively. Then, to better model the absorbing effects caused by water, we build a mapping correlation between $\boldsymbol{\sigma}_{\text{w}}$ and the RGB channel during rendering as follows:

$$\mathbf{C} = \int_{t_n}^{t_f} \mathbf{T}(t) \odot (\sigma_{\text{obj}} + \boldsymbol{\sigma}_{\text{w}}) \odot \bar{\mathbf{c}}\, dt, \tag{4}$$

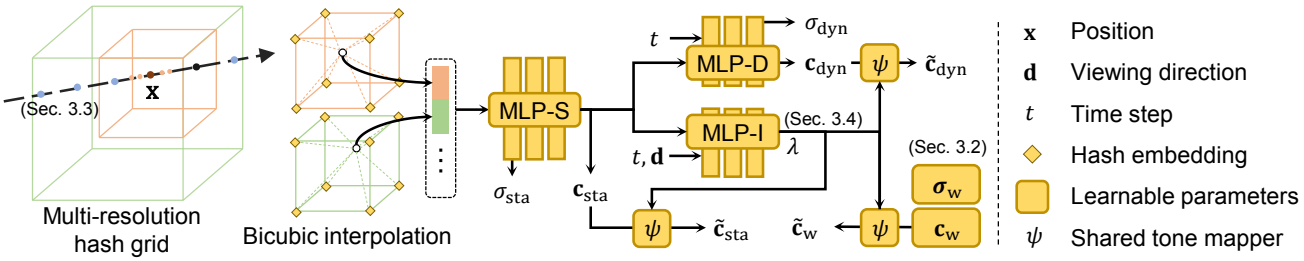


Figure 2. The structure of the proposed method. MLP-S is designed to learn the density and color $(\sigma_{\text{sta}}, \mathbf{c}_{\text{sta}})$ of the static branch f_{sta}, by taking the hash embedding [25] of position $\mathbf{x}$ as input. MLP-D, or the dynamic branch f_{dyn}, conditioned on the features of the positions along with the time step t, aims to model the density and color of the dynamic scene, $(\sigma_{\text{dyn}}, \mathbf{c}_{\text{dyn}})$ for dynamic objects. MLP-I stands for the illumination field f_{I}, which reconstruct the unstable illumination λ, dependent on time and the spherical harmonics encoding [31] of viewing direction $\mathbf{d}$. The tone mapper $\psi(\cdot)$ and water parameters $(\boldsymbol{\sigma}_{\text{w}}, \mathbf{c}_{\text{w}})$ are also optimized when participating in the process of converting network outputs to non-linear colors, and when rendering water effects.

where $\bar{\mathbf{c}}$ is the alpha-composite of $\mathbf{c}_{\text{w}}$ and $\mathbf{c}_{\text{obj}}$, using $\boldsymbol{\sigma}_{\text{w}}$ and σ_{obj} as weights, and $\odot$ stands for element-wise multiplication. Note that $\boldsymbol{\sigma}_{\text{w}}$ is considered as a density triplet of RGB channels, namely the water has different transmittance in different wavelengths.

The density triplet is in accordance with radiative transfer function [8], a physical model used to describe light propagation process, and it can also explain the physics-based model in previous works [1, 2, 7] focusing on image restoration that has the form of

$$\mathbf{C} \approx \mathbf{J} \cdot e^{-\beta^D \cdot z} + \mathbf{B}^{\infty} \cdot (1 - e^{-\beta^B \cdot z}), \tag{5}$$

where $\mathbf{C}$ is the observed color of an underwater surface point, $\mathbf{J}$ is the actual color of the clear scene, β^D and β^B are the coefficients for attenuation and backscatter, dependent on wavelength, z is the depth of the surface point, and $\mathbf{B}^{\infty}$ is the backscatter color at infinity caused by water. Note that Eq. (5) is only an approximation form, as image-based methods do not have 3D representation of the scene. By setting the wavelength-dependent $\beta^D = \boldsymbol{\sigma}_{\text{w}}$, our formulation can reach the exact formulation of underwater scenes.

