Underwater 3D Capture using a Low-Cost Commercial Depth Camera

Sundara Tejaswi Digumarti
Disney Research Zurich, ETH Zurich
dtejaswi@mavt.ethz.ch

Aparna Taneja*
Disney Research Zurich

Amber Thomas
Walt Disney World
Amber.X.Thomas.-ND@disney.com

Gaurav Chaurasia
Disney Research Zurich
gaurav.chaurasia@disneyresearch.com

Roland Siegwart
ETH Zurich
rsiegwart@ethz.ch

Paul Beardsley
Disney Research Zurich
pab@disneyresearch.com

Abstract

This paper presents underwater 3D capture using a commercial depth camera. Previous underwater capture systems use ordinary cameras, and it is well-known that a calibration procedure is needed to handle refraction. The same is true for a depth camera being used underwater. We describe a calibration method that corrects the depth maps of refraction effects. Another challenge is that depth cameras use infrared light (IR) which is heavily attenuated in water. We demonstrate scanning is possible with commercial depth cameras for ranges up to 20 cm in water.

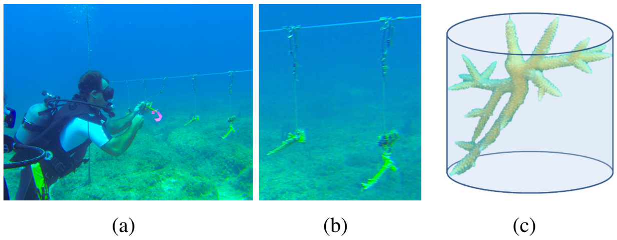
The motivation for using a depth camera under water is the same as in air – it provides dense depth data and higher quality 3D reconstruction than multi-view stereo. Underwater 3D capture is being increasingly used in marine biology and oceanology; our approach offers exciting prospects for such applications.

To the best of our knowledge, ours is the first approach that successfully demonstrates underwater 3D capture using low cost depth cameras like Intel RealSense. We describe a complete system, including protective housing for the depth camera which is suitable for handheld use by a diver. Our main contribution is an easy-to-use calibration method, which we evaluate on exemplar data as well as 3D reconstructions in a lab aquarium. We also present initial results of ocean deployment.

1. Introduction

Depth cameras have grown in importance since the introduction of the first Kinect [3], a general purpose low-cost technology alongside ordinary cameras. The underlying principle is to emit infrared (IR) light and capture the light reflected by scene objects. The two common approaches for using IR light to measure depth are time-of-flight (TOF) and structured light, wherein a structured pattern of IR light is projected on the surface and the distortion in reflected structure is used to calculate surface geometry.

The advantage of a depth camera is that it produces dense and reliable depth measurements, albeit over a limited range. While the Kinect was developed for gaming, it was also demonstrated for 3D scene capture [20]. Google Tango is a prototype mobile phone with depth camera that is targeted at 3D scene capture [5]. Our goal is to use an underwater depth camera to capture 3D models of submerged objects like marine flora/fauna. Applications that make use of this data require centimeter to millimeter accuracy in order to capture minute changes in geometry. Our use of depth cameras is motivated by such high-accuracy requirements.



(a) (b) (c)

Figure 1. (a) The coral nursery, (b) close-up of suspended corals. and (c) the *ecological volume* is the volume of the enclosing elliptic cylinder for the coral.

The main motivation for our work comes from coral reef research being conducted at Disney's Castaway Cay and Great Abaco in the Bahamas. Pieces of coral that naturally break off a reef, through storm, wave activity or impact, are taken to a coral nursery where they are suspended from a frame as shown in Fig 1(b). Measurements of coral volume are taken at six month intervals and corals which show healthy growth are transplanted back to the reef. The current method for measuring coral size is to manually es-

[*] The presented work was done when this author was at Disney Research Zurich. The author is currently at Google, Zurich.

timate the *ecological volume* [17] i.e. the enclosing elliptic cylinder as shown in Fig 1(c). We seek to automate and improve the accuracy of volume measurement by capturing a 3D model of the coral using a depth sensor, and estimating its true volume. Keeping this application in mind, our target is to develop a cheap and compact solution that enables handheld scanning of marine life for divers.

There are two challenges in underwater depth sensing – firstly, IR light used by depth cameras is heavily attenuated by water. We demonstrate a system that is capable of capturing underwater surfaces within a range of 20 *cm*, which is compatible with our application. Secondly, the images or depth scans captured by any camera underwater do not follow the principles of perspective projection [14] because of refraction at the transparent interface of the housing of the camera. A calibration procedure is required to account for the refraction. There are existing methods for calibrating ordinary cameras for underwater use [24], but to the best of our knowledge, there is no analogue for underwater calibration for the new generation of commercial depth cameras. We present a model for refraction in our setup and an easy-to-use calibration method which requires only a single depth image of a plane. Our approach extends straightforwardly to multiple images if needed for improved conditioning. We have tested our calibration method on two different depth cameras – Intel RealSense [6] as an example of a structured light camera, and Creative Senz3D [4] as an example of a TOF camera.

In the remainder of the paper, we discuss previous work in Sec. 2. We describe our scanning hardware and housing for the cameras in Sec. 3 and the refraction model along with the calibration algorithm in Sec. 4. We present our results on lab as well as ocean environments in Sec. 5 for Intel RealSense.

We have experimented with both structured light and TOF depth cameras; we focus on structured light cameras like Intel RealSense. We describe how to adapt our approach to TOF cameras like Creative Senz3D in Appendix A.

