IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 23, 2017, accepted July 18, 2017, date of publication August 2, 2017, date of current version August 29, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2733003

Underwater 3-D Scene Reconstruction Using Kinect v2 Based on Physical Models for Refraction and Time of Flight Correction

ATIF ANWER, SYED SAAD AZHAR ALI, (Senior Member, IEEE), AMJAD KHAN AND FABRICE MÉRIAUDEAU

Centre for Intelligent Signal and Imaging Research, Department of Electrical and Electronic Engineering, Universiti Teknologi PETRONAS, Perak 32610, Malaysia

Corresponding author: Fabrice Mériaudeau (fabrice.meriaudeau@u-bourgogne.fr)

This work was supported by Universiti Teknologi PETRONAS through Yayasan UTP under Grant 0153AA-E52.

ABSTRACT Commercial RGB-D cameras provide the possibility of fast, accurate, and cost-effective 3-D scanning solution in a single package. These economical depth cameras provide several advantages over conventional depth sensors, such as sonars and lidars, in specific usage scenarios. In this paper, we analyze the performance of Kinect v2 time-of-flight camera while operating fully submerged underwater in a customized waterproof housing. Camera calibration has been performed for Kinect's RGB and NIR cameras, and the effect of calibration on the generated 3-D mesh is discussed in detail. To overcome the effect of refraction of light due to the sensor housing and water, we propose a time-of-flight correction method and a fast, accurate and intuitive refraction correction method that can be applied to the acquired depth images, during 3-D mesh generation. Experimental results show that the Kinect v2 can acquire point cloud data up to 650 mm. The reconstruction results have been analyzed qualitatively and quantitatively, and confirm that the 3-D reconstruction of submerged objects at small distances is possible without the requirement of any external NIR light source. The proposed algorithms successfully generated 3-D mesh with a mean error of ±6 mm at a frame rate of nearly 10 fps. We acquired a large data set of RGB, IR and depth data from a submerged Kinect v2. The data set covers a large variety of objects scanned underwater and is publicly available for further use, along with the Kinect waterproof housing design and correction filter codes. The research is aimed toward small-scale research activities and economical solution for 3-D scanning underwater. Applications such as coral reef mapping and underwater SLAM in shallow waters for ROV's can be a viable application area that can benefit from results achieved.

INDEX TERMS Kinect v2, underwater, refraction correction, 3D reconstruction.

I. INTRODUCTION

3D scanning and scene reconstruction is a technique to digitize real-world objects and surfaces into 3-D models or meshes. These digitized models are then used for various applications such as generating physical prototypes, historical record preservation, maintenance and inspection, or for use in modern entertainment applications like Augmented Reality (AR) or Virtual Reality (VR). Particularly, for engineering and industrial applications, 3D scanning enables qualitative and quantitative analysis of equipment after withstanding operation in real environment. By comparison of the 3D scanned models to the original design intent, weathering effects on the equipment can be monitored. This is particularly true in harsher environments such as underwater, where changes such as erosion, rust and other deteriorating effects due to water are standard, and equipment must be monitored periodically. 3D scanning for underwater environment is being utilized much more since the last couple of decades due to significant advancement of 3D depth sensing using sensors technologies such as sonars and lidars as well as other modern imaging techniques. For marine applications, 3D scanning and scene reconstructions are being used extensively for ship hull inspection, mapping and restoring the coral reefs [1], scanning underwater terrain, exploring and mapping sunken ships and aircraft [2] just to name a few. It is also opening new avenues for underwater robotics by enabling real-time mapping and depth sensing for autonomous and semi-autonomous navigation in GPS-deprived environments.

In the field of 3D mapping and scene reconstruction, terrestrial applications and robotics have seen extensive work over the last three to four decades. 3D reconstruction in an underwater environment, however, is a much more challenging task due to the requirement of expensive, specialized equipment or services, as well as the challenges faced due to the harsh environment and properties of water as an imaging medium. Underwater RGB cameras offer methods for real-time depth sensing and 3D reconstruction in the harsh underwater environment, but are heavily dependent on the presence of ambient or artificial lighting and have limited range due to the properties of light propagation in water. The prohibitive cost of acquiring up-to-date data through traditional methods such as airborne LiDARS, advanced ship-based Sonars or static 3D scanning sensors limit the work of many researchers and organizations with a limited budget and access to specialized equipment. There is a significant gap in the provision of a cost-effective, real-time, underwater 3D scene reconstruction methods for underwater structures and objects.

