TOWARDS CONTROLLABLE AND PHYSICAL INTERPRETABLE UNDERWATER SCENE SIMULATION

Kaixin Chen¹, Lin Zhang¹*, Ying Shen¹, Yicong Zhou²

¹School of Software Engineering, Tongji University, Shanghai, China
²Department of Computer and Information Science, University of Macau, Macau, China

ABSTRACT

The realistic simulation of underwater scenes has important significance for many researches related to underwater vision, such as underwater image restoration, underwater moving object monitoring, etc. To date, however, the existing underwater scene simulation pipelines are either too complicated due to the continuous spectra and camera parameters involved, or difficult to control since the empirically controlled distance based fog effect is usually used by them. In this paper, we try to fill in this research gap by proposing an Underwater Scene Simulation approach, namely USSim, which especially focuses on the influence of ocean water. In USSim, Jerlov water type and depth are regarded as main variables to control the simulation effects. In addition, the spectra of the incident light is decomposed into three primary components and their attenuations are modeled separately, and finally the simulated scene is generated via the hybrid underwater imaging model proposed by us. USSim greatly reduces the computational complexity and enables the fog effect to be controlled by variables with explicit physical meanings. The controllability, physical interpretability and simulation effects of our USSim under different conditions have been verified by extensive experiments. To make our results reproducible, the source code is made online available at https://cslinzhang.github.io/USSim/.

Index Terms— Underwater Scene Simulation, Jerlov Water Types, Underwater Imaging Model

1. INTRODUCTION

Recent years have witnessed a growing interest in the exploration and researches on marine mineral resources [1] and biological populations [2] based on vision technologies. However, it is highly expensive and even impractical to get real underwater images in some cases. A feasible solution to conquer this problem is to use virtual simulation technologies to create realistic underwater scenes, which is also the focus of this paper.

From the theoretical perspective, the primary issue of simulating underwater images is how to model the imaging mechanism of the underwater object. The underwater optical imaging models related to the imaging mechanism can be roughly sorted into four categories [3], the point spread function model (PSFM) [4], the turbulence degradation model (TDM) [5], the Jaffe-McGlamery model (JMM) [6–8], and the foggy image degradation model (FIDM) [9, 10]. Among them, PSFM and TDM lack physical interpretability correlated with the optical properties of seawater, and thus is not suitable for the virtual simulation. JMM attempts to decompose the light received by the underwater camera, and is the most widely used underwater imaging model so far. However, because of different emphases in various usage scenarios [11–13], the formulas of JMM are not uniform when used. Therefore, it is necessary to simplify JMM when it is applied in underwater scene simulations. FIDM regards the underwater image as a linear superposition of the anhydrous scene and the water body. It is mainly used for dark channel prior based image dehazing. When FIDM works in an atmospheric environment, it can accurately estimate the optical distance from the camera to the object, but when applied to an underwater environment, it cannot accomplish this task well [14–16].

From the application aspect, the existing underwater scene simulation methods are mostly based on the FIDM and use distance fog effect in game engines [17, 18]. For example, in [17], Liarokapis et al. specify the fog effect’s color and density empirically and calculate the loss and intensity of extra illumination by linear interpolation. Thompson et al. [18] directly extract the color of the fog effect from the reference photos. Although these application oriented methods are able to simulate underwater scenes, they share some common shortcomings, such as low controllability and lack of physical interpretability.

To solve the aforementioned problems in the application-oriented method, in this article, we propose a hybrid underwater imaging model and build a Underwater Scene Simulation pipeline in Unity based on this model, named USSim. Our contribution can be summarized as follows:

1. A simulation-specific underwater optical imaging model is proposed to build the distance fog in the game engines rather than setting the distance fog empirically like other works, which ensures our simulation methods with strong physical interpretability.

2. We propose the simulation pipeline USSim, which to our knowledge, is the first underwater environment simulation pipeline in the game engines that applies the classification criteria of the Jerlov water types. As a result, our USSim can simulate underwater scenes independently without reference images.

3. We also propose a strategy to calculate the color of underwater ambient light. By decomposing the incident light into three primary rays using color matching function, we avoid the calculation of the entire continuous spectrum and thus reduce the computational complexity.