2. Related Work

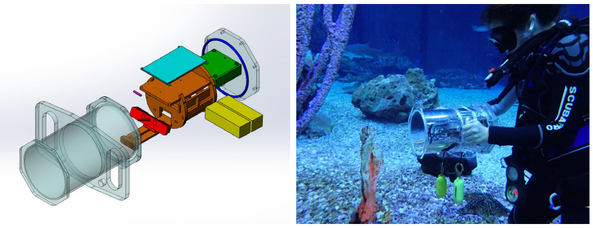
The success of terrestrial 3D reconstruction for visualizing natural and man-made environments has spurred a similar movement in marine science. The XL Catlin Seaview Survey [13] is a scientific initiative for capturing imagery for the world's coral reefs. The Computer Vision Coral Ecology Project [9] is focused on classification and automatic image annotation of coral reef images, although not concerned with 3D reconstruction. The latter project is also associated with CoralNet [1], a Citizen Science website that allows users to upload and label coral images. Hydrous [2] is a scientist-artist initiative in which 3D models of corals are created from images using AutoDesk. These conservation and visualization related projects require high quality 3D reconstructions of ocean bed, marine life etc.

The most prominent work on underwater 3D reconstruction include monocular [25] and stereo vision [8, 7]. Vision based approaches have difficulty generating dense and accurate 3D point clouds for complex geometry; depth cameras often give much better quality, at least for terrestrial scenes. Our intuition is that if we overcome the complexity of refraction and IR attenuation, using underwater depth cameras can significantly improve reconstruction quality over stereo vision.

Some previous approaches project structured light [19, 10] or laser [18] patterns on underwater surfaces and compute the surface geometry using the reflected patterns captured using ordinary cameras. Their results are better than stereo vision, but their use of visible light makes them impractical for our purpose. Such approaches can only be used in lab conditions where the ambient lighting can be controlled. Dancu et al [12] used Kinect to reconstruct an underwater surface. However, they had to hold the Kinect outside water because it is only effective for distances greater than 50 *cm* and IR attenuation under water renders the Kinect ineffective beyond 20 *cm* or so. All these approaches account for some of the refraction-borne complexity, but they are not usable for actual underwater scanning *in the wild*.

Our target is to design a system that is effective within the IR underwater attenuation range of circa 20 *cm*. To this end, we have experimented with Intel RealSense and Creative Senz3D cameras. Instead of designing a custom hardware setup, we use off-the-shelf depth cameras and account for attenuation and refraction issues in software. The end result is an affordable device that can be used by a diver for handheld scanning of the ocean bed or coral reefs.

3. Capture device



(a) (b)

Figure 2. Capture device. (a) Exploded schematic of the hardware. The red oblong is the Intel RealSense and the cyan rectangle on the top-right is the screen. (b) The device being tested underwater.

Our capture hardware is a self-contained device (see Fig. 2) in a waterproof housing, suitable for handheld use by a diver. The housing is made of acrylic, sealed with silicone. The rear cover is removable for accessing the interior when outside of the water. The system consists of:

- Intel RealSense depth camera,
- Intel NUC mini-PC,
- Waveshare 7” screen,
- SSD for captured data,
- LiPo batteries, and
- magnetic switches.

The magnetic switches allow a diver to activate or deactivate a small number of functions, such as ‘start/stop recording’, from outside the housing. The screen allows the diver to see the current view of the depth camera. The Intel RealSense generates a stream of both RGB images and depth images, which are recorded to the SSD. The 3D reconstruction is performed offline (see Sec. 4); we plan to attempt online 3D reconstruction in the future. The housing has an external attachment point for ballast. The entire assembly measures $250\ cm \times 18\ cm$.

4. Calibration of a Depth Sensor for Underwater Operation

The pinhole camera model is the *de facto* standard in computer vision applications. This model is not valid for underwater captures, as light rays are refracted at multiple interfaces. An underwater camera requires a transparent protective housing; light is refracted in the housing material as well as water. Previous approaches described solutions in the context of color cameras [24]. Some approaches circumvent this problem by placing the camera at the center of a hemispherical housing; light rays from/to the camera’s pinhole are orthogonal to the housing and water interfaces, thereby undergoing no refraction. It is physically challenging to construct such a housing for the Intel RealSense because it has two pinholes, the projector and the receiver (sensor). Instead of relying on custom fabricated hardware, we present a generic solution to this problem by accounting for refraction in our mathematical model.

The Intel RealSense is an infrared (IR) structured light camera that consists of a projector and a receiver. The structured pattern of IR rays emitted by the projector undergoes refraction at the air-housing interface and then at the housing-water interface as shown in Fig. 3. The reflected rays undergo the same refractions in the reverse order. We use this 4-stage refraction path instead of a pinhole camera model for the RealSense (see Fig 4). The parameters required to describe the complete path of a IR ray are:

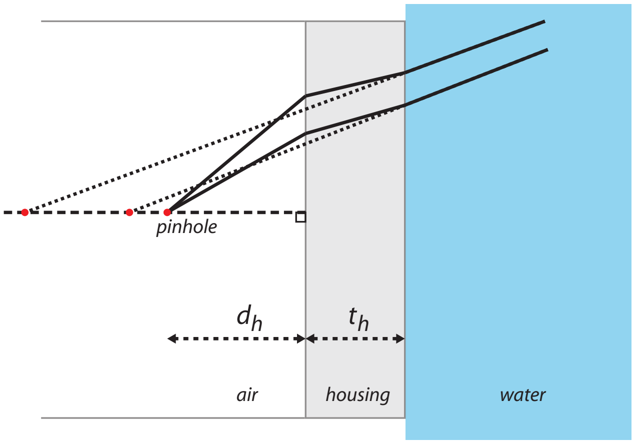


Figure 3. Refractions in an underwater camera. A ray through the pinhole is refracted at the air-housing and the housing-water interface. Different rays in water when traced back into the air do not meet at the pinhole.

