Vision, Modeling, and Visualization (2011)
Peter Eisert, Konrad Polthier, and Joachim Hornegger (Eds.)

Simulating Deep Sea Underwater Images Using Physical Models for Light Attenuation, Scattering, and Refraction

A. Sedlazeck[1] and R. Koch[1]

[1] Institue of Computer Science, Christian-Albrechts-University of Kiel, Germany

Abstract

When adapting computer vision algorithms to underwater imaging, two major differences in image formation occur. While still traveling through the water, light rays are scattered and absorbed depending on their wavelength, creating the typical blue hue and low contrast in underwater images. When entering the underwater housing of the camera, light rays are refracted twice upon passing from water into glass and into air. We propose a simulator for both effects based on physical models for deep sea underwater images captured by cameras in underwater housings with glass port thicknesses in the order of centimeters. Hence, modeling refraction by explicitly computing the correct path of the rays allows to accurately simulate distortions induced by underwater housings. The Jaffe-McGlamery model for effects on color is often used in computer vision algorithms as a base for simplification. We extend this model to incorporate color images, shadows, and several light sources.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual reality I.6.8 [Simulation and Modeling]: Types of Simulation—Visual

1. Introduction

When working with underwater images in the area of computer vision, two fundamental differences between image formation in air and image formation in water need to be considered. First, while still traveling through the water body, light rays are attenuated and scattered [Mob94], the former leading to a loss of photons, while the latter leads to a gain of photons. Both effects are dependent on the wavelength and thus effect the colors, being responsible for the bluish/greenish colors typical for underwater images. Secondly, light rays are refracted at the water-air interface [Hec05] of the camera housing, causing geometrical distortions in the image.

Both effects cause classic computer vision algorithms to have difficulties when being applied to underwater images, and therefore require those algorithms to be adapted to the special scenario. For example the computation of a panorama or of 3D structure from a moving camera (Structure from Motion, SfM [HZ04]) requires to match correspondences between images and the use the implicitly contained geometric information for computing the camera poses. Deteriorated colors effect the feature matching while refraction causes a violation of the assumed pinhole camera model (fig. 1), thus applying established algorithms to underwater images is at best difficult. When researching suitable adaptations, simulated underwater images with known, exact ground truth are a huge advantage, when it comes to testing the algorithm adaptations. We therefore present a renderer for underwater images, simulating both, color and geometrical effects using physical models in order to gain accurately simulated underwater images, with known parameters for the physical models.

The change in colors is simulated by the Jaffe-McGlamery model [Jaf90], [McG75], often used as a basis for computer vision algorithms for color restoration, e.g. [TOA06], [SK04], [TS08]. The model used in [NCB09] can be derived from the Jaffe-McGlamery model by integrating the backscatter component, as was shown in [SK04].

For our simulator, the Jaffe-McGlamery model is extended in order to incorporate several light sources, render shadows, and uses a parametrized volume scattering function. The change in the geometry induced by the glass inter-

DOI: 10.2312/PE/VMV/VMV11/049-056

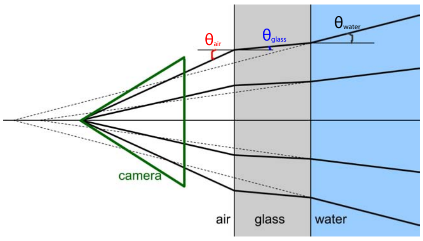

Figure 1: *Flat port underwater housing. The solid lines show how each ray is refracted at the water-glass and the glass-air interface. When tracing the underwater rays without refraction towards the camera center (dashed lines), it becomes apparent that the pinhole camera model with its single view point is invalid because the rays do not meet in one common center of projection.*

face of the underwater housing is simulated by parametrizing the housing and explicitly modeling refraction. We concentrate on deep sea scenarios, where no natural sun light exists, but lamps immersed in the medium illuminate only a few meters before the light is attenuated completely. Thickness of camera housing ports reaches the order of centimeters, causing strong distortions due to refraction. However, these scenarios are of great importance to ROVs (Remotely Operated Vehicles), which can operate at great water depths and are able to deliver high resolution images and video sequences.

The following paper is organized as follows. First, a section on related work introduces existing systems for rendering underwater images. Then the refractive camera model will be explained, followed by the radiometric model for changes in colors. After that, results will be presented, followed by conclusion and outlook.

2. Related Work

In the literature, a multitude of methods for rendering water or participating media in general exist. Consequently, there are different approaches to rendering effects of water on light. Photon maps are used by several authors, mainly when rendering scenarios including caustics, refer for example to the work of Jensen et al. [JC98]. [GSMA08] use a bio-optical model including absorption, elastic and inelastic scattering and then also use photon mapping in order to render their images. Others, like [Mob94] solve the radiative transfer equation analytically.

A lot of papers are concerned with rendering the ocean surface including waves. In order to compute the correct water color however, models for underwater light propagation need to be used as well, even if they can be grossly simplified for this purpose. Refer to [DCGG11] for a recent overview, or to [PA01] for a model for underwater light propagation, which is a simplification of the Jaffe-McGlamery model.

