



An Optical Digital Twin for Underwater Photogrammetry

GEODT—A Geometrically Verified Optical Digital Twin for Development, Evaluation, Training, Testing and Tuning of Multi-Media Refractive Algorithms

David Nakath¹  · Mengkun She¹ · Yifan Song¹ · Kevin Köser¹

Received: 30 September 2021 / Accepted: 21 December 2021 / Published online: 21 January 2022

© The Author(s) 2022

Abstract

Most parts of the Earth's surface are situated in the deep ocean. To explore this visually rather adversarial environment with cameras, they have to be protected by pressure housings. These housings, in turn, need interfaces to the world, enduring extreme pressures within the water column. Commonly, a flat window or a half-sphere of glass, called flat-port or dome-port, respectively is used to implement such kind of interface. Hence, multi-media interfaces, between water, glass and air are introduced, entailing refraction effects in the images taken through them. To obtain unbiased 3D measurements and to yield a geometrically faithful reconstruction of the scene, it is mandatory to deal with the effects in a proper manner. Hence, we propose an optical digital twin of an underwater environment, which has been geometrically verified to resemble a real water lab tank that features the two most common optical interfaces. It can be used to develop, evaluate, train, test and tune refractive algorithms. Alongside this paper, we publish the model for further extension, jointly with code to dynamically generate samples from the dataset. Finally, we also publish a pre-rendered dataset ready for use at <https://git.geomar.de/david-nakath/geodt>.

Keywords Underwater computer vision · Refractive geometry · Underwater dataset · Digital twin

Zusammenfassung

Ein optischer digitaler Zwilling für Unterwasser-Photogrammetrie. Der größte Teil der Erde ist von der Tiefsee bedeckt. Um diese visuell herausfordernde Umgebung mit Kameras zu explorieren, müssen diese durch Druckgehäuse geschützt werden. Letzere benötigen wiederum optische Schnittstellen zur Außenwelt, die dem extremen Druck in der Wassersäule standhalten müssen. Diese werden normalerweise in Form eines flachen Glasfensters (Flat-Port) oder einer Glashalbkugel (Dome-Port) realisiert. Dadurch entstehen multi-media Schnittstellen zwischen Wasser, Glas und Luft, die entsprechende Brechungseffekte nach sich ziehen. Um korrekte 3D-Messungen oder geometrisch verlässliche Rekonstruktionen einer Szene zu erlangen, müssen diese Effekte bedacht werden. Daher publizieren wir einen geometrisch verifizierten optischen digitalen Zwilling eines wissenschaftlichen Wasser-Testtanks, welcher die beiden gängigsten optischen Schnittstellen aufweist. Dieser kann genutzt werden, um Brechungs-Algorithmen zu entwickeln, testen, trainieren, verbessern und schließlich zu evaluieren.

✉ David Nakath
dnakath@geomar.de

Mengkun She
mshe@geomar.de

Yifan Song
ysong@geomar.de

Kevin Köser
kkoeser@geomar.de

¹ GEOMAR Helmholtz Centre for Ocean Research Kiel,
Wisshofstrasse 1-3, 24148 Kiel, Germany

Wir veröffentlichen außerdem unser Model, vorgerenderte Bilder und den Code zum synthetisieren weiterer Bilder unter: <https://git.geomar.de/david-nakath/geodt>.

1 Introduction

The biggest part of Earth's surface is covered by the deep sea (Eakins and Sharman 2012). Hence, vast amounts of the seafloor and the majority of the water column above it is yet to be thoroughly explored. Cameras have to be protected from salt water and their housings must sustain enormous pressures of approximately 1 bar per 10 m of depth. This especially holds true for the optical windows, the *ports*, of the housings. Glass domes, so called dome ports, are mechanically very stable and require thicknesses of up to one centimeter for commonly used dome diameters. The stability of flat ports depends strongly on their size and material, where larger windows quickly require thicknesses of many centimeters. Light rays collected by lenses behind these ports traverse different media and are refracted at the interfaces, which complicates underwater photogrammetry and associated applications of computer vision (Fig. 1).

The ocean can be coarsely separated into euphotic, disphotic and aphotic light zones. The bottom of the first zone is defined at 200[m] where only 1% of the surface photosynthetic available radiation (PAR) is remaining. Furthermore, no significant portion of sunlight reaches depths below and is totally extinct after 1000[m], which marks the beginning of the aphotic zone (Kirk 1994). Hence, deep ocean photogrammetry needs artificial light sources, which also have to be accommodated in the same kind of housings as cameras. In such a scenario, also the cones of the lights are subject to refraction effects.

As ship time is expensive, and working on the ship is very demanding and allows only limited modifications of a system, it is desirable to test, tune and verify sensors and corresponding algorithms up front. In our experience, it is a good development practise to follow a development model with increasing complexity levels:

1. test correctness and stability of the core measurement model, observation equations and estimation algorithms by unit tests and numeric simulations
2. simulate as realistic as possible sensor data (here images) to evaluate the algorithm end-to-end, with the same pipeline to be used on real data
3. repeat the above experiments in a controlled, but real, setting (e.g. test tank), with increasing complexity
4. finally, perform experiments in the ocean

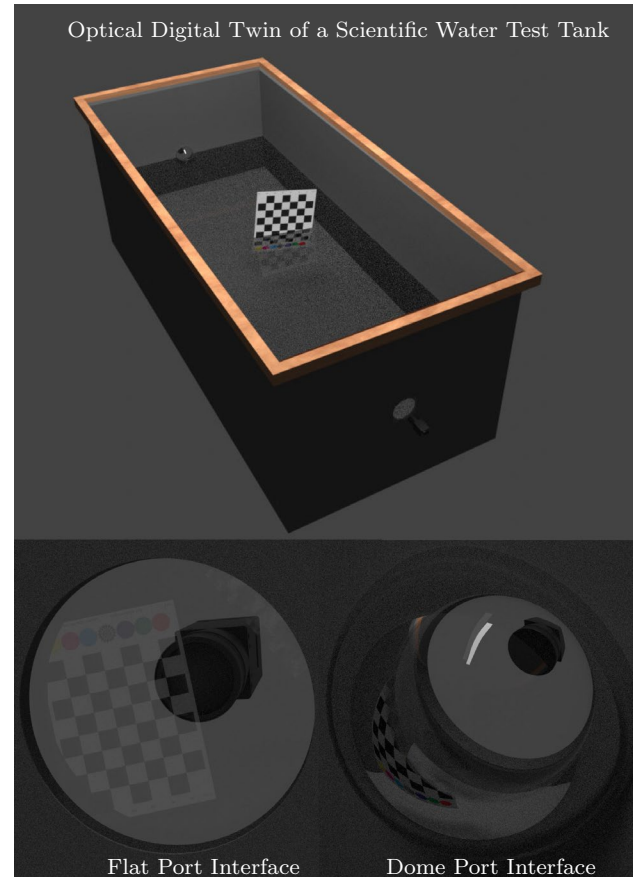


Fig. 1 Geometrically verified optical digital twin of a water lab tank featuring a dome port and a flat port interface

In particular, steps 2 and 3 are important to understand issues and limitations of a photogrammetric system, when the data becomes more realistic. We have, therefore, built a tank that allows to attach underwater cameras to test underwater imaging algorithms. Still, setting up experiments here means substantial effort and many effects can be observed already in simulated data. For underwater photogrammetry applications, in particular refraction is important, and we observe that for a large part the photogrammetry community is not considering refraction explicitly, potentially due to a lack of supporting software and high costs / burdens to set up underwater equipment.