$\mathbf{n}$	normal to the interface
d_h	distance from the pinhole to the housing
t_h	thickness of the housing
b	baseline between projector and receiver

In the following sections, we review the mathematical details of ray traversal in our model and then describe the calibration method to obtain the above parameters.

4.1. Refraction model

Let the pinhole of the receiver be $\mathbf{O}_\mathrm{r}$ with center of the projector at $\mathbf{O}_\mathrm{p}$ (see Fig 4). We assume that the receiver of the depth camera is already calibrated in air and its intrinsic matrix is known. We model the projector as a virtual camera with the same intrinsic matrix as the receiver. Furthermore, the image planes of the projector and receiver are parallel with no vertical disparity.

The depth map captured by the receiver contains one depth measurement d_meas for each pixel $(u_\mathrm{r}, v_\mathrm{r})$. The depth measurement is the distance along the principal axis of the receiver. Using the intrinsic matrix of the receiver, every pixel $(u_\mathrm{r},\ v_\mathrm{r})$ can be back-projected to compute the direction of the incident ray in air $\mathbf{X}_a^\mathrm{r}$.

This ray intersects the air-housing interface at $\mathbf{x}_i^\mathrm{r}$. The angle of incidence θ_a and the angle of refraction θ_h (with respect to the normal $\mathbf{n}$) at the interface can be computed as:

$$\theta_a = \arccos\left(\frac{\mathbf{n}\cdot\mathbf{X}_a^\mathrm{r}}{\|\mathbf{X}_a^\mathrm{r}\|}\right) \quad (1)$$

$$\theta_h = \arcsin\left(\frac{\eta_a \sin\theta_a}{\eta_h}\right) \quad (2)$$

where η_a and η_h are the refractive indices of air and the

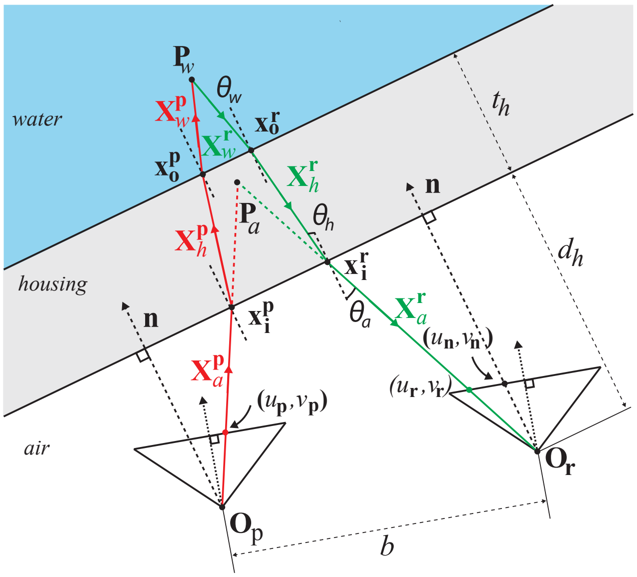

water
housing
air
$\mathbf{P}_w$
θ_w
$\mathbf{X}_w^\mathrm{p}$
$\mathbf{X}_w^\mathrm{r}$
$\mathbf{x}_\mathrm{o}^\mathrm{r}$
$\mathbf{x}_\mathrm{o}^\mathrm{p}$
t_h
$\mathbf{X}_h^\mathrm{r}$
$\mathbf{n}$
$\mathbf{P}_a$
θ_h
$\mathbf{X}_h^\mathrm{p}$
d_h
$\mathbf{x}_\mathrm{i}^\mathrm{r}$
$\mathbf{n}$
$\mathbf{x}_\mathrm{i}^\mathrm{p}$
θ_a
$\mathbf{X}_a^\mathrm{r}$
$(u_\mathrm{n}, v_\mathrm{n})$
$\mathbf{X}_a^\mathrm{p}$
$(u_\mathrm{r}, v_\mathrm{r})$
$(u_\mathrm{p}, v_\mathrm{p})$
$\mathbf{O}_\mathrm{r}$
$\mathbf{O}_\mathrm{p}$
b

Figure 4. Refraction model showing the complete path of a structured IR ray in an Intel RealSense along with all model parameters. The ray starting from the projector is refracted multiple times before reaching the receiver.

housing material respectively.

The direction of the refracted ray inside the housing material can then be computed by rotating the ray $\mathbf{X}_a^\mathrm{r}$ by an angle $\theta_a^h = \theta_h - \theta_a$ about an axis $\mathbf{n}_a^h$ perpendicular to both the interface normal and the incident ray:

$$\mathbf{n}_a^h = \frac{\mathbf{n} \times \mathbf{X}_a^\mathrm{r}}{\|\mathbf{X}_a^\mathrm{r}\| \sin\theta_a} \qquad (3)$$

The point $\mathbf{x}_i^\mathrm{r}$ can be thus computed as:

$$\mathbf{x}_i^\mathrm{r} = \mathbf{O}_\mathrm{r} + \lambda_h \frac{\mathbf{X}_a^\mathrm{r}}{\|\mathbf{X}_a^\mathrm{r}\|}, \qquad \lambda_h = \frac{d_h}{\cos\theta_a}, \qquad (4)$$

where d_h is the distance from the pinhole of the receiver to the housing interface.