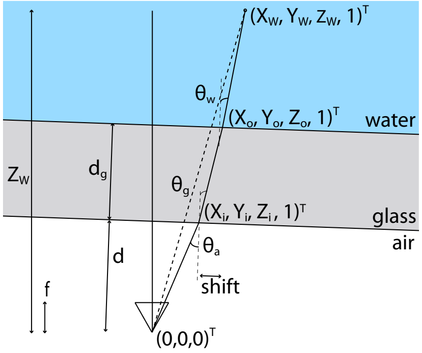

Figure 2: *When unprojecting a point, the ray runs from the camera through air intersecting the inner interface plane* $(X_i, Y_i, Z_i)^T$. *The refracted ray in glass intersects the outer interface plane* $(X_o, Y_o, Z_o)^T$, *is refracted again, yielding the ray in water on which the 3D point* $(X_w, Y_w, Z_w)^T$ *lies.*

Numerous recent papers target rendering underwater images close to the surface at water depths which the sun light can reach. In those cases, effects like shafts of light and caustics are of interest for rendering. Exemplary papers are [IDN02] and [CP09]

In our case, we need a simulator using a physical model. However, instead of finding a solution to the radiative transfer equation, we decided to use the Jaffe-McGlamery model, due to it being the basis of a lot of existing computer vision algorithms. Comparable approaches, where the volume is divided into voxels and/or point spread functions are used for blurring, can also be found e.g. in [PARN04] or [CSCS02]. Another advantage of the Jaffe-McGlamery model is that it can easily be combined with the refractive camera model, which will be described in the next section.

3. Refractive Camera Model

The derivation of the ray cast in the physical underwater model (fig. 2), presented here, is essentially the same as derived by Kunz and Singh in [KS08] with slight differences in the port parametrization. When using a flat port in front of an underwater housing, the distance to the port, the glass thickness, and the normal of the glass surface within the camera coordinate system are important parameters. Here, the inner interface plane is parametrized by $\Pi_i = (\underbrace{n_1, n_2, n_3}_{\mathbf{n}_\Pi}, -d)$ containing the normal vector and the port's distance to the origin. In addition, the outer interface plane is parametrized by the same normal vector and glass thickness d_g: $\Pi_o = (n_1, n_2, n_3, -(d+d_g))$ (refer to figure 2). When unprojecting (a camera *projects* 3D points to 2D points in the image plane, while *unproject* denotes the computation of 3D points or rays from 2D points) an image point, the goal is the computation of the point on the outer interface plane and the direction of the ray in water. First, an image point is unprojected to the unit length ray within the camera's underwater housing $\tilde{X}_a$, using the camera's intrinsic parameters including radial distortion. This ray is in the camera coordinate system, meaning that the center of projection is in the origin. In order to find the intersection X_i between ray and interface the following equation is used:

$$\Pi^T \begin{pmatrix} \lambda_g \tilde{X}_a \\ \lambda_g \tilde{Y}_a \\ \lambda_g \tilde{Z}_a \\ 1 \end{pmatrix} = 0 \quad \Rightarrow \quad \lambda_g = \frac{d}{<\mathbf{n}_\pi, \tilde{X}_a>} \qquad (1)$$

$$\Rightarrow X_i = \lambda_g \tilde{X}_a.$$

The intersection of the port's inner plane and the ray, parametrized by λ_g, is used to determine the point on the inner plane of the interface X_i. In order to compute the ray within the glass, the incidence angles θ_a and θ_g are computed. The angle θ_a between normal and incident ray before refraction is derived by the scalar product between the plane's normal $\mathbf{n}_\Pi$ and the ray in air, then Snell's law [Hec05] is applied to compute θ_g:

$$\theta_a = acos\left(\frac{<\mathbf{n}_\Pi, \tilde{X}_a>}{\|\mathbf{n}_\Pi\|\|\tilde{X}_a\|}\right) \quad \Rightarrow \theta_g = asin(sin\theta_a \frac{n_a}{n_g}). \quad (2)$$

Now, the ray being incident upon the inner interface plane needs to be rotated/refracted. This rotation is described by a rotation around the normal resulting from the cross product of the plane normal and the incoming ray. With $\theta_{rot} = \theta_g - \theta_a$, the unit quaternion for the rotation can be defined:

$$\mathbf{n}_{rot} = \frac{\mathbf{n}_\Pi \times \tilde{X}_a}{\|\mathbf{n}_\Pi\|\|\tilde{X}_a\|sin\theta_a} \quad \Rightarrow \quad \mathbf{q} = \begin{pmatrix} \frac{sin\left(\frac{\theta_{rot}}{2}\right)}{\|\mathbf{n}_{rot}\|}\mathbf{n}_{rot} \\ cos\left(\frac{\theta_{rot}}{2}\right) \end{pmatrix}. \qquad (3)$$

This quaternion is applied to the ray $\tilde{X}_a$, yielding the refracted ray $\tilde{X}_g$, which describes the light's traveling direction within the glass. Now, the point on the outer interface is computed:

$$X_o = X_i + \lambda_w \tilde{X}_g \quad \text{with} \quad \lambda_w = \frac{(d_g + d - <\mathbf{n}_\pi, X_i>)}{<\mathbf{n}_\pi, \tilde{X}_g>}. \quad (4)$$

The ray within the glass is refracted again, using the indices of refraction for glass and water, the cross product, and the unit quaternion rotation. The result is the ray in water $\tilde{X}_w$.