To facilitate the development of refractive algorithms, from calibration to multiview relation estimation, bundle adjustment or dense stereo reconstruction, we therefore provide a geometrically verified virtual test environment that provides easy access to refraction effects both with dome and flat ports, where users can set water properties and of course also add other objects or scenes as needed. Actually, implementing and verifying such a model takes substantial time and might block people from further research in this direction, which is why we want to make our efforts available to others.

1.1 Contribution and Outlook

In this paper, we specifically contribute the following: (i) we devise an optical digital twin of an underwater setting (real lab tank), which (ii) has been geometrically verified against a numerical simulation and real imagery. Furthermore, we will publish a (iii) Blender-based dataset-generator with a convenient YAML-based interface. Finally, we will (iv) publish a pre-rendered dataset for the dome-port and the flat port interfaces for the conditions *no-water*, *half-water* and *full-water*.

The remainder of this paper is structured as follows. In the subsequent Sect. 2, we will present related work in the fields of refractive geometry and underwater image simulations. We will then turn to a detailed description of the water tank environment and its geometric optical digital twin (GEODT) in Sect. 3. The geometric verification of the just-introduced digital twin will be presented in Sect. 4. In Sect. 5, the pre-rendered dataset and the interface for dataset-generation will be described in detail. Finally, this paper will conclude with Sect. 6.

2 Related Work

2.1 Refractive Geometry

It is well known that refractions are an integral part of the underwater image-formation model and thus have to be carefully taken into consideration in photogrammetric applications, see e.g., (Shmutter 1967; Moore 1976; Kotowski 1988; Fryer and Fraser 1986) as well as (Harvey and Shortis 1998; Jaffe et al. 2001; Kunz and Singh 2008; Drap 2012).

Underwater imaging systems involving flat ports actually become axial cameras (Treibitz et al. 2008), and refraction at such camera housings, considered two times for thick glass, significantly complicates forward projection (Agrawal et al. 2010) and structure from motion (Jordt 2014; Jordt et al. 2016). Exactly centering a pinhole camera inside a dome port on the other hand can avoid refraction of principal rays, but doing so requires some effort (She et al. 2019; Menna

et al. 2016). Decentered dome systems also become axial cameras, though with different geometry (She et al. 2022) and suffer from refraction (Menna et al. 2020). For both, dome and flat ports, efficient refraction models, approximations and algorithms are still an active area of research (see e.g., (Nocerino et al. 2021; Menna et al. 2017) as well as (Jordt and Koch 2011; Mulsow and Maas 2014; Duda and Gaudig 2016; Hu et al. 2021)).

2.2 Simulated Underwater Datasets

While real underwater datasets are—of course—the most desirable kind of data, it remains costly and difficult to obtain them. In addition, it is extremely challenging and sometimes even impossible to obtain ground truth by annotating the data or even by taking independent measurements. Hence, simulated datasets are a valid option, too—provided they can synthesize images with a satisfactory quality and accuracy.

On the simulation side, there exists some prior work especially on the simulation of Autonomous Underwater Vehicles (AUVs) equipped with cameras. The holistic AUV simulator UW Sim (Prats et al. 2012), which simulates an AUV and its sensor-suite, comprises a simple underwater-camera. The UUV-simulator (Manhães et al. 2016) rests on Gazebo (Koenig and Howard 2004) to provide a very comprehensive and interactive AUV-simulation suite. Both approaches, in turn, rely on the Robot Operating System (ROS) (Quigley et al. 2009), to allow for a tight integration with actual robots. Further underwater camera simulators model shallow sea water (Cozman and Krotkov 1997) with the *Fog* model (Nayar and Narasimhan 1999) or deep sea environments (Song et al. 2021) with the Jaffe-McGlamery model (Jaffe 1990; McGlamery 1975).

However, the above approaches neglect the issue of refraction, introduced by water–glass–air interfaces, while the focus mainly rests on the issues of attenuation (Akkaynak et al. 2017) and scattering (Preisendorfer 1964; Mobley et al. 2021). Above the water, Agrafiotis et al. (2021) synthesized images taking the refractive surface of water into account. Underwater, Kahmen et al. (2019) employed a refracted projection for multi camera systems for flat interfaces, which basically corresponds to our numerical verification approach (She et al. 2022; Jordt-Sedlazeck and Koch 2012; Kunz and Singh 2008). Also for flat ports, Sedlazeck and Koch (2011) proposed a Jaffe–McGlamery-based (Jaffe 1990; McGlamery 1975) image formation model that customly added refraction effects. While having been a great tool at the time, due to the rasterization-based technique of the renderer, volumetric effects are only approximated coarsely in a post-processing step and the system was hand-crafted for a particular flat-port.

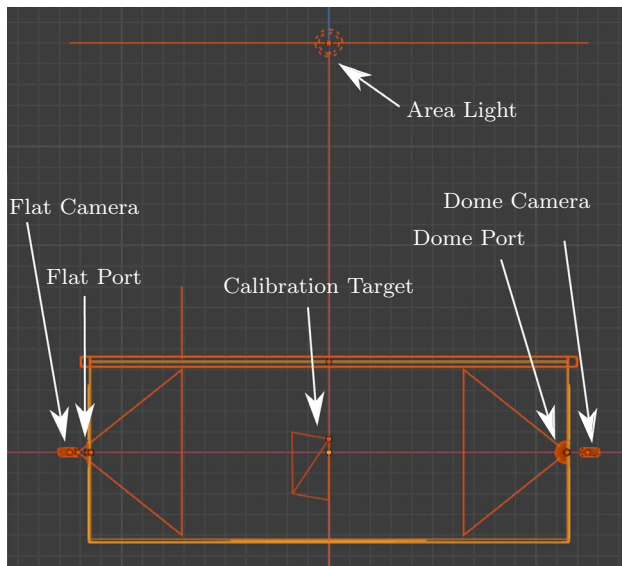


Fig. 2 Schematic setup of the lab tank

In *rasterized* rendering approaches, geometry is transformed into the image space in a feed forward process. Optical effects occurring in the process have to be described by approximate models. Physically based rendering is an alternative way of synthesizing images (Pharr et al. 2016). In such a *raytracing* (Whitted 1980) approach light rays are shot through a scene, and their behavior is defined based on physical models. Multiple rays are shot per image pixel and the computed intensities are subsequently integrated to obtain a color value. Finally, a realistic—and physically sound—image can be obtained by repeating this process for every pixel. A well-known software bundle to design scenes and perform raytracing on them is Blender (Blender Community 2018). In Zwilmeyer et al. (2021), it is used to simulate underwater images, however, totally neglecting the issue of refraction. We too use Blender as a dataset generator, with a special focus on refraction at the interfaces between media with differing optical densities.

3 GEODT—A Geometrically Verified Optical Digital Twin of a Scientific Lab Tank

We model an actually existing water lab tank, which is in every day use, to virtually make it available to the underwater-photogrammetry community, for testing, development, training, and tuning purposes.