2. METHODOLOGY

In this section, the hybrid underwater imaging model which is the basis of USSim will be given first in Sec. 2.1. Then, in Sec. 2.2 the workflow of USSim will be presented briefly. Finally, in Sec. 2.3 we will introduce the details of method designed to calculate the color of underwater ambient light.
2.1. A Hybrid Underwater Imaging Model

The proposed hybrid underwater imaging model in this article is based on JMM [6–8] and FIDM [9, 10]. So, in this subsection, we first briefly introduce JMM and FIDM, and subsequently present our model.

Jaffe-Mcglamory Model. JMM divides the light received by the underwater camera in the line of sight $E_T$ into three components: 1) the light directly reflected from the captured scene $E_d$, 2) the light that reaches the camera after being scattered by small particles $E_f$, and 3) the light from the atmosphere and reflected by suspended particles $E_b$. The relationship among these components and $E_T$ conforms to,

$$E_T = E_d + E_f + E_b. \quad (1)$$

As when the objects are relatively close to the camera, $E_f$ has less influence on underwater imaging than $E_d$ and $E_b$, so like other studies [19], we ignore $E_f$ and only consider the rest.

When the camera and the object are in the same horizontal plane, the received radiance of light $E_T$ can be obtained by,

$$E_T = E(d, \lambda)e^{-\beta(\lambda)z} + B^{\infty}(\lambda)(1 - e^{-\beta(\lambda)z}), \quad (2)$$

where $E(d, \lambda)e^{-\beta(\lambda)z}$ represents the direct transmission light $E_d$ using the Beer-Lambert law [20] and $B^{\infty}(\lambda)(1 - e^{-\beta(\lambda)z})$ denotes the backward scattering light $E_b$ [21]; $\beta(\lambda)$, $\lambda$, and $z$ denote the distance of the optical path, respectively.

Foggy image degradation model. FIDM [9, 10] is composed of two parts, the attenuation component and the ambient light component. It can be expressed as,

$$I(x) = J(x) \cdot t(x) + A \cdot (1 - t(x)), \quad (3)$$

where $x$ means the coordinate of a pixel in the image, $A$ is a constant value of the ambient light from horizon sky, and $J(\cdot)$, $t(\cdot)$, and $I(\cdot)$ return the foggy image, the fog-free image, and the transmittance, respectively.

Although according to [9], $t(x)$ is generally taken as $e^{-\beta z(x)}$ in the atmosphere, in different usage scenarios such as dark channel prior dehazing, $t(x)$ can also be calculated with other forms [14].

A hybrid underwater imaging model. From Eq. 2 and Eq. 3, we can have some useful findings that when combining the physical meaning of JMM with the form of FIDM and making $t = e^{-\beta z(x)}$, we are able to construct a hybrid underwater imaging model which is formed as,

$$I(\lambda) = J(\lambda) \cdot e^{-\beta(\lambda)z} + A(\lambda) \cdot (1 - e^{-\beta(\lambda)z}), \quad (4)$$

where $J(\lambda)$ is the observed radiance representing the simulated underwater scene, $A(\lambda)$ is the anhydrous scene radiance, $A(\lambda)$ is the radiance of ambient light from backward scattering, $\beta(\lambda)$, $\lambda$, and $z$ are the same as defined in Eq. 2.

However, Eq. 4 is not suitable when simulating underwater scenes by distance fog in the game engines, since calculations can only be performed in the sRGB color space instead of the entire spectra. When we use Eq. 4 to simulate underwater scenes in RGB channels separately, if $\beta(\lambda)$ and $J(\lambda)$ are small, the simulation scenes will be reddish, which is unreasonable. Therefore, in practical applications, we simplify the Eq. 4 to the form as,

$$I(x) = J(x) \cdot e^{-\beta z(x)} + A \cdot (1 - e^{-\beta z(x)}), \quad (5)$$

where $I(x)$ is the RGB color at a certain coordinate in the imaging plane of the simulated underwater scene, $J(x)$ is the RGB color of the same coordinate in the anhydrous scene, $A$ is the color of the ambient light, and $\beta$ is a constant value of the attenuation coefficient. The methods of obtaining these variables will be introduced in the following content.