The 3D point can be computed if the distance *dist* between the camera center and the 3D point is known. The following equation can be solved for α_w yielding the distance the ray needs to travel from the interface point:

$$\|X_o + \alpha_w \tilde{X}_w\| = dist \quad \Rightarrow \quad X_w = X_o + \alpha_w \tilde{X}_w. \quad (5)$$

X_w is still in the camera coordinate system, but using the transform of the camera with rotation and translation, the point can easily be transformed into the world coordinate system. Figure 3 gives an overview of the whole system. This section described how the ray in water is computed, given a completely defined camera (top two boxes in figure 3). After intersecting the water ray with the 3D structure, the radiometric model needs to be applied, which is described in the next section.

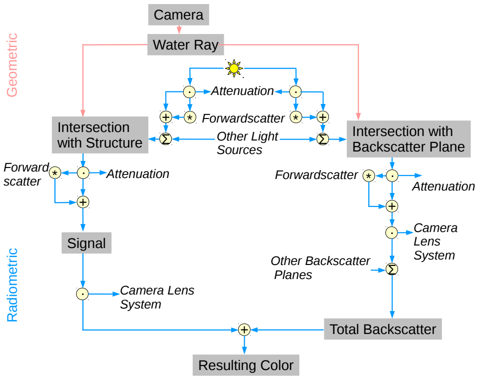

Figure 3: *Overview of system with both models. In red is the geometric part with computation of water ray given the camera and computation of intersections with 3D structure and backscatter planes. Blue: the radiometric part with derivation of signal and backscatter. Note that computation for one exemplary backscatter plane and one exemplary light source are drawn and that other light sources and backscatter planes are added to the model by the $\sum$-operators. Attenuation is denoted by a $\cdot$-operator, and as in equations 13, (15), (17), and (18) it leads to a loss of flux due to multiplication with a number* < 1. *Forward scatter means convolution of a Gaussian filter with the current image and adding the result to the current image (eq. (14), (16), (17)).*

4. Color Model

In this section the model for the change in color due to absorption and scattering will be described. The basic model is the Jaffe-McGlamery model ([Jaf90], [McG75]), upon which many computer vision algorithms are based.

4.1. Absorption

While traveling through a volume of water, photons can be absorbed by differently sized particles, like water molecules, dissolved salt, yellow matter (dissolved organic particles), phytoplankton, or anorganic particles caused by erosion and similar processes. These groups of matter cause photons to be absorbed to a different degree depending on the photon's wavelength and the particle concentration. For each group of matter, absorption is parametrized by the absorption coefficient, describing how much light per traveling distance is absorbed. However, instead of modeling all effects separately, the sum of all absorption coefficients as the overall absorption coefficient a in $[m^{-1}]$ is used. This coefficient is depending on the wavelength λ:

$$E(z,\lambda) = E(0,\lambda)e^{-za(\lambda)} \quad [Wm^{-2}] \tag{6}$$

with $E(0,\lambda)$ being the irradiance before traveling any distance through the water, z being the distance traveled, and a being the spectral absorption coefficient depending on the wavelength λ.

4.2. Scattering

Apart from being absorbed, when colliding with molecules or other matter, photons can also be scattered, meaning the photon's direction is changed after the collision. The volume scattering function (VSF, fig. 4) describes how much light is scattered towards a certain angle $\psi \in [0,\pi]$ by $\beta(\psi,\lambda)$ in $[sr^{-1}m^{-1}]$. Integrating $\beta(\psi,\lambda)$ in all directions yields the scattering coefficient, the equivalent to the absorption coefficient:

$$b(\lambda) = 2\pi \int_0^{\pi} \beta(\psi,\lambda) sin\psi d\psi \quad [m^{-1}], \tag{7}$$

which describes the loss of photons from a beam of light into all directions per distance. Both, absorption and scattering coefficient add up to form the attenuation coefficient:

$$c(\lambda) = a(\lambda) + b(\lambda) \quad [m^{-1}] \tag{8}$$

Scattering is a two-folded phenomenon. Light is scattered out of the beam, causing a decrease in irradiance. In addition, multiple scattering events also cause photons to be scattered into a beam of light causing an increase of irradiance. The loss is simply modeled using the scattering coefficient $b(\lambda)$ in

$$E(z,\lambda) = E(0,\lambda)e^{-zb(\lambda)}. \tag{9}$$

Modeling the gain is more complicated: scattering is often partitioned into forward scattering $\psi \in [0,\pi/2]$ and backward scattering $\psi \in [\pi/2,\pi]$. Several models exist that show, that forward scattering happens mostly in the very small angles at the beginning of the interval and is therefore approximated by a linear filter [Vos91], which is in essence a low pass filter [SK04].