For tractable computation, the rendering process in Eq. (3) is calculated from summing over the intervals of $[s_i, s_{i+1}]$, where $t_n = s_0 < s_1 < \cdots < s_N = t_f$. As discussed before, in the proposed method we further break down the object part into static and dynamic. The rendered color is calculated as

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=1}^{N} \mathbf{T}_i \odot (1 - e^{-\sigma_{\text{sta},i} - \sigma_{\text{dyn},i} - \boldsymbol{\sigma}_{\text{w}}}) \odot \bar{\mathbf{c}}_i, \tag{6}$$

where σ_i and $\mathbf{c}_i$ is the assumed constant density and color in the interval of $[s_i, s_{i+1}]$, and $\mathbf{T}_i = \exp(-\sum_{j<i} \delta_j (\boldsymbol{\sigma}_{\text{w}} + \sigma_{\text{dyn},i} + \sigma_{\text{sta},i}))$ denotes the transmittance of the three channels, with $\delta_j = s_{j+1} - s_j$ referring to the sampling interval. $\bar{\mathbf{c}}_i$ is the weighted non-linear color of $\tilde{\mathbf{c}}_{\text{sta},i}$, $\tilde{\mathbf{c}}_{\text{dyn},i}$, and $\tilde{\mathbf{c}}_{\text{w},i}$ in the proposed network, formulated as

$$\bar{\mathbf{c}}_i = \boldsymbol{\beta}_{\text{sta},i} \tilde{\mathbf{c}}_{\text{sta},i} + \boldsymbol{\beta}_{\text{dyn},i} \tilde{\mathbf{c}}_{\text{dyn},i} + \boldsymbol{\beta}_{\text{w},i} \odot \tilde{\mathbf{c}}_{\text{w},i}. \tag{7}$$

To allow for a cleaner separation between different objects in the same position, we empirically design the weight function that emphasizes the component with more density, enabling a faster optimization, given by

$$\boldsymbol{\beta}_{\{\text{sta,dyn,w}\},i} = \frac{1}{2} \sin\left(\frac{\pi \cdot \{\sigma_{\text{sta},i}, \sigma_{\text{dyn},i}, \boldsymbol{\sigma}_{\text{w}}\}}{\sigma_{\text{sta},i} + \sigma_{\text{dyn},i} + \boldsymbol{\sigma}_{\text{w}}} - \frac{\pi}{2} \right) + \frac{1}{2}. \tag{8}$$

Note that the outputs of the f_{sta} and f_{dyn} are not fed into Eq. (7) directly, which will be explained in Sec. 3.4.

To encourage the network to learn the correct density of sparse objects in the water medium, we design a loss function that penalizes the ambiguity of objects along the ray, given by

$$\mathcal{L}_{\text{entropy}} = -\sum_{i=1}^{N} \mathbf{w}_i \log(\mathbf{w}_i + \epsilon) \cdot \text{clip}(r_v - k_0, n_0, n_1), \tag{9}$$

where $\mathbf{w}_i$ is the weight of the i-th sampled segment along the ray. r_v is the visible reciprocal ratio, equivalent to the number of training views divided by the number of views where the segment is visible. Note that being "visible" only means that the position falls in the frustum of the i-th view. k_0 is the threshold of visible reciprocal ratio, and "clip" is simply the clipping function, with n_0 and n_1 being the lower and upper bounds of the penalty. ϵ is a small constant to avoid numerical instability.

3.3. Mixed progressive sampling strategy

Though previous works have evolved from importance sampling [23] to sparse voxel grid [18], or even learning-based sampling [4, 5], the underlying rationale is still placing more importance on those sampling points that contribute more, to improve fidelity. However, the influence of water cannot be neglected like NeRF above the surface of the water. To address the unique issue, We introduce a mixed progressive sampling strategy. For each ray $\mathbf{r}$, we initially take N_m uniform samples to approximate the water medium's properties. This is complemented by N_s samples, denoted as $\{t_i \cdot \delta_k\}_{i=1}^{N_s}$, strategically chosen based on

a multi-resolution density grid to represent objects within the scene. These samples adhere to the constraint that their densities on the static branch should exceed a predefined threshold τ:

$$\sigma_{\text{sta}}(\mathbf{o}(\mathbf{r}) + t_i \cdot \delta_k \cdot \mathbf{d}(\mathbf{r})) > \tau, \tag{10}$$

where k indicates the progressive sampling level, and δ_k is the dynamic sampling interval—adjusted as optimization progresses. The variable t_i, chosen from the set of natural numbers $\mathbb{N}$, refers to the i-th segment along the ray, with $t_i < t_{i+1}$ ensuring an orderly progression.