The ray inside the housing material $\mathbf{X}_h^\mathrm{r}$ intersects the housing-water interface at $\mathbf{x}_o^\mathrm{r}$ and undergoes another refraction. The point $\mathbf{x}_o^\mathrm{r}$ can be computed as:

$$\mathbf{x}_o^\mathrm{r} = \mathbf{x}_i^\mathrm{r} + \lambda_w \frac{\mathbf{X}_h^\mathrm{r}}{\|\mathbf{X}_h^\mathrm{r}\|}, \qquad \lambda_w = \frac{d_h + t_h - \mathbf{n} \cdot \mathbf{X}_i^\mathrm{r}}{\cos\theta_h} \qquad (5)$$

The ray direction in water $\mathbf{X}_w^\mathrm{r}$ can be computed using Snell's law and the refractive index of water η_w as before. The true 3D point $\mathbf{x}_w$ lies on this ray $\mathbf{X}_w^\mathrm{r}$. This point is also the point of intersection of a ray from the projector $\mathbf{X}_w^\mathrm{p}$ with the ray $\mathbf{X}_w^\mathrm{r}$. Therefore, if we find the ray $\mathbf{X}_w^\mathrm{p}$ we can find the true 3D point. This ray can be found by using the working principle of a structured light camera. Structured light cameras project a sequence of light patterns or gray codes which is unique for each column. Assuming a row-aligned projector-receiver system, the depth of a 3D point can be computed as follows under no refractions:

$$d = \frac{f_\mathrm{x} b}{u_\mathrm{p} - u_\mathrm{r}} \qquad (6)$$

2D to 3D computation We have the depth measurement d_meas for each pixel $(u_\mathrm{r}, v_\mathrm{r})$ on the receiver. Under no refractions, substituting this in Eq 6 gives the column index, and hence the pixel $(u_\mathrm{c}, v_\mathrm{r})$ on the projector. In our above refraction model, we compute this intersection using a ray casting algorithm: we shoot rays from the projector's center along the column u_c, apply all refractions and compute the set of rays $S = \{\mathbf{X}_w^\mathrm{p}\}$ in water. We then select the ray that intersects the exactly one ray $\mathbf{X}_w^\mathrm{r}$ from the projector. The point of intersection gives the true 3D point $\mathbf{x}_w$.

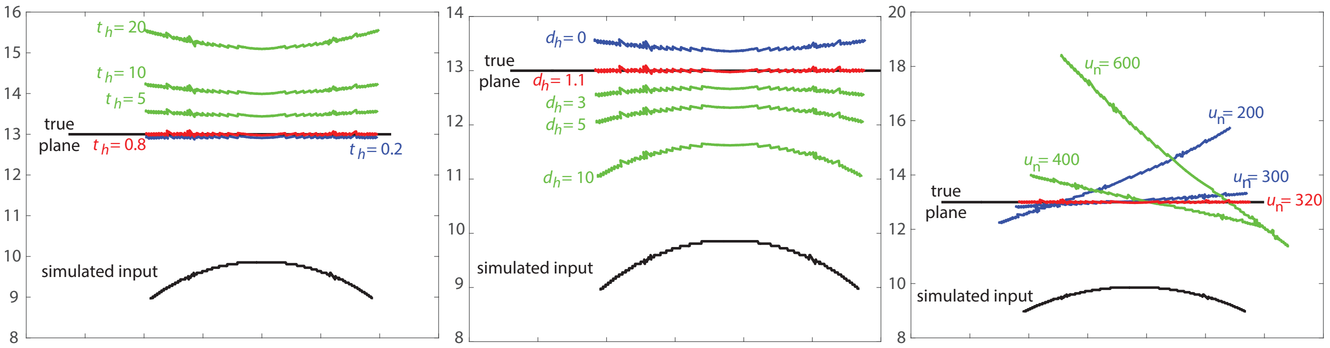
4.2. Calibration

We require a calibration step to measure the parameters of our refractive model from Sec. 4.1. The fundamental principle is that the depth measurements of a true planar surface should give coplanar 3D points after applying the 2D to 3D mapping of our refraction model as described in Sec. 4.1.

The didactic examples in Fig. 5 illustrate the effect of incorrect calibration. In all three examples, we assume a planar surface is being scanned, labeled “true plane”. Using a pinhole camera model without any refraction correction, the depth measurements from the sensors amount to a curved plane labeled “simulated input”. Our refraction model pushes this result towards the true 3D plane if the model parameters are estimated correctly. Each of these examples show computed 3D plane for different values of the model parameters. The legend on the left indicates the scale of curvature and displacement in the computed 3D result.

The above examples also show that the normal direction $\mathbf{n}$ has very limited impact on the planarity of the final result (see Fig 5(c)). This is only a consequence of the values of the refractive indices of the three media. We therefore fix $\mathbf{n}$ by construction instead of numerical computation which may become ill-conditioned.