The remainder of the paper is organized as follows. Section 2 discusses the research work that has been done on depth cameras underwater. Section 3 discusses the time of flight and refraction correction techniques proposed along with pseudocode. Section 4 presents results for Kinect$^{\text{ToF}}$ performance in underwater and 3D mesh reconstruction before and after corrections, followed by conclusions in Section 5.

II. RELATED WORK

With the advent of commercial RGB-D sensors, the possibility of fast, accurate and low-cost 3D sensing has become available. RGB-D sensors are a multi-camera system, with an RGB camera and a depth camera combined in one package. The depth camera can use any technique for depth sensing such as structured light or time of flight. Sensors like Kinect v1 (denoted as Kinect$^{\text{SL}}$ from here onwards) and Kinect v2 (denoted as Kinect$^{\text{ToF}}$ from here onwards), Intel Realsense, Asus Xtion and others provide the ability for real-time 3D scene reconstruction. These reconstructed scenes can be used for a multitude of purposes, such as visualizing and understanding the condition of the surrounding environment or advanced 3D robot navigation in unstructured environments, indoors as well as outdoors [3]. For Kinect$^{\text{SL}}$ as well as Kinect$^{\text{ToF}}$, several researchers have done an extensive metrological characterization of the sensors. Kinect$^{\text{ToF}}$ works on the phase difference between the transmitted and received NIR signals. The phase shift is a function of the time difference between the time of the transmitted and received signal. Detailed working and mathematical modeling of the phase difference methodology of Kinect$^{\text{ToF}}$ sensor has been covered by Jaremo *et al.* [4]. Effects of temperature and materials on the depth sensing performance have been discussed by Corti *et al.* [5]. Effect of different variables, such as varying illumination, ambient temperature, various calibration parameters including IR sensor and RGB camera calibration was in detail by the authors in [6]–[13]. Hertzberg *et al.* [14] provide calibration of the sensor along with issues about multiple sensors with overlapping field-of-views. Careful error analysis of the depth measurements from Kinect has been made by Khoshelham [15]. Recently, Huimin *et al.* [16] have proposed an improvement in depth map accuracy using underwater channels prior de-hazing model and inpainting.

It is interesting to note that even though, the release of the first Kinect was in 2011, limited work has been done in using Kinect, or any other commercial RGB-D sensor, in underwater environment. Tsui *et al.* [17] showed experimental results for depth image acquisition and point cloud generation using the Kinect$^{\text{SL}}$ and Softkinect DS311 time of flight camera. Both cameras operate between the 800-850 nm NIR region and they successfully acquired depth images underwater up to 0.9 m but with significant noise. Dancu *et al.* [18] demonstrated depth images acquired from Kinect$^{\text{SL}}$ for surface below the water level, however, the sensor was kept approximately 0.5 m above the surface of water, as Kinect has a minimum depth sensing limit of $\sim$50 cm. They were able to generate a 3D mesh of submerged rocks from depth images up to 30 cm below the water level. Butkiewicz [19] discussed Kinect$^{\text{ToF}}$ infrared camera distortion model and its issues. He obtained initial results for 3D scanning from above the water surface, showing that the Kinect$^{\text{ToF}}$ can acquire data up to a distance of 1 m underwater. Digumati *et al.* [20] were the first ones to successfully demonstrate 3D reconstruction from low-cost depth sensors in controlled and real underwater environments. Using the Intel Realsense structured light depth camera, the authors demonstrated a system capable of capturing depth data of underwater surfaces up to 200 mm. They presented two models for calibration of structured light and time of flight cameras for underwater depth imaging. The models incorporate refraction of light and different mediums between the image plane and the object of interest, however, results for only the structured light model processed at 1fps were discussed.