2.2. The Underwater Scene Simulation Workflow

Our USSim consists of three stages: anhydrous scene construction, ambient light’s color calculation, and distance fog generation.

In the anhydrous scene construction stage, we build an anhydrous scene in the game engine and select an observation camera. Once the scene and the camera are determined, we can easily obtain the color of the anhydrous scene $J(x)$ and the distance from the target to the camera $z(x)$ at any pixel $x$ in the imaging plane. In the ambient light’s color calculation stage, we use the water type, the water depth, and the color of the incident light above the water surface as variables to calculate the color of the ambient light $A$, and use it as the color of the distance fog.

In the final stage of distance fog generation, we refer to Eq. 5, use $e^{-\beta z(x)}$ to present the density of the fog effect, and add it to the anhydrous scene to generate the underwater scene. To integrate the different degrees of attenuation of the three wavelengths of RGB by water, we use the following equation to calculate $\beta$ as,

$$\beta = \frac{R}{R + G + B}\beta_R + \frac{G}{R + G + B}\beta_G + \frac{B}{R + G + B}\beta_B, \quad (6)$$

where $(R, G, B)$ are the components of the ambient light’s color $A$ in the RGB channels, and $\beta_R$, $\beta_G$, and $\beta_B$ are the attenuation coefficients in the case of the wavelengths of red, green, and blue light, which can be obtained from [22].

2.3. Derivation of the Underwater Ambient Light

From Sec. 2.2, $A$ is the only undetermined variable in Eq. 5. To this end, we design a pipeline as shown in Fig. 1 to determine $A$.

The three parts of the pipeline, the decomposition of light, the calculation of attenuation, and the conversion of the color space, will be introduced in this part.

2.3.1. Color Decomposition and Attenuation Calculation

Different wavelengths of light are attenuated differently, but simulating the attenuation of the entire continuous spectra is too com-
Table 1. Comparison of the controllable physical quantities.

<table>
<thead>
<tr>
<th></th>
<th>λ</th>
<th>z</th>
<th>d</th>
<th>$E / L$</th>
<th>$K_d$</th>
<th>$\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours (USSim)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Liarokapis et al. [17]</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Thompson et al. [18]</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
</tbody>
</table>

Complicated and the data acquisition requirements are too high, so we consider disassembling the sunlight from a discrete perspective.

According to the color matching function [23], light of any color can be matched by the three primary colors, and the ratio of the three primary colors can be expressed as tristimulus values, which are the required luminosities of the three primary colors to achieve a color match. So we use the color matching function to convert the D65 standard daylight simulation as the incident light into tristimulus values ($R_I$, $G_I$, $B_I$) representing luminosities.

However, the luminosity is only a subjective concept, to get an absolute luminance, we must un-scale the values using a set of relative luminances via,

$$
\begin{align*}
    r &= I_{R} \cdot R_l \\
    g &= I_{G} \cdot R_g \\
    b &= I_{B} \cdot R_b
\end{align*}
$$

where $r$, $g$, and $b$ are the luminances, $R_l = 1$, $R_g = 4.5907$ and $R_b = 0.0601$ mean the relative luminances.

The above-mentioned luminances are the intensity of light per unit area of their source. Since luminance is independent of energy, it cannot be directly used to calculate attenuation. So we need to convert ($r$, $g$, $b$) to radiances ($L_r$, $L_g$, $L_b$), which represent the radiant energy emitted per unit time in a specified direction by a unit area of an emitting surface, with a simple linear relationship [24]. Then, according to the definition of downward diffusion attenuation coefficient, the attenuation can be calculated via,

$$
\begin{align*}
    L_{2r} &= L_r \cdot e^{-K_{dr}d} \\
    L_{2g} &= L_g \cdot e^{-K_{dg}d} \\
    L_{2b} &= L_b \cdot e^{-K_{db}d}
\end{align*}
$$

where $d$ is the same as defined in Eq. 2; $K_{dr}$, $K_{dg}$, and $K_{db}$ are the coefficients of downward diffusion attenuation in the case of the wavelengths of red, green, and blue light. Furthermore, we convert ($L_{2r}$, $L_{2g}$, $L_{2b}$) back to luminances ($r_z$, $g_z$, $b_z$) with a simple linear relationship [24].