We use an optimization to compute the values of the parameters. We first select initial estimates for the parameters (d_h, t_h, b). We then capture a single depth image of a 3D plane. We first compute the 3D points corresponding to depth measurements using the refractive model under the current set of parameter values. We then fit a single plane and compute the fitting error normalized by the area of the



(a) Effect of t_h

(b) Effect of d_h

(c) Effect of x coordinate of $\mathbf{n}$

Figure 5. Effect of incorrect values of parameters thickness t_h, distance d_h and normal $\mathbf{n}$. The simulated plane indicates the computed plane if no refraction modeling is used. Other surfaces indicate the curvature and displacement from the true 3D plane for erroneous values of the model parameters; legend on the left gives the scale of error. This shows the importance of accurate calibration.

projection of the 3D points on the plane. This gives the fitting error as a function of the current set of parameter values. We minimize this objective function using BOBYQA (Bound Optimization BY Quadratic Approximation) [22]. The values of parameters obtained are then used for all experiments.

4.3. 3D to 2D correspondences

In Sec. 4.1, we described the algorithm to convert a measurement on the receiver $(u_{\mathrm{r}}, v_{\mathrm{r}}, d_{\mathrm{meas}})$ to a 3D point $\mathbf{x}_w$. We now describe the algorithm for the reverse for completeness. This reverse 3D to 2D mapping is frequently required by depth map registration algorithms for aligning successive frames of a scan [11, 23, 15]. This reverse step is also non-trivial and a ray casting algorithm is needed.

We solve this by performing an exhaustive search over the entire range of pixels. We cast rays from every pixel of the depth image, following the refraction model, and finding the ray in water closest to the 3D point. The pixel corresponding to this ray is the projection of the 3D point onto the depth image plane. We use binary search by recursively dividing the image and subimages into 4 quadrants and performing the search within the quadrant which gives the closest ray.

Implementation We used the InfiniTAM library [23, 15] to generate the 3D reconstructions with a volumetric representation [21]. We modified the core projection routines of the library to incorporate our refraction model and implemented the ray casting algorithms in their CUDA kernels. Our 2D to 3D and 3D to 2D transformations involve ray casting; which is much slower than the original real time 3D scan in air. We currently require 1 second of computation per frame.

5. Results

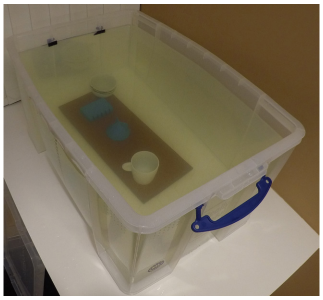
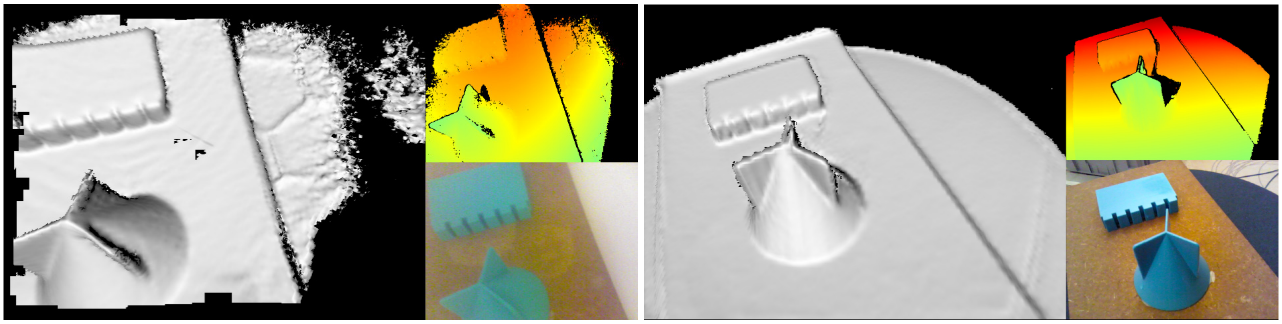


Figure 6. Experimental setup. The cups are not part of the scanned geometry; they have been placed to show the scale of the setup.

We tested our approach on some toy objects in an indoor tank as shown in Fig. 6. The scanning device is submerged into the tank and panned around the object to capture it from most sides. For this experiment, we connected a laptop to our scanning device instead of using the on-board SSD for debugging convenience.

Qualitative results Fig. 7(a) shows the quality of our depth maps directly from our implementation. For comparison, we also show the same test objects scanned in air using the same depth sensor without any refraction correction in Fig. 7(b). The visual quality of the depth maps clearly matches that of aerial scans. These figures are intermediate frames from two separate scans. As scanning was done by a hand held scanner, exactly same viewpoints were not available. Hence we show two viewpoints that are close.

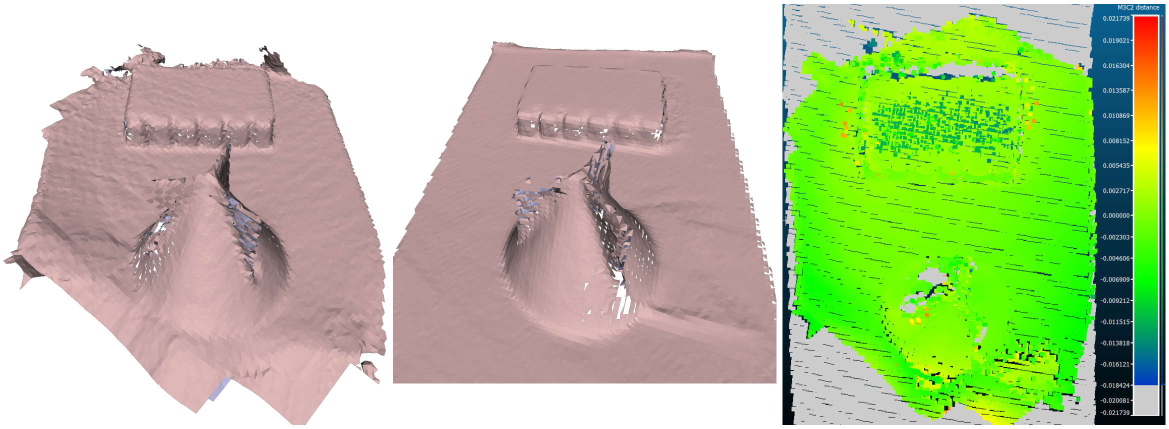
We extract a 3D mesh from the signed distance fields of the depth maps using simple Marching Cubes (see Fig. 8(a)). Better quality meshes can be reconstructed with more advanced techniques. This is however, not the main goal of the paper. We again compare our results to the mesh generated