RGB-D cameras like Kinect$^{\text{SL}}$, Kinect$^{\text{ToF}}$ and Intel Realsense, use active optical sensing techniques that work in the Near Infra-Red (NIR) region. The working wavelength of both Kinect$^{\text{SL}}$ and Kinect$^{\text{ToF}}$ is approximately 850 nm [11]. The attenuation or loss of light in water is governed by the combined effects of scattering and absorption and increases exponentially over the length of travel in a medium. NIR wavelength has a much higher attenuation rate in water than visible wavelengths [21]. For 850 nm wavelength in water, the absorption coefficient $a(\lambda)$ is 2.02 m^{-1}, scattering coefficients $b(\lambda)$ is 0.00029 m^{-1} and attenuation coefficient (k) comes out to be 2.02 m^{-1}, as calculated by [22] and [23]. Due to this high absorption rate, maximum amount of infrared radiation from the source is absorbed within the first few meters of water when the source is out of water. Alternately, if the NIR source is immersed in water, the distance that these light wavelengths can travel is severely attenuated.

Another important consideration for underwater imaging is the refractive index of the imaging medium which has a noticeable impact on all imaging sensors. For a standard pinhole camera, light from the scene is focused on a focal point using a lens. When a camera is immersed in water, the casing of an underwater camera also acts as a lens and results in refraction of light. Jordt *et al.* [24], [25] presented a geometrical imaging model for 3D reconstruction using optical cameras, covering the effects due to housing in underwater image formation process. The effect of refraction is also seen in the NIR wavelength in the form of radial distortion when acquiring depth images using Kinect$^{\text{ToF}}$ [19]. Digumarti *et al.* [20] proposed a method to cater for the refraction of various mediums on ToF cameras. However, the method is yet to be tested on a commercial ToF based camera underwater.

The possibility of using Kinect$^{\text{ToF}}$ for underwater applications was explored in our previous work [26] where we presented proof of concept of Kinect$^{\text{ToF}}$ data acquisition when fully immersed in water. We showed that noisy data acquired from the Kinect$^{\text{ToF}}$ could be preprocessed using basic median filtering before passing on to Kinect Fusion.

This paper presents comprehensive and detailed imaging analysis of Kinect$^{\text{ToF}}$ underwater calibration results and corresponding applied corrections. We also propose a time of flight correction method and an efficient refraction correction technique. The proposed methods are used for correcting the error in the acquired depth data and for removing distortion in the point cloud acquired underwater, respectively. Effects of the transparent housing material, as well as the air gap between the imaging plane and casing, are also accommodated in the proposed techniques. Details of our publicly available underwater dataset of depth, IR and RGB data acquired from the Kinect$^{\text{ToF}}$ are also discussed. Results of the data acquired and corrected have been analyzed qualitatively and quantitively in different environments and conditions, such as open air, clear and turbid waters. Our proposed solution for 3D scene reconstruction in underwater environment utilizes the Kinect Fusion algorithm [27], [28] using the Kinect SDK due to its extensive support for the Kinect$^{\text{ToF}}$ sensor.

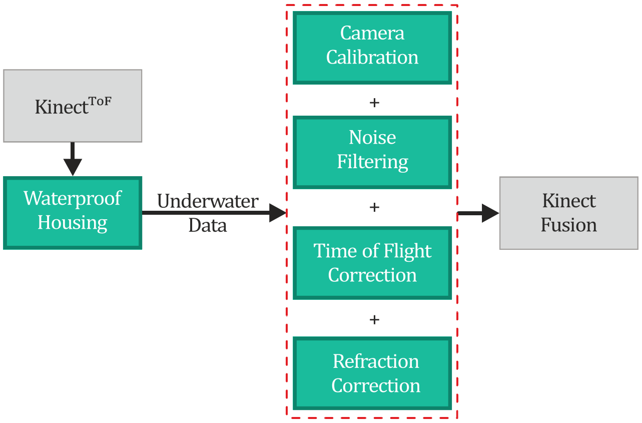


FIGURE 1. **Proposed workflow for 3D scene reconstruction from Kinect$^{\text{ToF}}$ sensor in underwater environment. The colored blocks highlight the proposed method.**