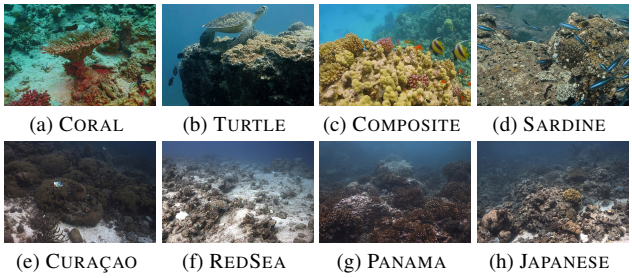
This strategy, integrating uniform and progressive sampling, allows for the simultaneous optimization of the water medium, illumination, and static elements within the scene. To model the dynamic objects, the estimated ray termination depth $D(\mathbf{r})$ from the static scene analysis is utilized. This estimation aids in delineating the boundary for sampling positions, ensuring that dynamic objects are only considered if they precede the static scene in the ray. We uniformly sample N_d points within the range of $[t_n, D(\mathbf{r}) + \epsilon]$, focusing the optimization process on these points for the dynamic branch. During this phase, the parameters of the illumination branch, static branch, and water remain unaltered, ensuring a focused optimization on dynamic elements.

3.4. Illumination fields and tone mapper

The unstable illumination underwater poses another unique challenge. As the illumination changes the appearance of both dynamic and static objects, it cannot be only modeled by the dynamic branch as in Eq. (4). As pointed out by Zhang *et al.* [37], most underwater objects do not change their appearances across different viewpoints. By setting the illumination as a shared factor, attributing the difference in appearance to illumination, a more robust reconstruction of the scene is achieved. We build an illumination field $f_{\text{I}} : (\mathbf{x}, \mathbf{d}, t) \rightarrow \lambda$ to model the unstable illumination, where λ is the value of exposure to be imposed on the static branch and the dynamic branch.

However, simply multiplying λ by the output color of f_{sta} and f_{dyn} is not appropriate to adjust exposure, since the color $\mathbf{c}$ used in Eq. (1) is in non-linear color space, which does not scales by naive multiplication. Therefore, we propose to operate on linear color space, taking advantage of its scaling properties [24]. The raw output color in the proposed design, namely $\mathbf{c}_{\text{sta}}$, $\mathbf{c}_{\text{dyn}}$, and $\mathbf{c}_{\text{w}}$, should thus be in linear color space.

To convert the linear color to non-linear color as the camera does, a tone mapper network $\psi(\cdot)$ is optimized simultaneously, which could derive the non-linear color $\tilde{\mathbf{c}}_{\{\text{sta,dyn,w}\}}$. To optimize robustly and to avoid the multiplication from exploding gradients, we draw inspiration from HDR imaging in computational photography, where log radiance is used so that the multiplication becomes addition, namely



(a) CORAL (b) TURTLE (c) COMPOSITE (d) SARDINE
(e) CURAÇAO (f) REDSEA (g) PANAMA (h) JAPANESE

Figure 3. Samples of the 4 monocular videos in the proposed dataset on the top row, along with samples of the 4 image sequence in SeaThru dataset [15].

changing the naive formulation of $\tilde{\psi}(E \odot \mathbf{c}) = \tilde{\mathbf{c}}$ into

$$\tilde{\psi} \circ \exp(\log(E) + \log(\mathbf{c})) = \tilde{\mathbf{c}}, \tag{11}$$

where $\circ$ denotes the composition operator of functions.

Taken one step further, we propose to use the log color $\mathbf{c}_{\{\text{sta,dyn,w}\}} \in (-\infty, +\infty)$ as the output of the network to adjust the exposure via addition instead of multiplication. The log color is then converted to non-linear color by

$$\psi(\lambda + \mathbf{c}_{\{\text{sta,dyn,w}\}}) = \tilde{\mathbf{c}}_{\{\text{sta,dyn,w}\}}, \tag{12}$$

where $\psi = (\tilde{\psi} \circ \exp)$ replaces the hypothetical tone-mapping function in Eq. (11).