(a) Depth map of underwater scan

(b) Depth map of aerial scan

Figure 7. Comparison of depth maps from (a) underwater scans using our approach and (b) aerial scans using state of the art techniques. These figures show the target scene from slightly different viewpoints. Our underwater results match the quality of aerial reconstruction using the same sensor.



(a) Mesh from underwater scan

(b) Mesh from aerial scan

(c) ICP registration error

Figure 8. Comparison of 3D meshes from underwater and aerial scans. (c) ICP registration error for meshes obtained from underwater and aerial scans shows that 93% of the mesh vertices have a registration error lower than 4 *mm*(green vertices).

from an aerial scan (see Fig. 8(b)). Clearly, our approach matches the quality of aerial scans. Note that the missing regions in the depth maps and 3D meshes are regions which fall in the shadow areas of the depth scanner.

Quantitative results We also show quantitative results of comparing aerial and underwater mesh. This is done to explicitly identify error arising from refraction effects. To this end, we register the two meshes using standard ICP (see Fig. 8(c)) and analyse the error in registration. This is similar to the evaluation method used by Koutsoudis et al[16]. The error heatmap shows that the two meshes are very well aligned: 93% of the mesh vertices have less than 4 *mm* registration error.

Experiments in the ocean Finally, we deployed our approach in a coral nursery for handheld scanning of stony corals by divers. Fig. 9 shows a color image and a depth map of the same coral for multiple examples. We did not encounter water turbidity as a problem in our initial ocean experiments as the water was clear at the coral nursery.

Turbulence of water was also not a problem. Stony corals showed no non-rigidity due to turbulence in a current of 2 *m*/*s*, which was encountered in our dive experiments. Turbulence can also potentially mix water between different depth. Different water layers have different temperatures, salinities and densities and this affects the refractive index. But these parameters are effectively constant up to depths of 50 *m* [26], while our coral reef experiments are confined to depths up to depths of 20 *m*. Therefore refractive index remains unaffected by turbulence at the depths where we tested our system.

6. Conclusion

We have presented a low-cost commercial depth camera and housing setup for underwater 3D scanning. Our goal was to develop a system that uses off-the-shelf hardware

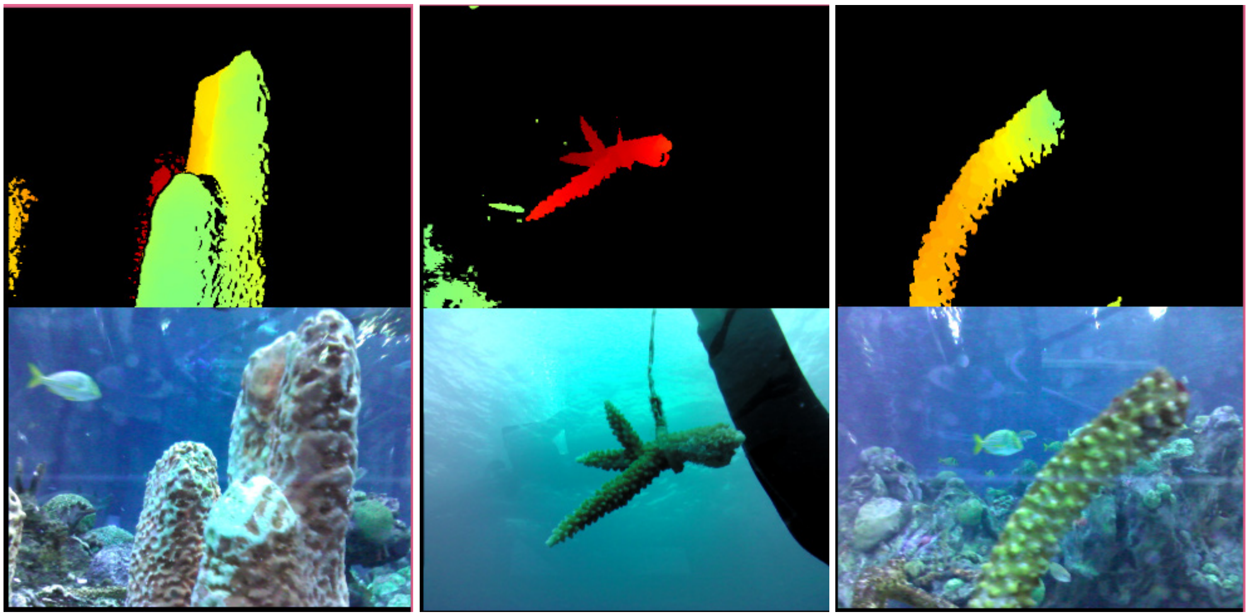


Figure 9. Depth maps (top) and images (bottom) from scans of multiple corals in a nursery.

for scanning. To this end, we created a waterproof housing for a self-contained scanning system including depth camera, screen, and onboard data storage. The main algorithmic contribution of the paper is an easy-to-use calibration method for a depth camera in water. We also modified KinectFusion to develop an algorithm that accounts for multiple refraction effects in our setup, and converts the scanned depth maps into a 3D mesh. Our results demonstrate that the calibration provides a correct model of refraction effects. We also presented initial results of deploying this system in a coral nursery for scanning of stony corals. Finally, we believe that our simple and light-weight system is an important step towards reliable solution for scanning of coral reefs.