Modeling backward scatter is more elaborate and involves using the volume scattering function explicitly, which is why it needs to be parametrized. Measuring the volume scattering function in natural water bodies is difficult, however some widely used measurements for reference exist in the literature [Pet72]. Those measurements are used to verify different parametrizations of the VSF as described in [Mob94]. We decided to use a combination of the Einstein-Smoluchowski and the Kopelevich parametrization. The Einstein-Smoluchowski model describes sea water scattering without particles with the model equation for the volume scattering function:

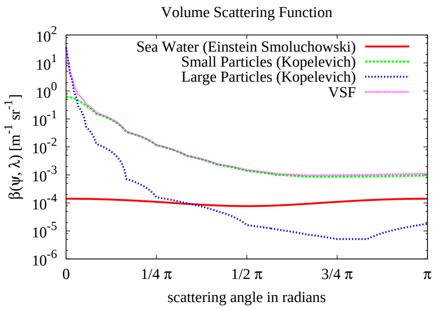

Figure 4: *Volume Scattering Function. Shown is the resulting function (magenta) and its additive components derived from Einstein Smoluchowski (red), Kopelevich small particles (green) and Kopelevich large particles (blue).*

$$\beta_w(\psi,\lambda) = \beta_w(90^\circ,\lambda_0)\left(\frac{\lambda_0}{\lambda}\right)^{4.32} \tag{10}$$

where values for λ_0, $\beta_w(90^\circ,\lambda_0)$, and the final $b(\lambda)$ as well as a model description can be found in [Mob94]. The Kopelevich model is used to add scattering at large and small particles in the water:

$$\beta(\psi,\lambda) = \beta_w(\psi,\lambda) + \nu_s\beta_s^*(\psi)\left(\frac{\lambda_0}{\lambda}\right)^{1.7} + \nu_l\beta_l^*(\psi)\left(\frac{\lambda_0}{\lambda}\right)^{0.3} \tag{11}$$

where ν_s and ν_l are the volume concentrations of small and large particles respectively, and $\beta_s^*(\psi)$ and $\beta_l^*(\psi)$ the volume scattering function for small and large particles per unit volume concentration. Values for a discrete set of angles for the β^* are given and interpolated, the concentrations ν_s and ν_l can be varied.

Both scattering and absorption are modeled in the Jaffe-McGlamery model.

4.3. Jaffe-McGlamery Model

Jaffe's work [Jaf90] was motivated by the development of a simulator for the design of underwater imaging systems. The goal in our case is developing a simulator for underwater images to be used for testing different computer vision algorithms on underwater images. Therefore, we adopt the model to incorporate color, several light sources, and shadows. In order to model the imaging process with consideration of attenuation and scattering, the irradiance being incident up on the pixel sensor is considered to be the sum of three different components:

$$\begin{aligned} E_T(total) = E_D(direct) + E_{fs}(fowardScatter) \\ + E_{bs}(backScatter) \end{aligned} \tag{12}$$

The following paragraphs describe each component. Due to the refractive camera model introduced above, which now

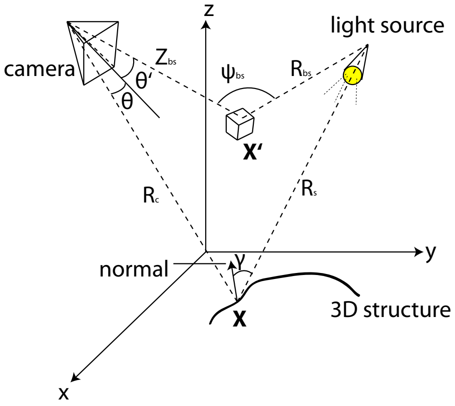

Figure 5: *Adapted from [Jaf90], this images shows the rays for signal and backscatter computation.*

needs to be incorporated into the Jaffe model, the distance between camera and object is considered to be the measured distance between object and glass interface of the camera's housing, which is the natural definition due to it being the actual distance the light traveled through water.

Direct Light (Signal) Light that travels directly from the light source to the object, is reflected and travels directly to the camera, while being attenuated on the way is investigated in this section. As already mentioned, attenuation unifies two effects: absorption and scattering of photons out of the beam described by the attenuation coefficient c. We adapt the model to deal with several light sources $L_j, j < M$ and use point light sources, being described by a position in space and their power, depending on the wavelength λ, $L_j(\lambda)$ in $[W]$. For the geometric computation, a point $\mathbf{x}$ is unprojected to a ray in water as described above. The ray is then intersected with the structure yielding the point $\mathbf{X}$ (fig. 3). Light from a light source L_j only reaches the camera if there is no other part of structure shadowing $\mathbf{X}$. The irradiance being incident upon the structure is then (fig. 5):

$$E_I'(\mathbf{X},\lambda) = \sum_{j<M} E_{I_j} \tag{13}$$

$$E_{I_j} = \begin{cases} 0 & \text{if } \mathbf{X} \text{ shadowed} \\ L_j(\lambda) cos\gamma_j \frac{e^{-c(\lambda)Rs_j}}{Rs_j^2} & \text{else} \end{cases} \quad \left[\frac{W}{m^2}\right]$$

with $\mathbf{X}$ denoting the coordinates on the structure in 3D space, Rs_j is the distance between the light source and $\mathbf{X}$, γ_j is the angle between the ray from the light source L_j and $\mathbf{X}$ and the structure's normal.