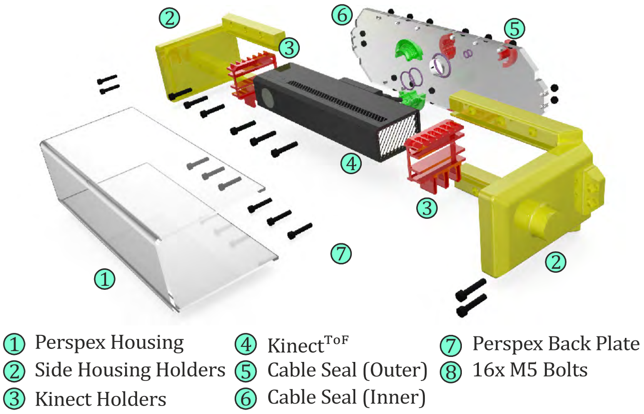


FIGURE 2. **Exploded view of the 3D printed waterproof housing assembly and Perspex for Kinect$^{\text{ToF}}$.**

III. DATA ACQUISITION, CAMERA CALIBRATION, TIME OF FLIGHT CORRECTION AND REFRACTION CORRECTION IN UNDERWATER 3D DATA

The methodology proposed by this work is given in Fig. 1, highlighted by the colored blocks. The first part is the design of a custom-built casing so that operation of the device is not hampered in any way in an underwater environment. The casing material in the field of view of the Kinect$^{\text{ToF}}$ must be selected so that it has minimum effect and absorption on near infrared wavelength. Issues like refraction, noise due to turbidity, absorption, and scattering of NIR wavelength will affect the acquired data. Therefore, corrections for these problems are applied in three separate stages. These additional filters are appended in the preprocessing stage to Kinect Fusion framework and can be enabled or disabled as required.

Kinect$^{\text{ToF}}$ does not have any visible or commonly known Ingress Protection (IP) rating information available. Since significant processing is done onboard the Kinect's specialized processor, it also requires heat dissipation via air circulation, using a 5V 40mm DC Fan inside the casing. There are air intakes on the sides of the device and an exhaust at the rear, for cooling purposes. The waterproof and transparent housing was designed so that not only can it protect the electronic hardware, but also enable a clear and obstruction free view to the onboard IR and RGB sensors. The housing has been developed using transparent Perspex held in place by a 3D printed casing, coated with multiple coats of spray paints and lacquer for waterproofing. Perspex has a refraction index of 1.49 and is light weight with a density of 1.18 g/cm^3. The transmittance of NIR wavelength is approximately ∼95% for the entire NIR range [29]. Therefore, a Perspex housing will not affect the light emitted by the Kinect's NIR source. The effect of NIR absorption due to Perspex was also experimentally verified by testing different thicknesses of Perspex sheets (2 mm, 3 mm, 5 mm and 8 mm) and processing the data through Kinect Fusion in open air. There is almost no visible distortion in the depth map reconstructed for varying thickness of Perspex. The casing for Kinect$^{\text{ToF}}$ was designed in a modular fashion. An exploded view of the assembly is given in Fig. 2. Clear automotive silicone paste was used as a sealing material for the casing. The device itself is held firmly in place limiting any vertical and lateral movement of Kinect$^{\text{ToF}}$ inside the casing while scanning objects

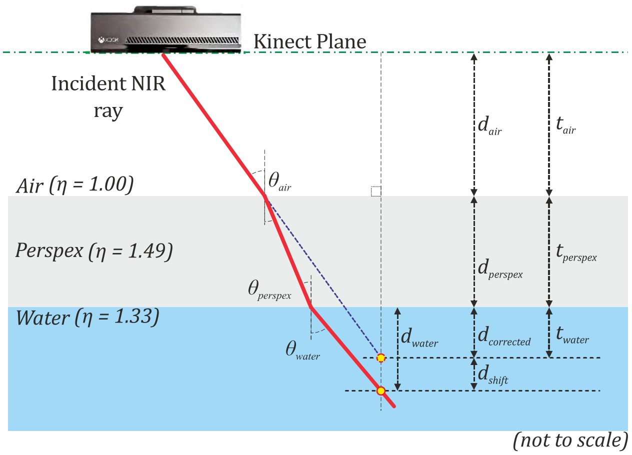


FIGURE 3. Refraction model showing the path of an NIR ray in Kinect$^{\text{ToF}}$ including refraction due to multiple mediums it the path.