3.1 General Setup

To resemble the real water tank as close as possible (see Fig. 2), we took the following steps. We model the tank as

such, a light source on top and the glass interfaces: (i) dome-port and (ii) flat-port on each side. We model a custom-built Develogic dome, as a 7[mm] thick spherical dome made from glass with an inner radius of 50[mm]. It can endure up to 6000[m] water depth using Vitrovex¹ glass, which in turn is based on Schott's Duran glass 3.3². The flat port is modeled as a glass plate with a thickness of 0.014[m], also using Vitrovex glass. The two sides also feature two pinhole cameras, which are designed to resemble a Basler acA1300-200um machine vision camera with a resolution of 1280 × 1024[px] and a field of view of 73[deg]. The dimensions of the tank are 0.8 × 1 × 2.3[m]. Hence, it can roughly accommodate 1800[l] of water, which is modeled as a water body with a cavity to accommodate the dome port. The latter is necessary to keep the volumetric water effects out of the dome itself. Finally, we place a calibration target in the tank, which—due to its known properties—can be used for calibration, training, as well as for verification purposes. The latter can either be e.g., a checkerboard or a random-dot-pattern-equipped (Li et al. 2013) calibration object (see Fig. 3).

3.2 Volumetric Raytracing

To obtain an image with the raytracing technique in a volumetric setting, some variant of the volumetric rendering equation (VRE) (Novák et al. 2018; Fong et al. 2017), which is a generalization of the rendering equation (Kajiya 1986), has to be solved. As a closed-form solution is usually intractable for any non-trivial scene configuration, Monte Carlo methods are usually employed to approximate the solution (Novák et al. 2018; Veach 1998). In this paper, we specifically use the path tracer of Blender's 2.83 LTS Cycles engine, which is build on top of OptiX (Parker et al. 2010), to obtain the result.

The path tracer needs geometry information, material definitions, and medium definitions as an input for its computations. The geometry is provided by models, we define in Blender itself. We further define all materials as diffuse Bidirectional Scattering Distribution Functions (BSDFs) (see e.g., Pharr et al. 2016), whose reflectance are either defined by a base color or by a texture (e.g., in the case of the calibration targets). In addition, a medium definition is required, to compute the beam transmittance and a phase function, which encodes probable scattering directions. In the following subsections, we will thus additionally give a detailed definition of the latter.

¹ <https://www.vitrovex.com/instrumenthousings/>.

² https://www.schott.com/d/tubing/66ee849c-cad9-42a3-97cb-67fd0534e3fb/schott_duran_brochure_gb_final.pdf.

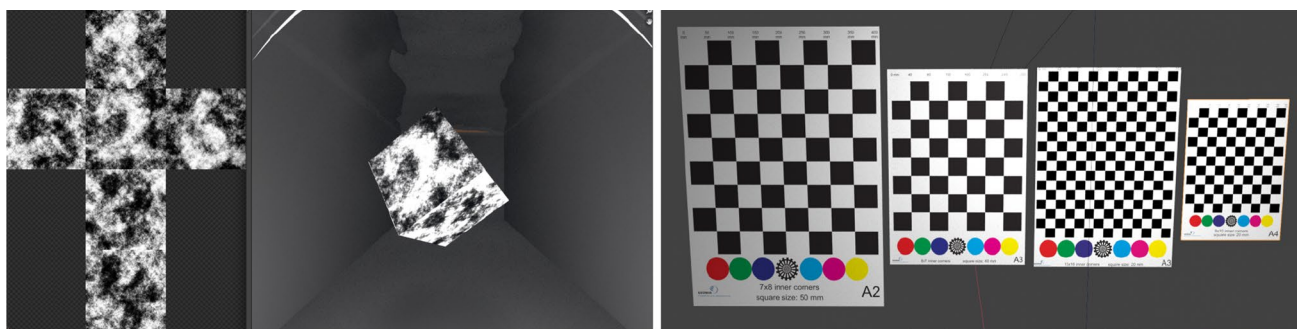


Fig. 3 Pre-defined calibration targets: left unwrapped random dot-pattern texture (Li et al. 2013) and corresponding calibration cube; right assortment of checkerboards. Of course, different calibration targets can be added and used as desired

3.2.1 Homogenous Scattering Medium

Throughout this paper, we assume the water volume as well as the glass volume to be exhaustively defined by a homogeneous scattering medium (see e.g., Pharr et al. 2016). The light propagation in such a medium is governed by the three following equations.

3.3 Attenuation

The attenuation describes the mean free path a ray can travel in the medium, it is given by the sum of absorption and out-scattering

$$\sigma_t = \sigma_a + \sigma_s. \tag{1}$$

Those values can be set for the wideband coefficients R, G, B in Blender.

3.4 Albedo

The albedo gives the scattering ability of the medium by defining the probability of an absorption vs. a scattering event (σ_s), once a particle is hit in the medium. It is given by

$$p_a = \frac{\sigma_s}{\sigma_t}. \tag{2}$$

Again, these values can be set for the wideband coefficients R, G, B in Blender.

3.5 Scattering

The scattering itself has to be carried out in a certain direction, commonly the Henyey Greenstein phase function is employed to describe a distribution over the unit sphere of directions (Henyey and Greenstein 1941)

$$p_{HG} = (\cos\theta) \frac{1}{4\pi} \frac{1 - g^2}{(1 + g^2 + 2g(\cos\theta))^{3/2}}. \tag{3}$$

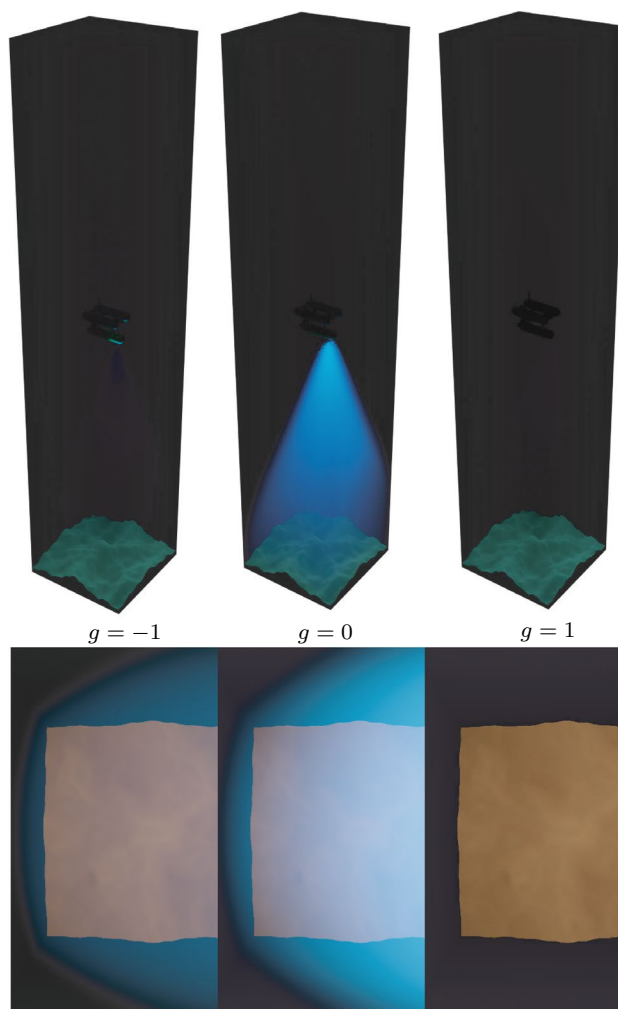


Fig. 4 The Henyey–Greenstein phase function: the g value indicates the mean scattering direction and ranges from -1 full backward-scattering, over $g = 0$ isotropic scattering, to $g = 1$ full forward-scattering. up: view onto the water column; down view through the camera of the AUV