As described above, small angle forward scattering adds to the irradiance E_I'. In the Jaffe-McGlamery model, such light, being scattered forward at small angles, is generally modeled linearly by convolution:

$$E_I(\mathbf{X},\lambda) = E_I'(\mathbf{X},\lambda) * g(\mathbf{x}|\overline{Rs}, G, c(\lambda), B) + E_I'(\mathbf{X},\lambda) \tag{14}$$

$$g(\mathbf{x}|\overline{Rs}, G, c(\lambda), B) = \left[e^{-G\overline{Rs}} - e^{-c(\lambda)\overline{Rs}}\right] \mathcal{F}^{-1}\{e^{-B\overline{Rs}f}\}$$

with g being the filter mask dependent on the two empirical values G and B, and $\mathcal{F}^{-1}$ being the inverse Fourier transform. $\overline{Rs} = \frac{1}{M}\sum_{j=1}^{M} Rs_j$ is the mean of all distances Rs_j for the different light sources, an addition to be able to incorporate several light sources. Using the linearity of convolution, this approximation allows modeling several light sources with low computational overhead. Since parts of this convolution are modeled by empirical parameters and Schechner and Karpel [SK04] noted that the linear convolution is a low pass filter, we approximate (14) with a Gaussian filter, for which the filter mask depends on the distance the light traveled. The amount of forward scatter added is weighted by one empirical factor K similar to [TOA06].

In order to compute the reflected light and attenuate it properly, the Jaffe-McGlamery model uses a reflectance map $M(\mathbf{X})$, which is embedded in the x-y-plane of the coordinate system. Instead, we use only one reflectance coefficient M, and can model arbitrary 3D structures specified by 3D models. While traveling from the structure to the camera after reflection, the light is attenuated again. In addition, the camera itself is usually no ideal pinhole camera and effects like vignetting, f-number, and lens transmittance are combined in the fundamental radiometric relation, which turns scene radiance into irradiance incident upon the pixels of the sensor [Sze11]:

$$E_d(\mathbf{x},\lambda) = \tag{15}$$

$$\underbrace{\frac{E_I(\mathbf{X},\lambda)e^{-c(\lambda)R_c}M(\lambda)}{\pi}}_{\text{Scene Radiance}} \underbrace{\frac{cos^4\theta T_l(R_c - F_l)^2\pi}{4 f_n R_c^2}}_{\text{Camera Transmittance}} \quad [Wm^{-2}],$$

with R_c being the distance between $\mathbf{X}$ and the camera. The angle θ is used to describe the angle between the incoming ray and the camera's optical axis. The cosine term models vignetting (refer also to [Sze11]), T_l is the lens transmittance, f_n the camera's f-number, and f_l is the camera's focal length in mm.

Forward Scatter (Signal) The forward scattered part of the signal can easily be derived as seen above, using the distance dependent Gaussian filter:

$$E_{fs}(\mathbf{x},\lambda) = E_d(\mathbf{x},\lambda) * g(\mathbf{x}|R_c, G, c(\lambda), B) \tag{16}$$

The irradiance of the signal incident upon pixel $\mathbf{x}$ has now been derived including the proportion of light coming directly from the light source into the camera via reflection on the object and including part of the scattering proportion, namely, a proportion of small angle forward scattering. Small angle forward scattering has been found to increase signal irradiance, but mainly cause the image to blur depending on the distance between camera and object.

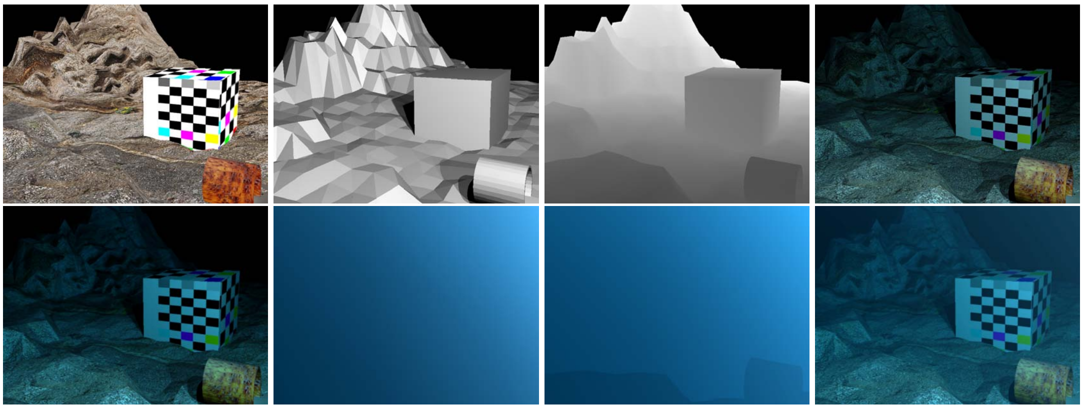

Figure 6: *Lamps were on the right hand side of the camera. Single components, top row from left to right: scene, 3D surface with shadows, depth map, light incident on 3D structure. Bottom row: signal, first backscatter plane, overall backscatter, complete result. Please look at the electronic version due to contrast issues.*