under water. The designed housing also allows unrestricted airflow intake and exhaust of Kinect$^{\text{ToF}}$ for cooling during operation. Kinect$^{\text{ToF}}$ needs external power and USB 3.0 connection to the host computer for real-time processing. Power to the device was supplied with an external 12V 7Ah sealed lead-acid battery. Kinect's IR and RGB cameras were calibrated using the open source GML C++ Camera Calibration Toolbox [30]. Reference checkerboard (5x6 matrix with each square size of 30x30mm) was printed and plastic coated to be taken underwater.

A. NOISE FILTERING OF Kinect$^{\text{ToF}}$ DEPTH DATA ACQUIRED UNDERWATER

Water is a rich environment full of suspended microbial, invisible and semi-invisible life forms. Passive imaging sensors working in water must cater for unwanted noise in the images due to absorption and scattering phenomenon. Additionally, active sensors must also accommodate the unwanted reflections of the transmitted rays from these suspended bodies. The noise in the acquired data and must be canceled out actively before it can be used further. Noise removal is achieved with the addition of a layer of noise removal algorithms as part of the pre-processing stage. The depth data acquired underwater from Kinect$^{\text{ToF}}$ has a significant amount of noise which results in erroneous depth values reported for a single pixel over several frames. Kinect Fusion includes a bilinear filter for reducing the noise in the depth data before fusing it in a mesh. Since the air in a normal environment does not have a significant number of suspended particles that reflect infrared rays, therefore a simple bilinear filter is sufficient for normal use. Underwater environment poses a significantly more challenging environment for underwater imaging in any wavelength. For the case of depth images taken underwater, the noise can be categorized as of salt and pepper type, an additional 5x5 window median filter on the acquired depth image was found to perform well with acceptable results.

B. DEPTH VALUE CORRECTION BY ADJUSTING TIME OF FLIGHT VALUES

Kinect$^{\text{ToF}}$ calculates depth by emitting a continuous wave of IR from the embedded NIR source and measuring the phase difference between transmitted and received signals. The hardware calculates the time it takes for the returning beam to calculate the distance. However; Kinect$^{\text{ToF}}$ does not consider the difference of speed of light in different mediums. Although trivial in nature, this results in the creation of 3D meshes at an incorrect distance from the camera position. Since Kinect$^{\text{ToF}}$ measures the time for the return of the NIR rays, the measured depth by Kinect$^{\text{ToF}}$ is reported to be farther than the actual distance to the object, as it takes longer for the signal to return due to light propagating at a slower speed.

Additionally, since Kinect$^{\text{ToF}}$ is encased in a housing, the infrared light must pass three different mediums, as shown in Fig. 3. First is air between the housing inner face and the imaging sensor, second is the housing material and last is water. Originating from Kinect$^{\text{ToF}}$, the NIR beam will therefore face two media transitions. The actual distance measured by Kinect$^{\text{ToF}}$ at each pixel (d_{pixel}) will be a sum of the three different distances, namely: the distance between the image sensor and the housing's inner face (d_{air}), the thickness of the housing material ($d_{perspex}$) and the distance in water to the actual object or surface (d_{water}). Similarly, the time taken by the NIR beam by Kinect$^{\text{ToF}}$ is also split into three different parts, as given in Eqn. (1) and (2):

$$d_{water} = d_{pixel} - (d_{air} + d_{perspex}) \tag{1}$$
$$t_{water} = t_{pixel} - (t_{air} + t_{perspex}) \tag{2}$$

The original time used by Kinect$^{\text{ToF}}$ for calculating the depth (t_{water}) can be found and updated using Eqn. (3) and (4). The new time in water can then be used to calculate the new distance ($d_{corrected}$) using the speed of light in water as given in Eqn. (5). This corrected distance can then be used to update the new depth at each pixel in water by Eqn. (6).

$$t_{pixel} = \frac{d_{pixel}}{c_{air}}, \quad t_{air} = \frac{d_{air}}{c_{air}}, \; t_{perspex} = \frac{d_{perspex}}{c_{perspex}} \tag{3}$$
$$t_{water} = t_{pixel} - (t_{air} + t_{perspex}) \tag{4}$$
$$d_{corrected} = t_{water} * c_{water} \tag{5}$$
$$d_{pixel} = d_{corrected} + d_{air} + d_{perspex} \tag{6}$$

As a result, the updated depth value calculated is closer than the one calculated by Kinect$^{\text{ToF}}$. The same calculation is performed for each 512 × 424 depth image, to get the actual depth at each pixel.