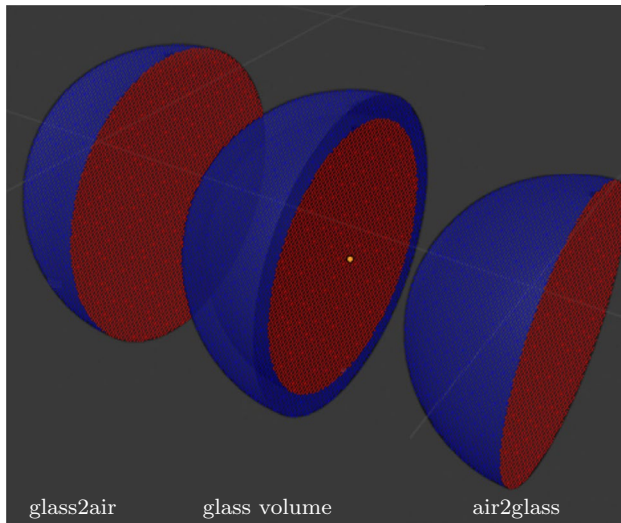


Fig. 5 Setup of the dome interface, from right to the left: first there is the air2glass interface, followed by a volume accounting for the glass body, and finally the glass2air interface. When modeling the interface, the normals of the models have to be taken into account. Here, blue indicates the front of an object (i.e. the direction of the normals), while red indicates the backside. The flat port is modeled in the same manner

It has the mean scattering direction parameter g which can be set in Blender. Its behavior is depicted in Fig. 4.

3.5.1 Modelling of Refractions

Refractions occur at the interfaces between participating media with different optical densities and are governed by Snell's Law (Glassner 1989). It is defined by the ratio of the sine of the angles θ_2 and θ_1 of the in- and outgoing ray w.r.t. the surface normal of an interface

$$\frac{\sin \theta_2}{\sin \theta_1} = \frac{v_1}{v_2} = \frac{ior_2}{ior_1}, \quad (4)$$

which equals to the ratio of the speed of light before v_1 and v_2 after transitioning to the other medium. We will use the reciprocal ratio of the indices of refraction to define the properties of an interface. The actual refraction of a ray within the simulation depends on the incident angle w.r.t. the surface normals of the model we use in the simulation (see Fig. 5), hence we have for the different interfaces

$$air2glass = \frac{ior_{air}}{ior_{glass}}, \text{ and } glass2air = \frac{ior_{glass}}{ior_{air}} \quad (5)$$

as well as

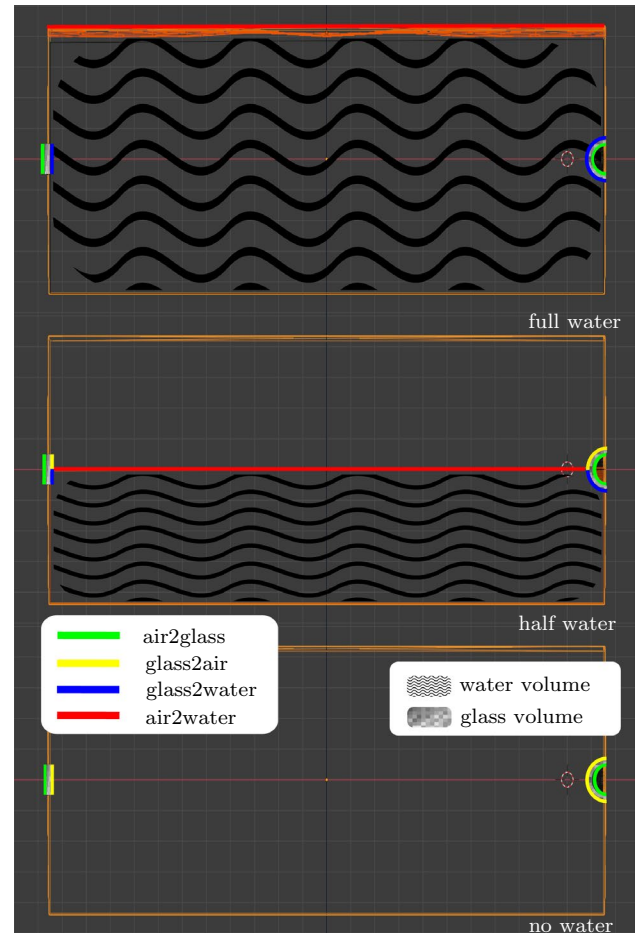


Fig. 6 Holistic medium interface design for the different conditions: down: no water, middle: half water, up: full water

$$glass2water = \frac{ior_{glass}}{ior_{water}}, \text{ and } water2air = \frac{ior_{water}}{ior_{air}}. \quad (6)$$

3.5.2 Holistic Interface Modeling

After definition of all interface types, we modeled three different tank fill-rate configurations for the no-water, half-water, and full-water case, to ensure a proper holistic handling of light rays shot through the scene (see Fig. 6). To enable versatile evaluation strategies, we provide three different water levels, which can be used as a verification step (full vs. no water: can we undo the water effects in the images?) or for information retrieval (half-water-case) in an e.g., calibration approach (She et al. 2019).

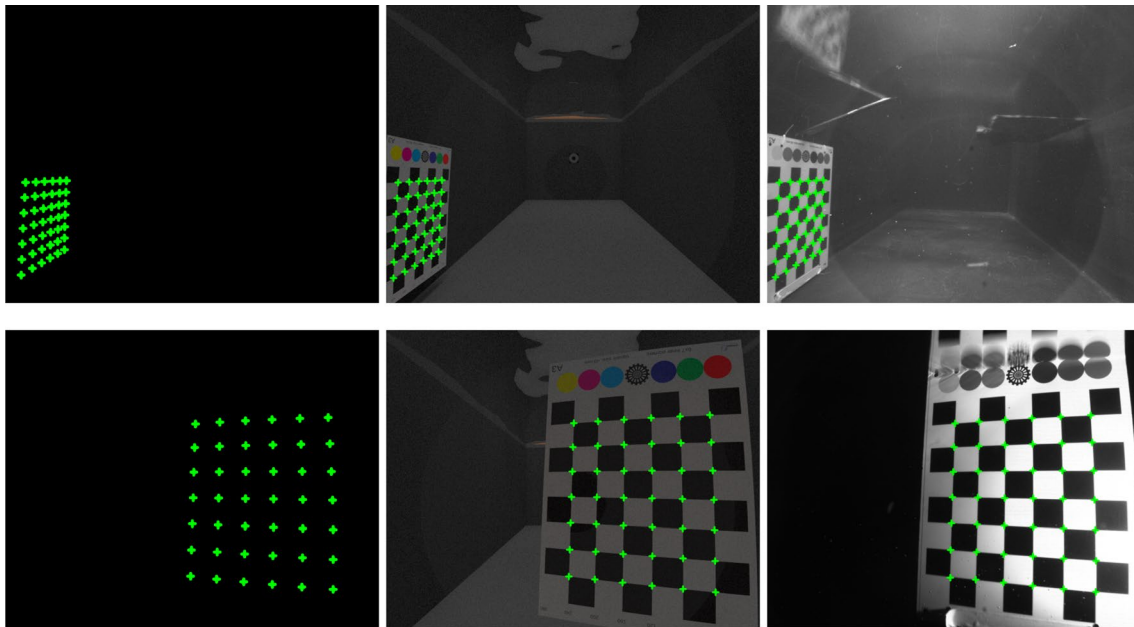


Fig. 7 Geometric Verification. The images synthesized by the *GEODT* (middle) have been verified against a numerical simulation (left) and real images (right). The upper and lower row show two coinciding example poses from the corresponding verification image set

3.6 Full Water

In the *full water case*, the water body is modeled as a homogeneous scattering medium with a surface, which does not interact with the light. In addition, it is carved in to accommodate the dome-port without simulating water inside the port itself. To complete the water-body, its surface is explicitly modeled as an *air2water* interface (see Fig. 6). The dome-port is modeled as an *air2glass*-interface, followed by a *glass-volume* and finally a *glass2water* interface. The flat-port has the same interface/volume structure as the dome-port, it just exhibits a different (i.e. planar) geometry (c.f. Fig. 6).