Backscatter (Veiling Light) The part that is still missing, is backscatter. Light is scattered in all directions multiple times, even one light source has numerous beams that loose photons due to scattering, which can then be scattered into other beams. Those scattering events happen so often that backscatter can be viewed as some kind of ambient or veiling light being present in the water in vicinity to a light source or close to the water surface during the day.

As stated in McGlamery's work, the 3D-space through which the back scatter is accumulated, is sliced into N planes of thickness Δz_i that are parallel to the camera's image plane. For each of those planes, the amount of irradiance incident upon it due to backscatter, can be computed similar to above, including the forward scattered amount of light. After that, the irradiance based on the back scatter portion of the light, is the superposition of all the slices. Again, the original model is extended to incorporate several light sources for backscatter computation. This creates the need of applying the VSF to the irradiance on the backscatter plane at this point instead of later as in the original model, which is possible due to the linearity of convolution. The result is:

$$E_s(\mathbf{X'}) = E_{s,d}(\mathbf{X'}) + E_{s,fs}(\mathbf{X'}) \quad \left[Wm^{-2}sr^{-1}m^{-1}\right] \tag{17}$$

$$E_{s,d}(\mathbf{X'}) = \sum_{j<M} E_{s,d_j}$$

$$E_{s,d_j} = \begin{cases} 0 & \text{if } \mathbf{X'} \text{ shadowed} \\ L(\lambda)_j \frac{e^{-c(\lambda)Rbs_j}}{Rbs_j^2} \beta(\psi_j,\lambda) & \text{else} \end{cases}$$

$$E_{s,fs}(\mathbf{X'}) = E_{s,d}(\mathbf{X'}) * g(\mathbf{X'}\ \overline{Rbs}, G, c(\lambda), B),$$

$\beta(\psi,\lambda)$ denotes the volume-scattering function, and the angle ψ_j the angle between a line from the volume to the light source and a line from the volume to the camera (see 5). Following McGlamery, the resulting direct back scatter component can be derived as:

$$E_{bs,d}(\mathbf{x}) = \sum_{i=1}^{N} \underbrace{\frac{e^{-c(\lambda)Zbs_i} E_s(\mathbf{X'},i)\Delta z_i}{cos\theta(\mathbf{X'})}}_{\text{Backscatter (Scene Radiance)}} \underbrace{\frac{cos^4\theta T_l(Zbs_i - F_l)^2\pi}{4 f_n Zbs_i^2}}_{\text{Camera Transmittance}} \tag{18}$$

where Δz_i denotes the thickness of the back scattering volume slice, Zbs_i is the distance between the center point of the slice i and the camera, and i is the index of the back scatter slab. (18) can again be split into a scene radiance part due to backscatter and a camera transmission part, identical to the one in (15), which transfers scene radiance into pixel irradiance (refer also to fig. 3).

Wavelengths The effects described above are wavelength dependent. This is especially true for the absorption and scattering coefficients. The natural extension of the Jaffe-McGlamery model to incorporate colors, is modeling the 3 color channels using 3 discrete bands of frequencies for λ with wavelengths according to the color channels. This concludes the model introduction. The next section will present some rendering results.

5. Results

Due to the need of using our own camera model, it was difficult using ready-made software packages for the implementation. Therefore, we build upon BIAS (www.mip.informatik.uni-kiel.de) and the Open Source library OpenSceneGraph (www.openscenegraph.org/projects/osg) in order to build the renderer. The run-time for rendering one image is in the order of a few minutes, however, we never targeted at running in real-time - being close to the physical models was of greater importance. The motivation for