C. REFRACTION CORRECTION

Point clouds acquired by Kinect$^{\text{ToF}}$ underwater show significantly amplified pincushion radial distortion. This distortion is much more visible for depth images of a flat surface where the reconstructed mesh shows a concave shape, bulging towards the camera along the corners and pinched backward from the center. The concave nature of the acquired

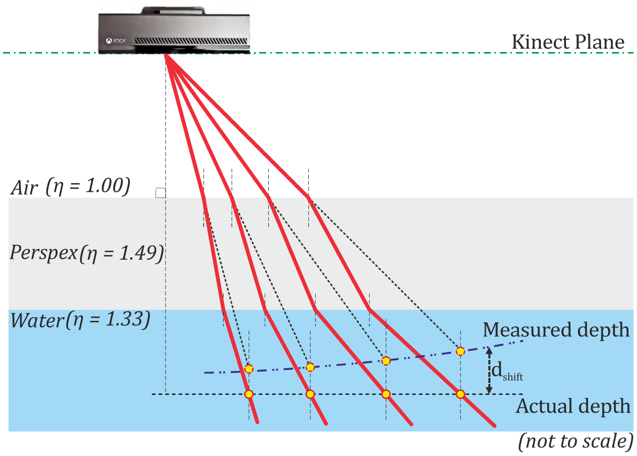


FIGURE 4. Formation of virtual image at an offset than the actual position due to refraction.

point cloud can be visualized as shown in Fig. 4, which illustrates the detection of a depth point in water. The point is detected in-line with the incident NIR ray along the normal to the actual point. As we move away from the center, the error increases with the increase in the angle of incidence of the NIR ray. Note that it is assumed that NIR emitters and the depth camera sensor have no horizontal offset between them and the principal axis of the depth camera is aligned with the optical axis of the image sensor, for the sake of simplification of the problem. Using Snell's law, it can be shown that the shift in the depth (d_{shift}) is dependent on incident and refraction angles and the actual depth measured. Using the nomenclature defined in III-A, we define the incident angle as θ_{air} and refracted angle in Perspex as $\theta_{perspex}$ for a single ray originating from Kinect$^{\text{ToF}}$ and passing through the space of air between the sensor surface and the Perspex housing. The refracted angle from Perspex to water is given by θ_{water}. Note that the only numerical measures available are the uncorrected depth value and the thickness of Perspex. Using the Law of Sines the shift in the depth d_{shift} can be calculated by Eqn. (7)

$$d_{shift} = \frac{d_{corrected}}{\sin(90 - \theta_i)} \cdot \frac{\sin(\theta_i - \theta_r)}{\sin \theta_r} \tag{7}$$

where:

$$\begin{aligned} d_{shift} &= \text{shift in distance measured} \\ d_{corrected} &= \text{ToF distance} \\ \theta_i, \theta_r &= \text{Incident and refracted angles} \end{aligned}$$

This shift must be incorporated for every pixel in the (512 × 424) depth image by calculating the incident angles, refraction angles and radial depths of each pixel with respect to the principal axis of the depth camera. The incident and refracted angles are first calculated for the transition from air to Perspex. The refracted ray then becomes the incident ray and the angles are recalculated for Perspex to water. To calculate the incident and refracted angles, we assume that a single ray of light is generated for every pixel of the image sensor and returns to the same sensor pixel after being reflected by the scene in the FoV. This method is analogous to the ray-casting technique used in the field of computer graphics. We first translate all the pixel locations from the 2-dimensional Cartesian coordinate system (x, y) to a 3-dimensional spherical coordinate system (r, θ, φ). This is done by simple rectangular to spherical trigonometric transformation, resulting in three 512 × 424 matrices, one each for the radial distance r, inclination angle (in radians) and azimuth angle φ (also in radians). Using the θ and φ angles, we can calculate orthogonal projections of the radial distance r along the z-axis.