We use a shared tone mapper and illumination network for both the static branch, the dynamic branch and the water medium parameters, since the illumination field is expected to have a global effect at any given position. Besides, the output of illumination fields, $\lambda > 0$, is shared across RGB channels, so that the color is not altered but only scaled by the illumination field.

4. Experiments

We evaluate the proposed method on various underwater monocular videos gathered from the web, and show simulated results of scene editing. The experiment setting and details are introduced in Sec. 4.1. We then evaluate the proposed method on quantitative novel view synthesis task and qualitative evaluation, shown in Sec. 4.2, compared with other state-of-the-art methods, including a user study. We then conduct an ablation study in Sec. 4.3 to validate the components of the proposed design.

4.1. Implementation details

Datasets. The proposed dataset is collected from in-the-wild captured clips from the Internet, consisting of 4 monocular videos. Image samples are provided in Fig. 3.

We also conduct tests in the underwater scenes proposed in SeaThru-NeRF [15], to verify the efficacy of the proposed method. The SeaThru Dataset is composed of sparsely captured images, and has few temporal connections between adjacent frames in the dataset.

Network architecture. The network structure is illustrated in Fig. 2. "MLP-S" and "MLP-I" are both MLPs composed of 2 hidden layers of fully-connected layers with ReLU activation function, while 4 layers for "MLP-D". In addition, $\psi(\cdot)$ is a simple MLP with 1 hidden layer, and $(\boldsymbol{\sigma}_{\mathrm{w}}, \mathbf{c}_{\mathrm{w}})$ stand for the learnable water parameters.

Training details. Experiments are conducted on a single NVIDIA GeForce RTX 3090 GPU. Training the proposed method takes 45 minutes, and rendering a novel view result with a resolution of 1920×1080 takes less than 30 seconds. The Adam optimizer is applied, with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and a learning rate of 0.005 with a cosine scheduler. We use a pre-trained optical flow-based method [34] to generate a motion mask for each image, following the practice of DynamicNeRF [11]. The hyperparameters are set to $k_0 = 3$, $n_0 = 0.1$, $n_1 = 5$.

To locate and reconstruct the static components and sparse moving objects separately, we propose to train the static branch and dynamic branch in a two-stage manner. In the first stage, we jointly optimize f_{I}, f_{sta}, ψ, and the water medium parameters $(\boldsymbol{\sigma}_{\mathrm{w}}, \mathbf{c}_{\mathrm{w}})$, by minimizing the sum of reconstruction loss and entropy loss, as defined in Eqs. (2) and (9) respectively, on static region $\mathbf{M}_{\mathrm{sta}}$ only. The optimization in the second stage is performed on the moving objects with the mask $\mathbf{M}_{\mathrm{dyn}}$, as the static region has already been optimized in unmasked regions.

To effectively supervise the training of the dynamic branch, we propose to use optical flow [34] to estimate the motion mask of $\mathbf{M}_{\mathrm{sta}}$ and $\mathbf{M}_{\mathrm{dyn}}$. By focusing on these areas in a time variant way, our model is able to accurately capture and reconstruct the motion and changes in the scene. This step is crucial for refining the dynamic branch, ensuring that it becomes adept at processing motion-related data.

4.2. Evaluation

Baselines. Similar to the proposed method, SeaThru-NeRF [15] also focus on the reconstruction of underwater scenes, albeit not considering illumination changes and dynamic contents. Note that WaterNeRF [29] only works on raw underwater images and is thus excluded in the comparison. Meanwhile, some competent generic NeRF reconstruction methods are taken into account, namely Instant-NGP [25] and MIP-360 [5]. Besides, we also compare with DynamicNeRF [11], which is designed for scenes containing dynamic contents above the surface of the water.

For a fair comparison, a modified version of the proposed architecture is taken into consideration. We use "Proposed-**T**" to refer to the architecture modified to be time-invariant, to compare with SeaThru-NeRF [15] in the static setting. Specifically, we remove f_{dyn} entirely, and only condition the illumination field f_{I} on position and viewing direction. By removing the condition on time, and comparing it with methods designed for static scenes, the proposed formulation can be better validated.