3.7 Half Water

In the *half water case*, the water body is modeled in the same fashion as in the full-water case. The only difference is, that its height is exactly at the middle of the dome-port and the flat-port to enable direct comparison experiments. The dome-port is now modelled as an *air2glass*-interface followed by a *glass-volume*. To correctly account for the water level, exitant light now passes a splitted interface, where the upper part is modeled as an *glass2air*-interface, while the lower part is a *glass2water*-interface. Again, the flat-port has the same interface structure as the dome-port (c.f. Fig. 6).

3.8 No Water

Finally, in the *no water case*, we simply omit the water body. The dome-port is now modeled as an interface where light enters through an *air2glass* interface, passes the glass medium and exits through a *glass2air* interface. Here, the flat port is again modeled in a similar fashion like the dome port (See Fig. 6).

4 Geometric Verification

4.1 Approach

We verify the simulated lab tank *GEODT* against the two adjacent methods corresponding to step 1 and step 3 of our evaluation pipeline stated in the Introduction Sect. 1: namely *numerical* and *real* tank experiment (see Fig. 7). As an error-measure we chose the 2-norm of the mean the pixel-difference $|\mu_x, \mu_y|_2$ in image space of the detected corners on a known calibration target.

4.2 Setup

For the tank parametrization, we use the numbers stated in Sect. 3.1, to resemble the real tank as close as possible. In addition, we set the index of refraction $ior_{air} = 1.0$

Table 1 Results of the verification experiment using chessboard corners. The upper table shows the x- and y-components of the mean error μ and their standard deviations $\hat{\sigma}$. The lower table shows the 2-norm of the values shown above. All values are given in [px]

vs. / Err:	μ_x	μ_y	$\hat{\sigma}_x$	$\hat{\sigma}_y$
Numeric vs. GEODT	0.08	0.13	0.11	0.21
Real vs. GEODT	0.63	0.49	0.73	0.75
Numeric vs. Real	0.59	0.42	0.69	0.62
vs. / Err:	$ (\mu_x, \mu_y) _2$		$ (\hat{\sigma}_x, \hat{\sigma}_y) _2$	
Numeric vs. GEODT	0.16		0.23	
Real vs. GEODT	0.79		1.05	
Numeric vs. Real	0.73		0.93	

and for the water we use $ior_{water} = 1.333$. Finally, we set $ior_{glass} = 1.473$, as given by the manufacturer.

For further preparation of the actual data, we obtain the in-air intrinsics, of the Basler machine vision camera using standard chessboard calibration. It is given in terms of an opencv pinhole model $\{\mathbf{K}, \mathbf{d}\}$, where \mathbf{K} is the intrinsic matrix and the vector \mathbf{d} denotes the corresponding distortion coefficients. Here, we yield a calibration residual of 0.25[px]. In addition, we take actual photos in the lab tank and find the offset vector \mathbf{v}_o of the camera w.r.t. the dome center and to the checkerboard using (She et al. 2022). The reprojection error after this step is 0.52[px]. After having detected the board pose, we can precisely rebuild the whole scene with the parameters relevant to model the refraction effects, using

$$\mathbf{K}_1 = \begin{bmatrix} 864.9102 & 0 & 640 \\ 0 & 864.9102 & 512 \\ 0 & 0 & 1 \end{bmatrix}, \quad (7)$$

$$\mathbf{K}_2 = \begin{bmatrix} 870.305546 & 0 & 648.059854 \\ 0 & 869.840761 & 514.485508 \\ 0 & 0 & 1 \end{bmatrix}, \quad (8)$$

$$\mathbf{d}_2 = \begin{bmatrix} -0.113777 \\ 0.083360 \\ 0.000459 \\ 0.000180 \end{bmatrix}^T \text{ and } \mathbf{v}_o = \begin{bmatrix} 0.02554 \\ -0.000591841 \\ 0.00364148 \end{bmatrix}^T, \quad (9)$$

where \mathbf{k}_1 holds for the ideal pinhole camera, modeled in Blender, \mathbf{k}_2 is the real camera and $\mathbf{d}_1 = \mathbf{0}$ as well as \mathbf{d}_2 are the respective corresponding distortion parameters. Finally, \mathbf{v}_o denotes the offset vector of the camera w.r.t. the dome in [m] in Blender coordinates. It thus defines the extrinsics, when we know the dome position and assume no rotation w.r.t. it.

To obtain our *real* measurements, we extract the corners from the images taken in the tank and undistort them

Fig. 8 Geometric verification, shown per corner as the relative error over 12 images. Top, left: *real* vs *GEODT* Top, right: *numeric* vs *GEODT* Bottom left: *real* vs *numeric*. Please note that the single plots for the corner detection offsets are extremely magnified, i.e., they are on a different scale, ranging from -1 px to 1 px, w.r.t. the underlying checkerboard. Bottom, right, the used dataset has a versatile pose-set, hence capturing—and thus verifying—many different refractive effects

afterwards. The latter step allows us to investigate the refraction effects in the space of the ideal pinhole camera as simulated by Blender. We then use the available information to rebuild the scene in the *GEODT* and also subsequently extract the corners from the synthesized images. For the corner extraction step, we assume an error of 0.1[px]. Finally, we *numerically* forward project the refracted corners from 3D space to image space (Kunz and Singh 2008). We use the implementation of (She et al. 2022), however, also other implementations like (Jordt-Sedlazeck and Koch 2012) exist.

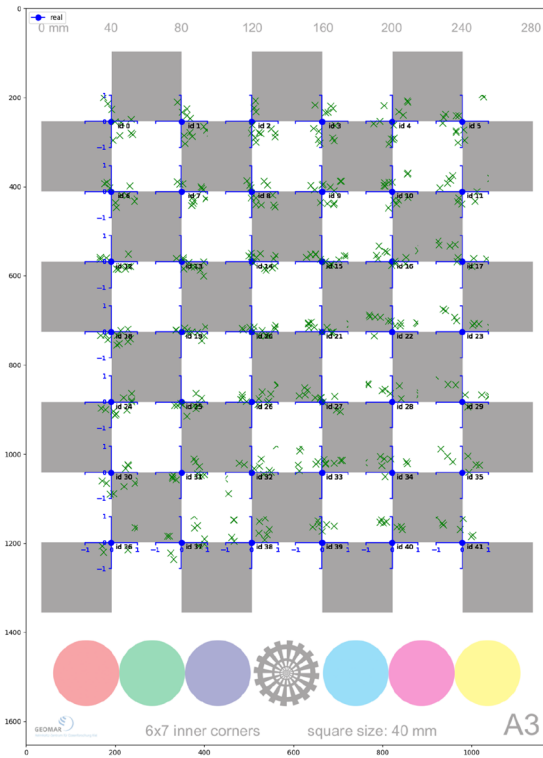
4.3 Results

For verification, we compare all 6×7 corners over 12 checkerboard poses and compute the mean μ as well as the standard deviation $\hat{\sigma}$ of the relative error. As we can see in Table 1, the mean error norm of the *GEODT* is very low (0.16[px]), when compared to the numerical simulation. The majority of this error can be explained by the corner detection noise stemming from the *GEODT* dataset. There is no detection noise to be accounted for in the numerical simulation, as the positions are directly computed. The comparisons with the *real* data generally yield a higher error for the *GEODT*, which holds as well for the *numerical* condition. Again, a lot of this error can be explained by the initial calibration (intrinsics and offset vector), which already yields a reprojection error of 0.52[px]. This has to be considered in addition to the corner detection noise on the *real* as well as on the *real GEODT* data. This leaves us only with a residual $\ll 1$ [px] in the mean error norms which can be caused by the *numerical* and *GEODT* models.