As a result, by adopting the pipeline above, we can obtain the attenuation of the red, green, and blue components of the discrete sunlight, and get their luminances ($r_z$, $g_z$, $b_z$) after the attenuation.

### 2.3.2 Color Space Transformation

To obtain a specific color $A$, we still need to transform the luminances of red, green, and blue in the color space CIE RGB to the color space sRGB.

The color space with RGB tristimulus values as coordinates is called CIE RGB. Because there are negative numbers in the color matching function [23], to facilitate the calculation, CIE proposed a three-dimensional conversion matrix $M$ to adjust all the tristimulus values to positive numbers and thus construct CIE XYZ with all coordinates in the range $[0,1]$.

So we use the conversion matrix $M$ to convert our $(r_z$, $g_z$, $b_z)$ from Eq. 8 to the coordinate in CIE XYZ via,

$$
[X \ Y \ Z]^T = M[r_z, g_z, b_z]^T.
$$

As CIE XYZ is independent of the camera equipment, it needs to be further converted to sRGB for rendering. To fulfill this goal, we need to transform CIE XYZ to sRGB in two steps [25]. First, the unprocessed sRGB coordinates can be obtained by multiplying the CIE XYZ coordinates via,

$$
[r_{raw}, g_{raw}, b_{raw}]^T = N[X \ Y \ Z]^T,
$$

where $[r_{raw}, g_{raw}, b_{raw}]^T$ is the unprocessed coordinate of a point in sRGB and $N$ is another conversion matrix. Second, the sRGB coordinates are obtained by a piecewise function,

$$
A_c = \begin{cases} 
12.92c_{raw} & \text{if } c_{raw} \leq 0.0031308 \\
3.9298c_{raw} - 0.1437 & \text{if } c_{raw} > 0.0031308
\end{cases}
$$

where $c_{raw}$ is $r_{raw}$, $g_{raw}$ or $b_{raw}$, $A_c$ is $A_r$, $A_g$, or $A_b$ and $A_c \in [0,1]$, and $(A_r, A_g, A_b)$ are the values of RGB channels of $A$.

Using the above-presented pipeline, the color of underwater ambient light $A$ can be determined according to the water types, the water depth, and the incident light. Consequently, Eq. 5 can be used to simulate the underwater color.

### 3. EXPERIMENTS AND DISCUSSIONS

In this section, we will evaluate the performance of our proposed USSim from both qualitative and quantitative experiments.

#### 3.1 Qualitative Experiments

We performed our qualitative experiments from two aspects: controllability and simulation outputs.
Fig. 3. Comparison of the underwater scene simulations. The images presented in (a)∼(c) are the simulation results of Liarokapis et al.’s [17], Thompson et al.’s [18] and our approaches, respectively. (d) Actual underwater images under similar conditions.

Table 2. Comparison of the average of image AuthESI values under oceanic and coastal conditions.

<table>
<thead>
<tr>
<th></th>
<th>oceanic AuthESI</th>
<th>coastal AuthESI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real images</td>
<td>3.253</td>
<td>2.075</td>
</tr>
<tr>
<td>Ours (USSim)</td>
<td>3.597</td>
<td>2.171</td>
</tr>
<tr>
<td>Liarokapis et al.</td>
<td>4.068</td>
<td>3.978</td>
</tr>
<tr>
<td>Thompson et al.</td>
<td>4.532</td>
<td>5.350</td>
</tr>
</tbody>
</table>

Analysis on controllability. To evaluate the controllability and the ability of physical interpretation of the proposed USSim, according to [22] and [26], some commonly used physical quantities related to underwater imaging are analysed in this subsection. Among them, $d$, $\beta$, $\lambda$, and $z$ are the same as defined in Eq. 2, $K_d$ is the same as defined in Eq. 8, and $E (L)$ means the irradiance (radiance), which represents the energy of the light. Two representative underwater simulation pipelines, Liarokapis et al.’s [17] and Thompson et al.’s [18], widely used in game engines, are compared with our method in terms of the above-mentioned physical quantities. The variable usage of each scheme is listed in Table 1.