Novel view synthesis. In this part, the proposed methods are tested on the 10% of images that are closest to the average position in the camera trajectory, instead of periodically selecting images for testing. We first compare the baselines and the "Proposed-T" method on the dataset of Seathru-NeRF [15]. As shown in Tab. 1, even a simplified version of the proposed method, "Proposed-T", achieves outstanding quantitative performance on all the scenes, under the metrics of PSNR, SSIM [32], and LPIPS [36]. This shows our method's ability to model the underwater structure, indicating that the proposed method effectively captures the underlying physical model under water. As shown in Fig. 4, the baseline methods are compared with "Proposed-T" on scenes in the SeaThru [15] dataset. Overall, "Proposed-T" method shows better fidelity, more intricate details, and less floater artifacts compared with other methods, which demonstrates the robustness of our approach across various underwater environments.

The proposed method is also tested on the proposed dataset, compared with Instant-NGP [25], SeaThru-NeRF [15], and DynamicNeRF [11]. As shown in Tab. 2, the proposed method achieves outstanding performance on these challenging scenes.

User study. We perform a user study to further validate our approach. Users are shown with sets of results rendered by the proposed method, SeaThru-NeRF [15], and Instant-NGP [25] in random order, each round 3 images from different perspectives, along with a ground truth reference view. They are asked to rate the three pairs of images in terms of realism, fidelity, and consistency from 0 to 5. With 122 users participating in the study and 2440 sets of comparisons collected, the preferences of the users are shown in Tab. 3. Clearly the proposed method is consistently favored by users, for not only generating realistic novel view synthesis results with high fidelity, but also free from floater artifact.