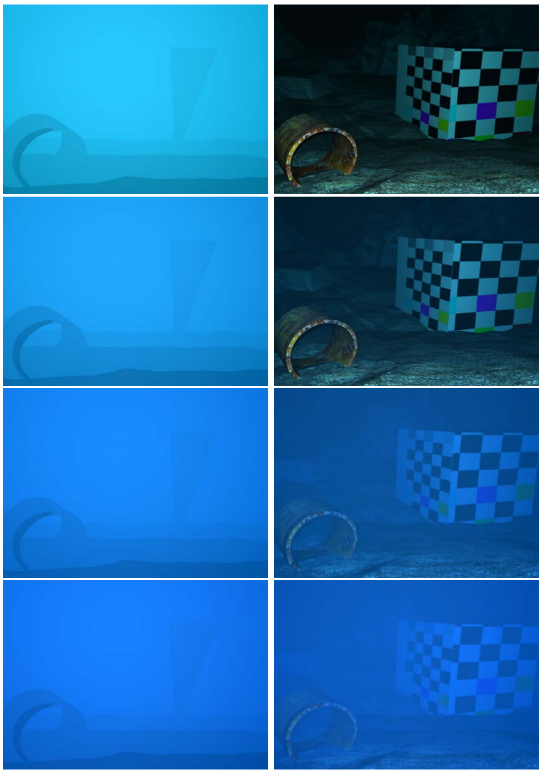

Figure 7: *In all 4 cases the lamps were positioned close to the camera. Left column: backscatter component, right column: complete result. From top to bottom, increasingly turbid water. Please look at the electronic version due to contrast issues.*

the simulator was creating synthetic images as captured by a ROV. ROVs usually carry a set of lamps attached to the front of the frame, often at the top, but approximately in the same plane as the camera. Our implementation contains an editor allowing to create a set of such lamps, which can be positioned very close to the camera, but also be moved further away.

In order to test the simulator, we build a synthetic model of an underwater scene and added a cube with a checkerboard pattern. Figure 6 shows the different components from the rendering process. The surface rendering clearly shows the shadows produced by 3 lamps to the upper right of the camera. Geometric ground truth is rendered in form of depth maps recording the camera-3D point distance for each pixel. The fourth image shows the light that is incident on the structure including the texture and reflection. After adding forward scatter and attenuating on the way to the camera, the signal (fifth image) results. The first backscatter plane is not yet occluded by the structure and clearly shows the decrease of the backscatter with growing distance from the light source. In the result image, the background structures are visible in contrast to the signal image because they are illuminated by the backscatter.

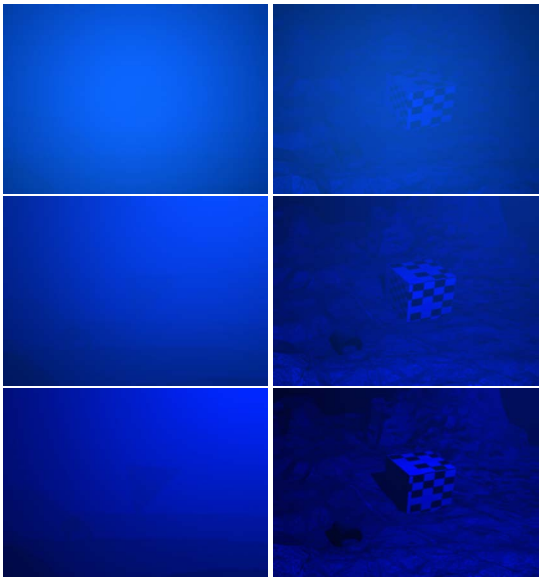

Figure 8: *The lamp was moved from being directly at the camera to* $2m$ *at the upper right. As Jaffe already concluded in his paper, the backscatter portion is causing the contrast to decrease if the lamp is close to the camera. The left column shows the backscatter component and the right column the resulting images. Please look at the electronic version due to contrast issues.*

5.1. Backscattering Effects

Rendering results for this scene with increasingly turbid water, e.g. increasing backscatter, can be seen in figure 7. The left column shows the backscatter component, while the right column shows the complete rendering result. As can be noticed, the contrast diminishes with growing concentration of scattering particles. One of the major conclusions in [Jaf90] concerned the placement of the lamps relative to the camera and the type of lamp in order to receive usable images from deep sea environments: there is always a compromise between contrast and power reaching the imaging system due to backscatter and attenuation. This can be observed in figure 8, where the lamp was moved away from the camera up to a distance of approximately $2m$.

5.2. Refractive Effects

If the necessary underwater housing is modeled explicitly, strong refractive effects are visible in the image. In addition, the object appears to be enlarged by a factor of 1.333, the index of refraction of water. Both effects can be observed in fig. 9 when comparing the left image (with underwater

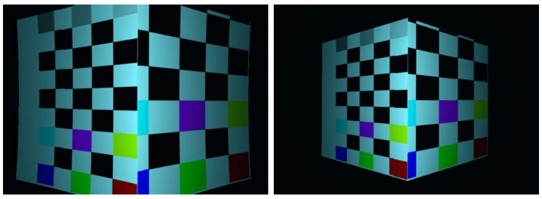

Figure 9: *Left: imaged with explicitly modeling refraction at the underwater housing. A perspective camera without any radial distortion is set into the housing. Right: rendered without the housing using the same perspective camera as on the left. Note the change in size and distortions introduced by refraction.*

housing) with the right image (no underwater housing. In the left image, the cube not only appears to be closer, but is also distorted at it's edges. The underwater housing in this case was parametrized with a glass thickness of $4cm$, the interface distance was $2cm$, and the interface normal was rotated by $0.1°$.