A. Refraction Model of a Time of Flight Depth Camera for Underwater Operation

This section describes our refraction model for underwater operation of a depth camera that works on the principle of Time of Flight. As noted in Sec. 4, a pinhole camera model is not valid underwater because light rays get refracted at two interfaces, the air-housing interface and the housing-water interface.

We model the depth camera as an aligned projector-receiver pair but unlike the model of the structured light camera, here we assume that the centres of the projector and the receiver coincide with each other and shall refer to them together as the *camera*. This simplification implies that the projected ray and the received ray coincide with each other (see Fig 10). The model is thus parametrized by:

$\mathbf{n}$	normal to the interface
d_h	distance from the pinhole to the housing
t_h	thickness of the housing

It must be noted that this model cannot be obtained simply by equating the value of the baseline in the model of the structured light sensor to zero because the principles of working of the two cameras are different from one another.

The refraction model is shown in Fig 10. The pinhole of the camera is at $\mathbf{O}_\mathrm{r}$. Once again we assume that the intrinsics of the depth camera in air are known. The depth image captured by the camera contains one depth measurement d_m for every pixel $(u_\mathrm{r}, v_\mathrm{r})$.

A Time of Flight based depth camera measures depth, d_m by indirectly measuring the time difference t_m between the instants at which a ray is emitted and at which the ray is received back. This can be written as

$$d_m = (t_m/2) \cdot c_{air}, \tag{7}$$

where c_{air} is the speed of light in air. When the sensor is used underwater, we have to account for the lengths of the ray in three different media; air, housing material and water and the corresponding speed of light in each medium.

We split the time t_m into three segments namely t_a, t_h and t_h for the times the ray travels in air, housing material and water respectively. This is written as

$$t_m = t_a + t_h + t_w. \tag{8}$$

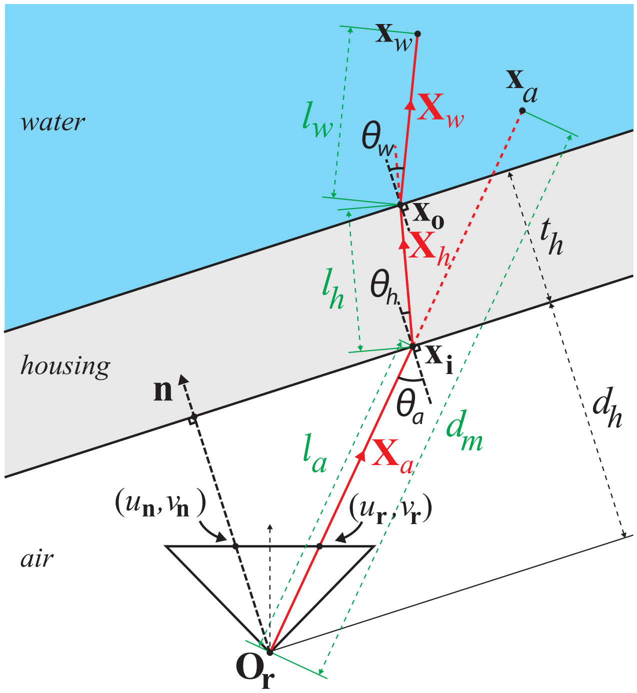


Figure 10. Refraction model showing the complete path of a structured IR ray in an Intel RealSense along with all model parameters. The ray starting from the projector is refracted multiple times before reaching the receiver.

Following the principles of refraction described in Sec 4 (Eq.s 1-5), for every pixel $(u_\mathrm{r}, v_\mathrm{r})$ of the depth image we can compute the following.

- $\mathbf{X}_a$, incident (or projected) ray in air,
- $\mathbf{x_i}$, intersection of $\mathbf{X}_a$ with the air-housing interface,
- $\mathbf{X}_h$, ray direction inside the housing,
- $\mathbf{x_o}$, intersection of $\mathbf{X}_h$ with the housing-water interface,
- $\mathbf{X}_w$, ray direction in water.

The true 3D point in water lies at $\mathbf{x}_w$ on the ray $\mathbf{X}_w$. The problem now is to find the distance l_w from $\mathbf{x_o}$ at which $\mathbf{x}_w$ lies. We can compute the distances l_a and l_h as

$$l_a = \|\mathbf{x}_i - \mathbf{O_r}\|, \qquad l_h = \|\mathbf{x_o} - \mathbf{x_i}\|. \tag{9}$$

The times for which the ray travels in air and inside the housing can then be computed as

$$t_a = \frac{2 \cdot l_a}{c_{air}}, \qquad t_h = \eta_h \frac{2 \cdot l_h}{c_{air}}, \tag{10}$$

where η_h is the refractive index of the housing material. Eqs 7,8 and 10 can be combined to get

$$t_w = \eta_w \frac{2 \cdot l_w}{c_{air}} = 2\left(\frac{d_m - l_a - \eta_h \cdot l_h}{c_{air}}\right), \tag{11}$$

where η_w is the refractive index of water. The true point $\mathbf{x}_w$ can then be computed as

$$\mathbf{x}_w = \mathbf{x_o} + l_w \frac{\mathbf{X}_w}{\|\mathbf{X}_w\|}, \qquad l_w = \frac{d_m - l_a - \eta_h \cdot l_h}{\eta_w}. \tag{12}$$

This gives us a mapping from a 2D depth image to 3D coordinates. The 3D to 2D mapping is done similar to the way it was done for the structured light based depth camera using ray casting.