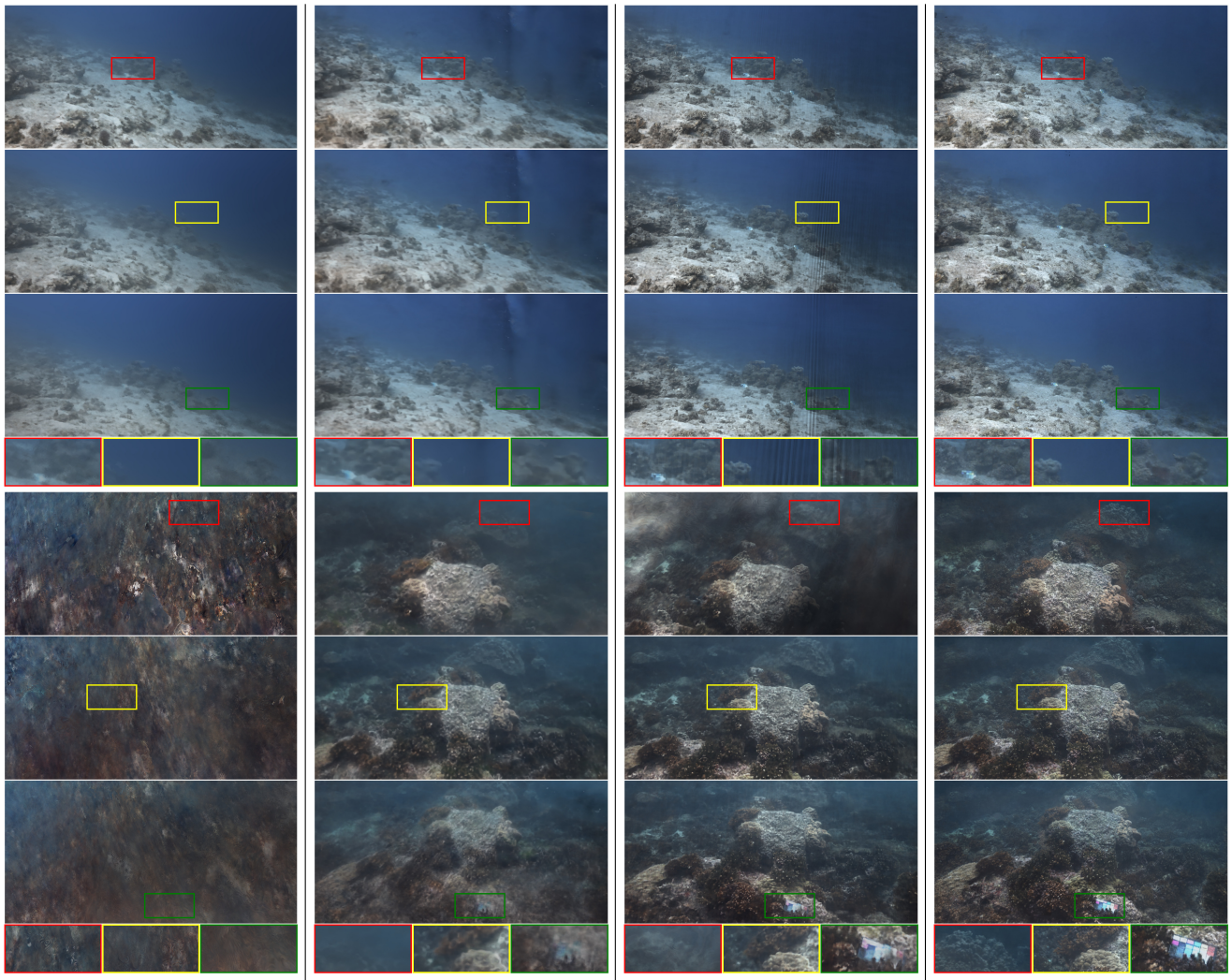
4.3. Ablation Study

A set of modified architectures are tested to justify the combined design. In addition to (1) removing all time-dependent network components as "Proposed-**T**" discussed before, we also consider the following variants: (2) "Proposed-**Ip**": removing the illumination field and tone mapper, using only the static branch and dynamic branch; (3) "Proposed-**I**": removing the illumination field, but keeping the tone mapper network along with the static and dynamic branch; (4) "Proposed-**SIp**": using only the dynamic branch to model the scenes. Since we remove the static branch in this setting, there is no constraint on the illumination field, and it is thus also removed.

From the results shown in Fig. 5(a), the removal of time-dependent network components in "Proposed-T" results in

Table 1. Quantitative evaluation results on Seathru dataset [15]. ↑ (↓) indicates larger (smaller) values are better. **Bold** font indicates the best results, while underlined number indicates the second best. The names of the scenes are listed in the first column.

PSNR(↑)/SSIM(↑)/LPIPS(↓)	MIP-360 [5]	Instant-NGP [25]	SeaThru-NeRF [15]	Proposed-T
CURAÇAO	28.23/.6834/.5713	27.66/.6840/.6057	29.27/.7413/.4430	**30.03/.8277/.2380**
REDSEA	19.55/.5097/.5198	20.85/.5187/.6229	22.48/**.6446**/.3903	**22.70**/.6240/**.3475**
PANAMA	18.32/.5559/.5951	21.85/.6039/.5949	23.70/.6644/.4034	**23.75/.6866/.2633**
JAPANESE	19.62/.6243/.4920	23.19/.7259/.4587	**25.93**/.8216/.2818	25.81/**.8533/.1825**



(a) MIP-360 [5] (b) Instant-NGP [25] (c) SeaThru-NeRF [15] (d) Proposed-T

Figure 4. Qualitative comparisons on the SeaThru dataset [15]. 3 novel view synthesis results are shown for the scene of REDSEA and PANAMA respectively, with each scene corresponding to a detailed local patch shown below the case. For more results, please refer to the supplementary material.

pure static representation of scenes, resulting in an averaged illumination and blurred effect of moving objects. Without illumination field,"Proposed-I" and "Proposed-Ip" lack the ability to model unstable illumination, leading to the distorted appearance in Figs. 5(b) and 5(c). Results rendered by "Proposed-SIp" show less high frequency details and more floater artifacts in Fig. 5(d), due to the lack of static components for constraint. Our complete model achieves the best visual results, showing both the moving tropical fish and the intricate textures of the corals in Fig. 5(e).

Note that we have considered removing the tone mapper only ("Proposed-**p**"), and try using the naive formulation in Sec. 3.4 to optimize the unstable illumination, but the network suffers from exploding gradient problem, and cannot be optimized properly. This phenomenon also justifies our approach to operate in the logarithmic space.

Table 2. Quantitative evaluation results on the proposed dataset.

PSNR(↑)/SSIM(↑)/LPIPS(↓)	Instant-NGP [25]	SeaThru-NeRF [15]	DynamicNeRF [11]	Proposed
CORAL	20.87/.4386/.7309	23.89/.6485/.4054	17.77/.5422/.8260	**26.17/.8282/.1573**
TURTLE	26.42/.8744/.2254	27.06/.8818/**.1917**	23.31/.8370/.4260	**28.10/.8997**/.2166
COMPOSITE	22.81/.5966/.5715	16.21/.4055/.8287	16.27/.7389/.4760	**25.09/.7990/.2386**
SARDINE	**21.73**/.6485/.4674	21.37/.5778/.6064	19.70/.6761/.6900	21.58/**.7226/.4541**

Table 3. The average user rating result of the proposed method, SeaThru-NeRF [15], and Instant-NGP [25]. The proposed model achieve the highest popularity in terms of realism, fidelity, and multi-view consistency (MVC).