As shown, our USSim involves more physical quantities. Among the three approaches in Table 1, both [17] and [18] use the distance based fog effect to simulate the underwater environment, but many of the quantities related to the fog effect is set empirically. Thus, their simulations lack diversity. Instead, our simulation outputs can be customized by changing the water type to adjust $K_d$ and $\beta$, modulating $d$ to control the attenuation, and regulating $z$ to affect the fog effect’s intensity, which makes our USSim more theoretical and practical, and also guarantees the controllability.

Simulation performance on underwater scene. Assuming that the incident light is D65 which is one of the standard daylight simulations, our underwater scene simulation results are illustrated in Fig. 2. Moreover, USSim was compared with two competitors [17, 18], with some simulation results and the samples listed in Fig. 3. All simulation results are generated in Unity 2020.3.18f1 LTS.

In Fig. 2 (a), we show the simulation results of different Jerlov water types. Jerlov categorized the ocean water into five oceanic types (I, IA, IB, II, and III) and five coastal types (1C, 3C, 5C, 7C, and 9C) in [27]. And in Fig. 2 (a), from left to right, the whole set of pictures of Fig. 2 (a) is the oceanic and the coastal water in the same sequences. As the water become more and more turbid, the simulated scene gradually changed from blue to yellow-green, and even to reddish brown. Fig. 2 (b) demonstrates some examples of the simulation outputs of the Jerlov II water type with depths ranging from 0 to 14m. As shown, with the water depth increasing, the brightness of the color changes greatly, and the hue changes slightly.

In addition, the simulation results of the methods in [17] and [18] as well as USSim under several underwater conditions are illustrated in Fig. 3, where each row of the images represents the simulation scenes and the sample photos under similar conditions. Since the source codes of [17] and [18] were unavailable, we reproduced these two methods according to their corresponding papers. Besides, we set the colors of their fog effects directly from the sample photos, and ignored some simulation skills such as the caustics, god rays, etc., which had little to do with the underwater imaging models. As shown in Fig. 3, it can be found that our USSim results appear smoother and more natural.

3.2. Quantitative Experiments

The real underwater environment is affected by many factors and has various characteristics in different regions, so there is still no perfect standard to quantify the quality of the simulations results. But considering that the distance fog is commonly used in game engines to simulate underwater scenes, we adopted the synthetic fog/hazy image realism evaluator (AuthESI for short) proposed in [28] to objectively evaluate the authenticity of the synthetic underwater images. A smaller AuthESI value indicates a more natural simulated image.

With the above knowledge, we designed a quantitative experiment. We divided underwater scenes into two types: oceanic environment and coastal environment. For each environment, we used our USSim to simulate 40 underwater images. The same operation was also applied to Liarokapis et al.’s method [17] and Thompson et al.’s method [18]. So there were 240 images in total. In addition, we collected 40 real underwater images for each environment as control groups. Then we calculated the images’ AuthESI values and got the averages of the 8 groups. The results are summarized in Table 2, from which it can be seen that our USSim can generate better simulation results in both oceanic and coastal environment.

4. CONCLUSION

In this article, to simulate the underwater scene, we take the water type and the water depth as the main variables, decompose the spectra of the incident light and design a pipeline to determine the underwater ambient light, and propose a hybrid underwater imaging model to fulfill this goal. Meanwhile, extensive experiments verify the controllability of the proposed method. In addition, some underwater color simulation results and the quantitative comparison also demonstrate that our approach can perform well in underwater environment. In future research, we will devote our efforts to explore more factors that affect the color of underwater imaging.

5. ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grants 61973235, 61936014, and 61972285, in part by the Natural Science Foundation of Shanghai under Grant 19ZR1461300, in part by the Shanghai Science and Technology Innovation Plan under Grant 20510760400, in part by the Dawn Program of Shanghai Municipal Education Commission under Grant 21SG23, in part by the Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0100, and in part by the Fundamental Research Funds for the Central Universities.
6. REFERENCES