References

[1] CoralNet: A web solution for coral reef analysis. http://coralnet.ucsd.edu.

[2] The hydrous. http://thehydro.us, 2010.

[3] Miscrosoft kinect for windows. https://dev.windows.com/en-us/kinect, 2012.

[4] Creative Senz3D. http://us.creative.com/p/web-cameras/creative-senz3d, 2015.

[5] Google Project Tango. https://www.google.com/atap/project-tango/, 2015.

[6] Intel RealSense. http://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html, 2015.

[7] P. Andono, E. Yuniarno, M. Hariadi, and V. Venus. 3D reconstruction of under water coral reef images using low cost multi-view cameras. In *International Conference on Multimedia Computing and Systems*, pages 803–808, May 2012.

[8] C. Beall, B. Lawrence, V. Ila, and F. Dellaert. 3D reconstruction of underwater structures. In *IROS*, pages 4418–4423, Oct 2010.

[9] O. Beijbom, P. J. Edmunds, D. I. Kline, B. G. Mitchell, and D. Kriegman. Automated annotation of coral reef survey images. In *CVPR*, 2012.

[10] F. Bruno, G. Bianco, M. Muzzupappa, S. Barone, and A. Razionale. Experimentation of structured light and stereo vision for underwater 3d reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(4):508 – 518, 2011.

[11] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *IEEE International Conference on Robotics and Automation*, pages 2724–2729 vol.3, Apr 1991.

[12] A. Dancu, M. Fourgeaud, Z. Franjcic, and R. Avetisyan. Underwater reconstruction using depth sensors. In *SIGGRAPH Asia 2014 Technical Briefs*, pages 2:1–2:4, 2014.

[13] M. Gonzalez-Rivero, P. Bongaerts, O. Beijbom, O. Pizarro, A. Friedman, A. Rodriguez-Ramirez, B. Upcroft, D. Laffoley, D. Kline, C. Bailhache, R. Vevers, and O. Hoegh-Guldberg. The Catlin Seaview Survey - kilometre-scale

seascape assessment, and monitoring of coral reef ecosystems. *Aquatic Conservation: Marine and Freshwater Ecosystems*, 24(S2):184–198, 2014.

[14] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[15] O. Kahler, V. A. Prisacariu, C. Y. Ren, X. Sun, P. H. S. Torr, and D. W. Murray. Very High Frame Rate Volumetric Integration of Depth Images on Mobile Device. *IEEE Trans. Vis. Comput. Graph. (Proceedings ISMAR*, 22(11), 2015.

[16] A. Koutsoudis, B. Vidmar, G. Ioannakis, F. Arnaoutoglou, G. Pavlidis, and C. Chamzas. Multi-image 3d reconstruction data evaluation. *Journal of Cultural Heritage*, 15(1):73–79, 2014.

[17] S. L, A. A., and R. B. Branch to colony trajectory in a modular organism: Pattern formation in the Indo-Pacific coral stylophora pistillata. *Developmental Dynamics*, 2006.

[18] M. Massot-Campos and G. Oliver-Codina. Underwater laser-based structured light system for one-shot 3D reconstruction. In *IEEE SENSORS*, pages 1138–1141, Nov 2014.

[19] S. Narasimhan and S. Nayar. Structured light methods for underwater imaging: light stripe scanning and photometric stereo. In *OCEANS*, pages 2610–2617 Vol. 3, 2005.

[20] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, pages 127–136, Oct 2011.

[21] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3D reconstruction at scale using voxel hashing. *ACM Trans. Graph.*, 32(6):169:1–169:11, Nov. 2013.

[22] M. J. Powell. The BOBYQA algorithm for bound constrained optimization without derivatives. Technical Report NA2009_06, Centre for Mathematical Sciences, University of Cambridge, Cambridge, England, 2009.

[23] V. A. Prisacariu, O. Kähler, M. M. Cheng, C. Y. Ren, J. Valentin, P. H. S. Torr, I. D. Reid, and D. W. Murray. A Framework for the Volumetric Integration of Depth Images. *ArXiv e-prints*, 2014.

[24] A. Sedlazeck and R. Koch. Perspective and non-perspective camera models in underwater imaging – overview and error analysis. In F. Dellaert, J.-M. Frahm, M. Pollefeys, L. Leal-Taixé, and B. Rosenhahn, editors, *Outdoor and Large-Scale Real-World Scene Analysis*, volume 7474 of *Lecture Notes in Computer Science*, pages 212–242. 2012.

[25] A. Sedlazeck, K. Koser, and R. Koch. 3D reconstruction based on underwater video from rov kiel 6000 considering underwater imaging conditions. In *OCEANS - EUROPE*, pages 1–10, May 2009.

[26] Wikipedia. Mixed layer — wikipedia, the free encyclopedia, 2015. [Online; accessed 25-January-2016].