Method	Realism	Fidelity	MVC
Instant-NGP [25]	3.13	3.64	3.92
SeaThru-NeRF [15]	3.56	2.29	1.71
Proposed	**4.46**	**4.77**	**4.29**

(a) Proposed-T (b) Proposed-Ip (c) Proposed-I

(d) Proposed-SIp (e) Proposed (f) Ground Truth

Figure 5. Comparison of removing different components during training in the proposed method, demonstrated in COMPOSITE.

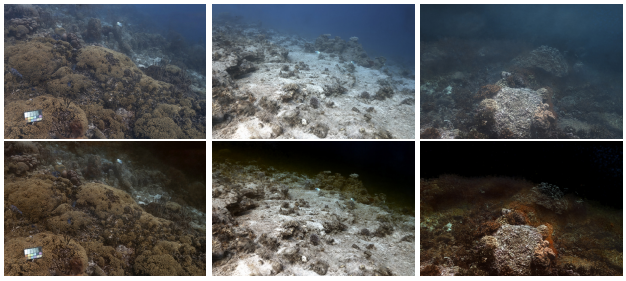


Figure 6. The first row demonstrates a novel view of the proposed method in CURAÇAO, REDSEA, and PANAMA. The second row displays the same view, but only with the water drained. For visualization, the shown images in CURAÇAO are multiplied by 2.

To verify the efficacy of the water medium parameters in the model, we also try removing it from the final rendering process, and obtain a realistic effect of drained underwater scene, as shown in the second row of Fig. 6. We also validate the pipeline by substituting the static branch with pre-trained NeRF model above the surface of the water, and create an immersive feeling of being in an underwater city, as shown in Fig. 7.

Figure 7. The effects of environment transferring on the LLFF dataset [22]. Displayed are two sets of comparisons, each pair featuring the original dataset (left) and its counterpart with simulated water effects (right).

5. Conclusions

This paper extends Neural Radiance Fields (NeRF) to represent underwater environments, addressing three dynamic factors which bring difficulty in underwater scene representation, namely distance-dependent visibility, unstable illumination, and dynamic objects. Our approach effectively simulates underwater scattering effects and optimizes water medium parameters alongside scene objects. The innovative illumination field and tone mapper module accurately capture dynamic lighting conditions underwater. Our two-stage reconstruction scheme robustly reconstructs static scenes, while also helping the accurate rendering of dynamic objects.

Our method outperforms existing models in both quantitative and qualitative evaluations of underwater scene modeling. It also enables editing possibilities for underwater scenes, such as removing water effects or transferring them to environments above the surface of the water, which can be used in film making, and virtual reality.

Limitations. Despite the promising performance of the proposed method, several limitations are still to be addressed in our future study. The fluidity of water is not taken into consideration. Water flows slowly enough to be neglected of its motion in the proposed dataset, while our model may not provide high-quality results when current becomes rapid. Moreover, the attenuation caused by water makes objects and visual features afar less recognizable, bringing instability to the camera poses estimated by structure-from-motion [28] methods.

Acknowledgement. This work is supported by National Natural Science Foundation of China under Grant No. 62136001, 62088102, 62276007. Renjie Wan is supported by the National Natural Science Foundation of China under Grant No. 62302415, Guangdong Basic and Applied Basic Research Foundation under Grant No. 2022A1515110692, 2024A1515012822, and the Blue Sky Research Fund of HKBU under Grant No. BSRF/21-22/16.