6. Conclusion and Future Work

We have presented a renderer for deep sea underwater images that utilizes two physical models, one for light propagation within the water and one for refracting the light rays at the underwater housing. In order to achieve realistic images, the Jaffe-McGlamery model has been extended to incorporate color channels and several light sources. In addition, shadows can be rendered. Explicitly modeling refraction at the water-glass-air interface allows to render the distortion effects typical for underwater images. The simulator therefore serves as a basis for generating underwater images with known ground truth compliant to the physical models that can be used to develop or adapt and evaluate computer vision algorithms for underwater scenarios.

In the future, we plan to improve efficiency by parallelizing the computation and to model material properties other than diffuse reflection. In addition, floating particles can add more realism. The simulator will be utilized in order to develop a Structure-from-Motion system that can incorporate refraction and does not break down due the low contrast common in underwater images. This computer vision system uses the refractive camera model described and simulated here.

References

[CP09] C. Papadopoulos G. P.: Realistic real-time underwater caustics and godrays. In *Proc. GraphiCon '09* (2009). 2

[CSCS02] Cerezo E., Serón F. J., Cerezo E., Serón F. J.: Rendering natural waters: merging computer graphics with physics and biology. In *in: Proceedings of Computer Graphics international CGI'02, 2002* (2002), pp. 481–498. 2

[DCGG11] Darles E., Crespin B., Ghazanfarpour D., Gonzato J.-C.: A survey of ocean simulation and rendering techniques in computer graphics. *Comput. Graph. Forum 30*, 1 (2011), 43–60. 2

[GSMA08] Gutierrez D., Seron F., Muñoz A., Anson O.: Visualizing underwater ocean optics. *Computer Graphics Forum (Proc. of EUROGRAPHICS) 27*, 2 (2008), 547–556. 2

[Hec05] Hecht E.: *Optik*. Oldenburg Verlag Muenchen Wien, 2005. 1, 3

[HZ04] Hartley R., Zisserman A.: *Multiple View Geometry in Computer Vision (Second Edition)*, second ed. Cambridge University Press, 2004. 1

[IDN02] Iwasaki K., Dobashi Y., Nishita T.: An efficient method for rendering underwater optical effects using graphics hardware. *Computer Graphics Forum 21*, 4 (2002), 701–711. 2

[Jaf90] Jaffe J.: Computer modeling and the design of optimal underwater imaging systems. *IEEE Journal of Oceanic Engineering 15*, 2 (1990), 101–111. 1, 3, 4, 5, 7

[JC98] Jensen H. W., Christensen P. H.: Efficient simulation of light transport in scences with participating media using photon maps. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1998), SIGGRAPH '98, ACM, pp. 311–320. 2

[KS08] Kunz C., Singh H.: Hemispherical refraction and camera calibration in underwater vision. In *OCEANS 2008* (15-18 2008), pp. 1 –7. 2

[McG75] McGlamery B. L.: *Computer Analysis and Simulation of Underwater Camera System Performance*. Tech. rep., Visibility Laboratory, Scripps Institution of Oceanography, University of California, San Diego, 1975. 1, 3

[Mob94] Mobley C. D.: *Light and Water: Radiative Transfer in Natural Waters*. Academic Press, 1994. 1, 2, 4

[NCB09] Nascimento E. R. d., Campos M. F. M., Barros W. F. d.: Stereo based structure recovery of underwater scenes from automatically restored images. In *Proceedings SIBGRAPI 09 (Brazilian Symposium on Computer Graphics and Image Processing)* (Los Alamitos, Oct. 11–14, 2009 2009), Nonato L. G., Scharcanski J., (Eds.), IEEE Computer Society. 1

[PA01] Premoze S., Ashikhmin M.: Rendering natural waters. *Comput. Graph. Forum 20*, 4 (2001), 189–199. 2

[PARN04] Premoze S., Ashikhmin M., Ramamoorthi R., Nayar S. K.: Practical rendering of multiple scattering effects in participating media. In *Rendering Techniques* (2004), pp. 363–373. 2

[Pet72] Petzold T. J.: *Volume Scattering Functions for Selected Ocean Waters*. Tech. rep., Scripps Institution of Oceanography, University of California, San Diego, 1972. 4

[SK04] Schechner Y. Y., Karpel N.: Clear underwater vision. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2004* (June 27–July 2, 2004), vol. 1, pp. I–536–I–543. 1, 4, 5

[Sze11] Szeliski R.: *Computer Vision Algorithms and Applications*. Springer-Verlag, 2011. 5

[TOA06] Trucco E., Olmos-Antillon A. T.: Self-tuning underwater image restoration. *IEEE JOURNAL OF OCEANIC ENGINEERING 31*, 2 (APR 2006), 511–519. 1, 5

[TS08] Treibitz T., Schechner Y. Y.: Active polarization descattering. *IEEE Transactions on Pattern Analysis and Machine Intelligence 31* (2008), 385–399. 1

[Vos91] Voss K. J.: Simple empirical model of the oceanic point spread function. *Appl. Opt. 30*, 18 (Jun 1991), 2647–2651. 4