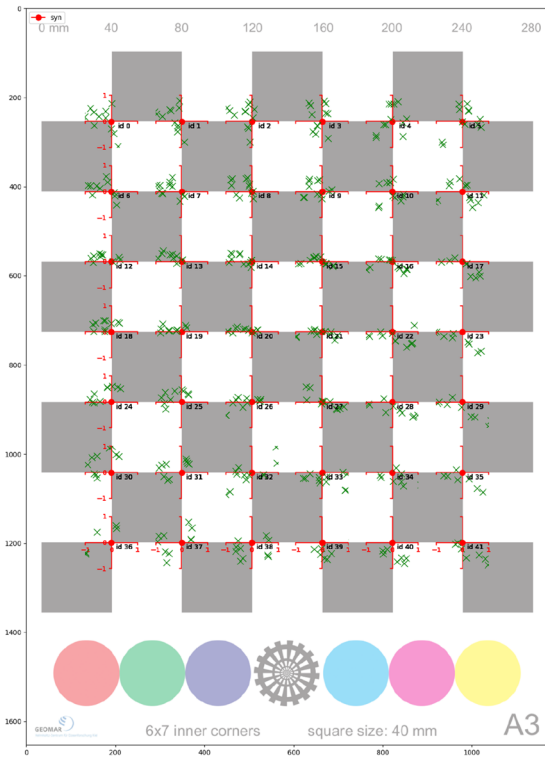
See Fig. 8 for an overview of the distribution of the relative error as well as the *real* dataset used for verification. As the test images cover a lot of different poses, we can expect a very close simulation of the reality by the *GEODT*.

5 Dataset

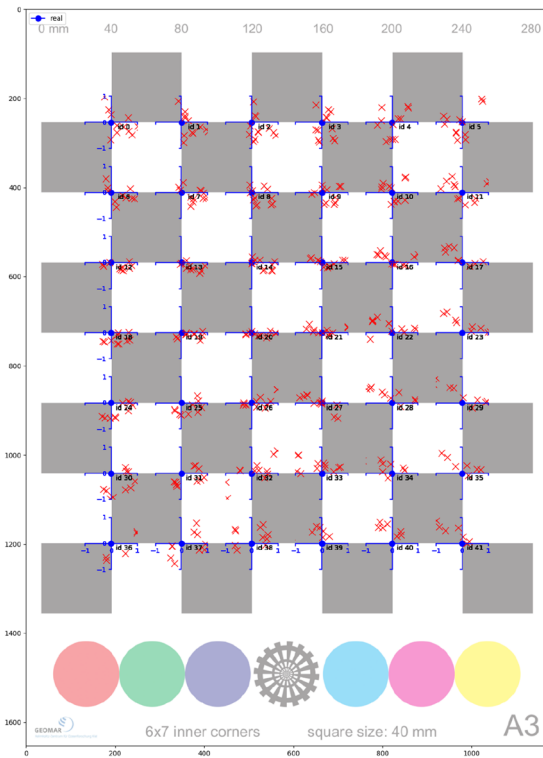
Our dataset comprises a set of rendered images as well as the Blender model of the tank which can be used to generate imagery with custom settings. For all rendered data, the corresponding YAML-config files are provided in the supplementary material, to allow for an easy extension.



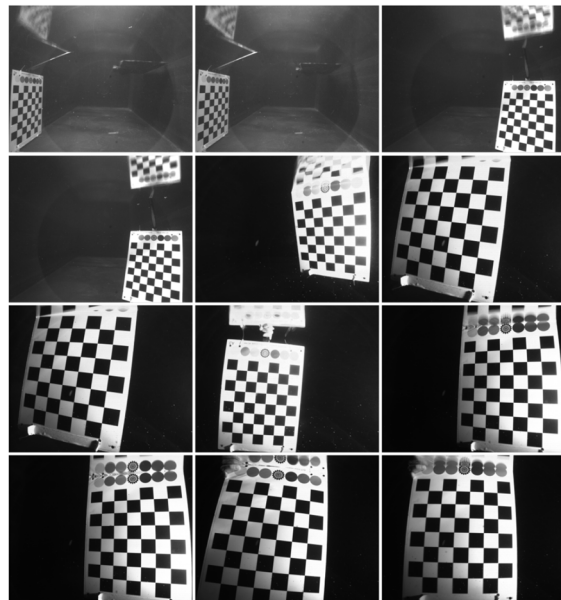
(a) real VS GEODT



(b) numeric VS GEODT



(c) real VS numeric



(d) Real Images used for evaluation

5.1 Pre-rendered Dataset

We rendered a dataset with 4096 samples per pixel (spp), not using any denoising steps to maintain the physical soundness of the images. In Fig. 9, example images are shown in the conditions full, half, no water in each row. Specifically, Fig. 9a–c show example images for the dome port using the

A3 calibration board and the board poses and the camera offset-vector \mathbf{v}_o extracted from the real tank data. Figure 9d–f use the same poses mirrored to the other side of the tank and shown through the flat port, whose camera is centered and has a distance of 2[cm] to it. In Fig. 9g–i an example pose from a set of 20 random poses using the calibration cube (see Fig. 3, left) is synthesized using the dome port. Finally,