References

[1] Derya Akkaynak and Tali Treibitz. Sea-Thru: A method for removing water from underwater images. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 4

[2] Derya Akkaynak, Tali Treibitz, Tom Shlesinger, Yossi Loya, Raz Tamir, and David Iluz. What is the space of attenuation coefficients in underwater computer vision? In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 4

[3] Codruta O Ancuti, Cosmin Ancuti, Christophe De Vleeschouwer, and Philippe Bekaert. Color balance and fusion for underwater image enhancement. *IEEE Transactions on Image Processing*, 2017. 2

[4] Relja Arandjelović and Andrew Zisserman. NeRF in detail: Learning to sample for view synthesis. *arXiv preprint arXiv:2106.05264*, 2021. 4

[5] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3, 4, 6, 7

[6] Dana Berman, Tali Treibitz, and Shai Avidan. Non-local image dehazing. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[7] Dana Berman, Deborah Levy, Shai Avidan, and Tali Treibitz. Underwater single image color restoration using haze-lines and a new quantitative dataset. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1, 2, 4

[8] Subrahmanyan Chandrasekhar. *Radiative transfer*. 2013. 2, 4

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

[10] John Y Chiang and Ying-Ching Chen. Underwater image enhancement by wavelength compensation and dehazing. *IEEE Transactions on Image Processing*, 2011. 2

[11] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proc. of International Conference on Computer Vision*, 2021. 2, 3, 6, 8

[12] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. NeRFReN: Neural radiance fields with reflections. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3

[13] Junlin Han, Mehrdad Shoeiby, Tim Malthus, Elizabeth Botha, Janet Anstee, Saeed Anwar, Ran Wei, Mohammad Ali Armin, Hongdong Li, and Lars Petersson. Underwater image restoration via contrastive learning and a real-world dataset. *Remote Sensing*, 2022. 2

[14] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010. 1, 2

[15] Deborah Levy, Amit Peleg, Naama Pearl, Dan Rosenbaum, Derya Akkaynak, Simon Korman, and Tali Treibitz. SeaThru-NeRF: Neural radiance fields in scattering media. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 1, 2, 3, 5, 6, 7, 8

[16] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2

[17] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. DynIBaR: Neural dynamic image-based rendering. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[18] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *Advances in Neural Information Processing Systems*, 2020. 2, 4

[19] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[20] Huimin Lu, Yujie Li, Lifeng Zhang, and Seiichi Serikawa. Contrast enhancement for images in turbid water. *Journal of the Optical Society of America. A, Optics, image science, and vision*, 2015. 2

[21] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the wild: Neural radiance fields for unconstrained photo collections. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2, 3

[22] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics*, 2019. 8

[23] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proc. of European Conference on Computer Vision*, 2020. 1, 2, 3, 4

[24] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. NeRF in the dark: High dynamic range view synthesis from noisy raw images. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 5

[25] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 2022. 1, 4, 6, 7, 8

[26] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proc. of International Conference on Computer Vision*, 2021. 2

[27] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. HyperNeRF: a higher-dimensional representation for topologically varying neural radiance fields. *ACM Transactions on Graphics*, 2021. 2

[28] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 8

[29] Advaith Venkatramanan Sethuraman, Manikandasriram Srinivasan Ramanagopal, and Katherine A Skinner. WaterNeRF: Neural radiance fields for underwater scenes. *arXiv preprint arXiv:2209.13091*, 2022. 3, 6

[30] Fengrui Tian, Shaoyi Du, and Yueqi Duan. MonoNeRF: Learning a generalizable dynamic radiance field from monocular videos. In *Proc. of International Conference on Computer Vision*, 2023. 2

[31] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 3, 4

[32] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004. 6

[33] Tianhao Wu, Fangcheng Zhong, Andrea Tagliasacchi, Forrester Cole, and Cengiz Oztireli. D^2NeRF: Self-supervised decoupling of dynamic and static objects from a monocular video. In *Advances in Neural Information Processing Systems*, 2022. 2

[34] Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *Proc. of International Conference on Computer Vision*, 2021. 3, 6

[35] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proc. of International Conference on Computer Vision*, 2021. 2, 3

[36] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 6

[37] Shaomin Zhang and S. Negahdaripour. 3-D shape recovery of planar and curved surfaces from shading cues in underwater images. *IEEE Journal of Oceanic Engineering*, 2002. 5

[38] Chengxuan Zhu, Renjie Wan, and Boxin Shi. Neural transmitted radiance fields. In *Advances in Neural Information Processing Systems*, 2022. 3

[39] Chengxuan Zhu, Renjie Wan, Yunkai Tang, and Boxin Shi. Occlusion-free scene recovery via neural radiance fields. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 3