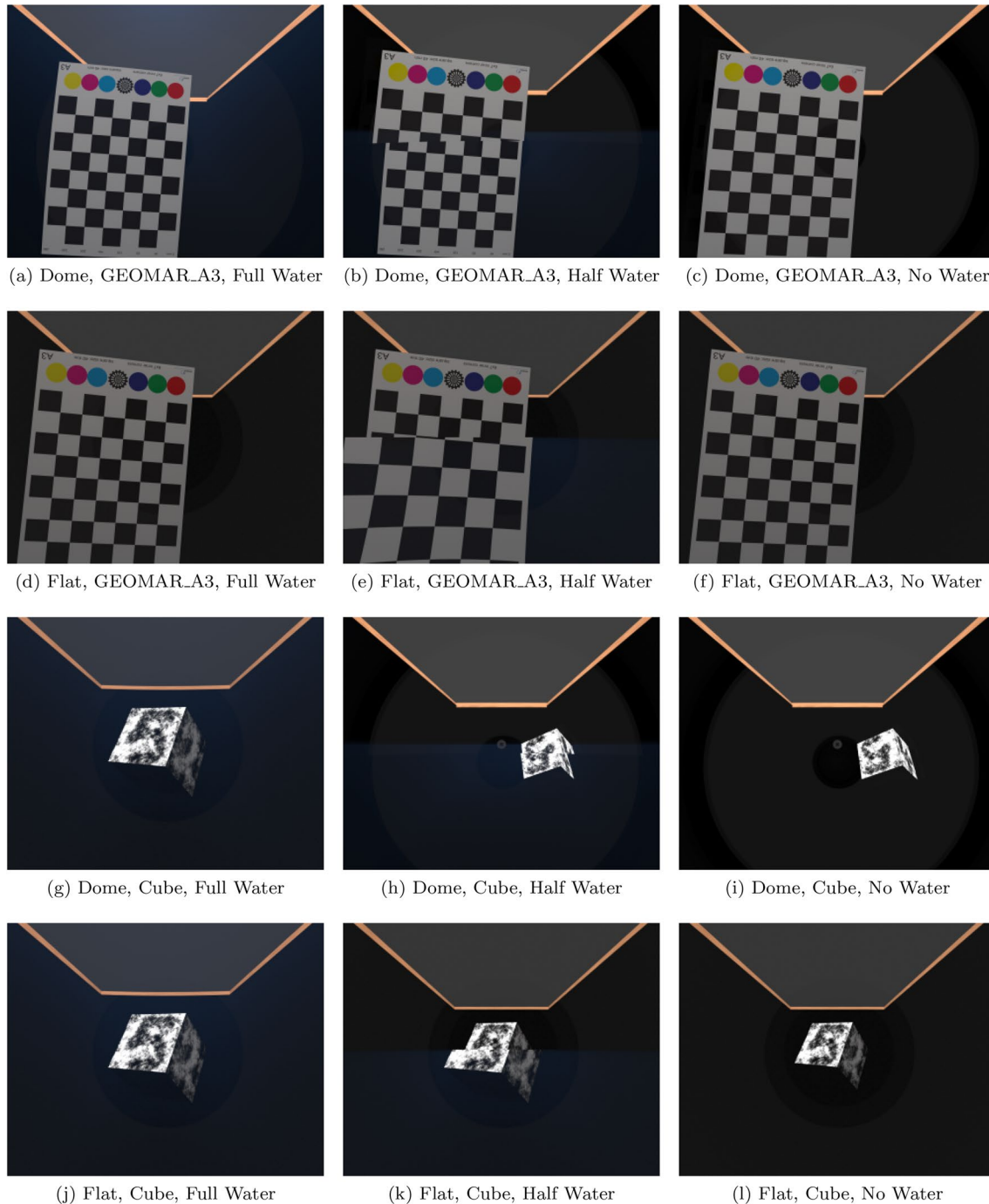


Fig. 9 Examples from the pre-rendered datasets using dome and flat-port as well as the conditions full-water (left), half-water (middle), and no-water (right). The both upper rows show the A3 calibration target, while the two lower rows show the calibration cube with random dot pattern

an example pose from the same set is shown in Fig. 9j–l through the flat port. In both latter conditions, we use the same camera settings as in the former ones.

5.2 Dataset Sampling

To generate more data, we provide the Blender file with code for automated generation of more images. i.e., if different poses, IORs or camera-settings are desired. Since the main challenge is to model the optical ports and interfaces, it should be easily possible to add more objects, textures or even whole scenes into the environment as needed. If desired, a seawater index of refraction can also be computed, based on temperature, pressure, salinity, density, and wavelength (Millard and Seaver 1990).

5.3 YAML Interface for Blender

For easy dataset generation, we implemented an interface, where the main parameters for dataset generation can be conveniently defined from the outside in a simple YAML file. Please see supplementary material (as indicated in the abstract) for examples.

6 Conclusion

In this paper, we introduced an optical digital twin of an underwater photogrammetry setting by modeling a real water lab tank and underwater cameras with the most common interfaces. Its main purpose is to facilitate further development of refractive photogrammetric algorithms and virtual verification experiments. We provide dome and flat ports with realistic (but still adjustable) optical properties as well as a convenient way to set the water properties in the virtual environment. We have shown by comparison to real camera images and to numerical forward projection of 3D coordinates that the refraction effects are properly simulated. Taking into consideration the errors caused by camera, dome offset calibration as well as corner detection, we can safely assume a geometrical modeling error $\ll 0.1$ [px]. This customizable basic tool box is easy to use for training, testing and verification in other multi-media refraction scenarios or environments.

6.1 Limitations and Future Work

As of now, the model does not account for diffraction (Radziszewski et al. 2009) or depth of field effects. In the future, a radiometric calibration, which is also influenced by refraction effects, would also be desirable to further enable the development, tuning, and testing of color-restoration

algorithms like e.g., (Akkaynak and Treibitz 2019; Nakath et al. 2021).

Acknowledgements This publication has been funded by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) Projektnummer 396311425, through the Emmy Noether Programme. We are also grateful for support from the Chinese Scholarship Council (CSC) for M. She (202006050015).

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Agrafiotis P, Karantzas K, Georgopoulos A, Skarlatos D (2021) Learning from synthetic data: Enhancing refraction correction accuracy for airborne image-based bathymetric mapping of shallow coastal waters. *PFG-J Photogram Remote Sens Geoinf Sci* 2:1–19
- Agrawal A, Taguchi Y, Ramalingam S (2010) Analytical forward projection for axial non-central dioptric and catadioptric cameras. In: *European Conference on Computer Vision*, pp. 129–143. Springer
- Akkaynak D, Treibitz T (2019) Sea-thru: A method for removing water from underwater images. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1682–1691
- Akkaynak D, Treibitz T, Shlesinger T, Loya Y, Tamir R, Iluz D (2017) What is the space of attenuation coefficients in underwater computer vision? In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 568–577. IEEE
- Community BO (2018) Blender - a 3D modelling and rendering package. Blender Foundation, Stichting Blender Foundation, Amsterdam <http://www.blender.org>
- Cozman F, Krotkov E (1997) Depth from scattering. In: *Proceedings of IEEE computer society conference on computer vision and pattern recognition*, pp. 801–806. IEEE
- Drap P (2012) Underwater photogrammetry for archaeology. In: D.C. da Silva (ed.) *Special Applications of Photogrammetry*, chap. 6. IntechOpen, Rijeka. <https://doi.org/10.5772/33999>
- Duda A, Gaudig C (2016) Refractive forward projection for underwater flat port cameras. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2022–2027. IEEE
- Eakins B, Sharman G (2012) Hypsographic curve of earth's surface from etopo1. NOAA National Geophysical Data Center, Boulder, CO 5
- Fong J, Wrenninge M, Kulla C, Habel R (2017) Production volume rendering: Siggraph 2017 course. In: *ACM SIGGRAPH 2017 Courses* 1–79
- Fryer JG, Fraser CS (1986) On the calibration of underwater cameras. *Photogram Rec* 12:73–85

- Glassner AS (1989) An introduction to ray tracing. Elsevier, Amsterdam
- Harvey ES, Shortis MR (1998) Calibration stability of an underwater stereo-video system : Implications for measurement accuracy and precision. *Mar Technol Soc J* 32:3–17
- Henye L, Greenstein JL (1941) Diffuse radiation in the galaxy. *Astrophys J* 93:70–83
- Hu X, Lauze F, Pedersen KS, Melou J (2021) Absolute and relative pose estimation in refractive multi view. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2569–2578
- Jaffe JS (1990) Computer modeling and the design of optimal underwater imaging systems. *IEEE J Oceanic Eng* 15(2):101–111
- Jaffe JS, Moore KD, McLean J, Strand MP (2001) Underwater optical imaging: Status and prospects. *Oceanography* 14:2
- Jordt A (2014) Underwater 3d reconstruction based on physical models for refraction and underwater light propagation. Ph.D. thesis, Christian-Albrechts-Universität zu Kiel, Germany
- Jordt A, Koch R (2011) Fast tracking of deformable objects in depth and colour video. In: McKenna S, Hoey J, Trucco M (eds.) Proceedings of the British Machine Vision Conference, BMVC 2011. British Machine Vision Association
- Jordt A, Köser K, Koch R (2016) Refractive 3d reconstruction on underwater images. *Methods Oceanogr* 15–16:90–113. <https://doi.org/10.1016/j.mio.2016.03.001>
- Jordt-Sedlazeck A, Koch R (2012) Refractive calibration of underwater cameras. In: European conference on computer vision, pp. 846–859. Springer
- Kahmen O, Rofallski R, Conen N, Luhmann T (2019) On scale definition within calibration of multi-camera systems in multimedia photogrammetry. *Remote Sensing & Spatial Information Sciences, International Archives of the Photogrammetry*
- Kajiya JT (1986) The rendering equation. In: Proceedings of the 13th annual conference on Computer graphics and interactive techniques, pp. 143–150
- Kirk JT (1994) Light and photosynthesis in aquatic ecosystems. Cambridge University Press, Cambridge
- Koenig N, Howard A (2004) Design and use paradigms for gazebo, an open-source multi-robot simulator. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No. 04CH37566), vol. 3, pp. 2149–2154. IEEE
- Kotowski R (1988) Phototriangulation in multi-media photogrammetry. *Int'l Archives of Photogrammetry and Remote Sensing XXVII*
- Kunz C, Singh H (2008) Hemispherical refraction and camera calibration in underwater vision. In: OCEANS 2008, pp. 1–7. IEEE
- Li B, Heng L, Koser K, Pollefeys M (2013) A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1301–1307. IEEE
- Manhães MMM, Scherer SA, Voss M, Douat LR, Rauschenbach T (2016) Uuv simulator: A gazebo-based package for underwater intervention and multi-robot simulation. In: OCEANS 2016 MTS/IEEE Monterey, pp. 1–8. IEEE
- McGlamery BL (1975) Computer analysis and simulation of underwater camera system performance. Tech. rep., Visibility Laboratory, Scripps Institution of Oceanography, University of California in San Diego
- Menna F, Nocerino E, Fassi F, Remondino F (2016) Geometric and optic characterization of a hemispherical dome port for underwater photogrammetry. *Sensors* 16(1). <http://www.mdpi.com/1424-8220/16/1/48>
- Menna F, Nocerino E, Remondino F (2017). Optical aberrations in underwater photogrammetry with flat and hemispherical dome ports <https://doi.org/10.1117/12.2270765>
- Menna F, Nocerino E, Ural S, Gruen A (2020) Mitigating image residuals systematic patterns in underwater photogrammetry. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLIII-B2-2020: 977–984*. <https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-977-2020>
- Millard R, Seaver G (1990) An index of refraction algorithm for seawater over temperature, pressure, salinity, density, and wavelength. *Deep Sea Res Part A Oceanogr Res Pap* 37(12):1909–1926
- Mobley C, Boss E, Roesler C (2021) Ocean optics web book. URL <http://www.oceanopticsbook.info>
- Moore EJ (1976) Underwater photogrammetry. *Photogram Rec* 8(48):748–763. <https://doi.org/10.1111/j.1477-9730.1976.tb00852.x>
- Mulrow C, Maas HG (2014) A universal approach for geometric modelling in underwater stereo image processing. In: 2014 ICPR Workshop on Computer Vision for Analysis of Underwater Imagery, pp. 49–56. IEEE
- Nakath D, She M, Song Y, Koser K (2021) In-situ joint light and medium estimation for underwater color restoration. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3731–3740
- Nayar SK, Narasimhan SG (1999) Vision in bad weather. In: Proceedings of the Seventh IEEE International Conference on Computer Vision 2: 820–827 vol.2. <https://doi.org/10.1109/ICCV.1999.790306>
- Nocerino E, Menna F, Gruen A (2021) Bundle adjustment with polynomial point-to-camera distance dependent corrections for underwater photogrammetry. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLIII-B2-2021: 673–679*. <https://doi.org/10.5194/isprs-archives-XLIII-B2-2021-673-2021>
- Novák J, Georgiev I, Hanika J, Krivánek J, Jarosz W (2018) Monte carlo methods for physically based volume rendering. In: SIGGRAPH Courses, pp. 14–1
- Parker SG, Bigler J, Dietrich A, Friedrich H, Hoberock J, Luebke D, McAllister D, McGuire M, Morley K, Robison A, Stich M (2010) Optix: A general purpose ray tracing engine. *ACM Trans Gr* 2:2
- Pharr M, Jakob W, Humphreys G (2016) Physically based rendering: From theory to implementation. Morgan Kaufmann
- Prats M, Perez J, Fernández JJ, Sanz PJ (2012) An open source tool for simulation and supervision of underwater intervention missions. In: 2012 IEEE/RSJ international conference on Intelligent Robots and Systems, pp. 2577–2582. IEEE
- Preisendorfer R (1964) Physical aspect of light in the sea. *Univ Hawaii Press Honolulu Hawaii* 51:60
- Quigley M, Gerkey B, Conley K, Faust J, Foote T, Leibs J, Berger E, Wheeler R, Ng A (2009) Ros: an open-source robot operating system. In: Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics. Kobe, Japan
- Radziszewski M, Boryczko K, Alda W (2009) An improved technique for full spectral rendering. *J WSCG* 17:9–16
- Sedlazeck A, Koch R (2011) Simulating deep sea underwater images using physical models for light attenuation, scattering, and refraction. In: Eisert P, Hornegger J, Polthier K (eds) VMV 2011: Vision, Modeling & Visualization, 978-3-905673-85-2. Eurographics Association, Berlin, Germany, pp 49–56
- She M, Nakath D, Song Y, Köser K (2022) Refractive geometry for underwater domes. *ISPRS J Photogramm Remote Sens* 183:525–540. <https://doi.org/10.1016/j.isprsjprs.2021.11.006>
- She M, Song Y, Mohrmann J, Köser K (2019) Adjustment and calibration of dome port camera systems for underwater vision. In: German Conference on Pattern Recognition, pp. 79–92. Springer
- Shmutter LB (1967) Orientation problems in two-media photogrammetry. *Photogrammetric Engineering* pp. 1421–1428
- Song Y, Nakath D, She M, Elibol F, Köser K (2021) Deep sea robotic imaging simulator. In: Del Bimbo A, Cucchiara R, Sclaroff S, Farinella GM, Mei T, Bertini M, Escalante HJ, Vezzani R (eds)

- Pattern Recognition. ICPR International Workshops and Challenges. Springer International Publishing, Cham, pp 375–389
- Treibitz T, Schechner YY, Singh H (2008) Flat refractive geometry. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR 2008, pp. 1–8
- Veach E (1998) Robust Monte Carlo methods for light transport simulation. Stanford University, Stanford
- Whitted T (1980) An improved illumination model for computer graphics. *Comm. ACM* 23(6):343–349
- Zwilmeyer PGO, Yip M, Teigen AL, Mester R, Stahl A (2021) The varos synthetic underwater data set: Towards realistic multi-sensor underwater data with ground truth. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3722–3730