Mathematically, let the spherical coordinates of the ray segment inside the housing be denoted by r_{z-air}, its projections by r_{zx-air}, r_{zy-air} and angles by θ_{air}, φ_{air}, respectively. After striking Perspex, the NIR ray changes path after refraction and θ_{air}, φ_{air} become the incident angles. This new path is denoted by $r_{z-perspex}$, $r_{zx-perspex}$, $r_{zy-perspex}$ and $\theta_{perspex}$, $\varphi_{perspex}$ for magnitude, projections, and angles inside Perspex, respectively. Lastly, the NIR ray is refracted again as it moves from Perspex to water and the path is traced to depth values in the water. The spherical coordinates are denoted by $r_{z-water}$, $r_{zx-water}$, $r_{zy-water}$ and θ_{water}, φ_{water} for magnitude, projections, and angles in water, respectively. The magnitude $r_{zy-water}$ will be the new depth value in water along the z-axis, incorporating the effect of refraction of all the mediums as it has been traced along the entire path. The error in distances for each pixel (d_{pixel}) can be calculated by subtracting the original depth value (d_{pixel}) from $r_{zy-water}$. Pseudocode of the devised method is given as algorithm 1.

The benefit of using the ray-casting inspired method as opposed to the ray-tracing based methods is that it results in a much more efficient and non-computationally intensive algorithm that can cater for the refraction effect of all the different mediums involved during depth acquisition.

IV. Kinect$^{\text{ToF}}$ PERFORMANCE IN UNDERWATER AND MESH RECONSTRUCTION RESULTS

To acquire Kinect$^{\text{ToF}}$ data underwater, several data acquisition experiments were done in controlled environments. The locations used were the Offshore Experiment Lab and the swimming pool available within university premises with a maximum depth of 1.5 m. Stagnant water contained in the offshore tank had slight turbidity with suspended particles and small amounts of algae. The swimming pool provided clear water free from any visible suspended particles. A customized Kinect Fusion implementation including the proposed noise filters, camera calibration, time of flight correction and refraction correction algorithms was developed using the Kinect Fusion SDK, in XAML and C#. To validate the results, reconstructed meshes needed to be evaluated qualitatively and quantitatively. However, no public dataset of depth images acquired underwater is available to compare this work quantitatively. Therefore, 3D printed objects with available 3D CAD models were used as ground truth to

Algorithm 1 Refraction Correction

```
// get incident ray matrix ($\theta$ and $\varphi$)
for $i$ = +Vertical FOV → -Vertical FOV in 424 steps
    for $j$ = +Horizontal FOV → -Horizontal FOV in 512 steps
        $\varphi_{Air} = i, \theta_{Air} = j$;
    end
end
for each $d_{pixel} \in$ depth image $\mathbf{D_i}$
        // calculate new refracted angles for air → Perspex (Snell's Law)
        $[\theta_{perspex}, \varphi_{perspex}]$ = calculateRefractedAngles ($\theta_{Air}, \varphi_{Air}, n_{air}, n_{perspex}$);
        // calculate ray projection magnitudes for Perspex
        $[r_{zx\_perspex}, r_{zy\_perspex}]$ = calculateRayProjections ($\theta_{perspex}, \varphi_{perspex}$);
        // calculate new refracted angles for Perspex → water (Snell's Law)
        $[\theta_{water}, \varphi_{water}]$ = calculateRefractedAngles ($\theta_{perspex}, \varphi_{perspex}, n_{perspex}, n_{air}$);
        // calculate ray projection magnitudes for water
        $[r_{zx\_water}, r_{zy\_water}]$ = calculateRayProjections ($\theta_{water}, \varphi_{water}$);
end
$d_{corrected} = r_{zy_{water}}$;
$d_{Shift} = r_{zy_{water}} - d_{pixel}$;
```

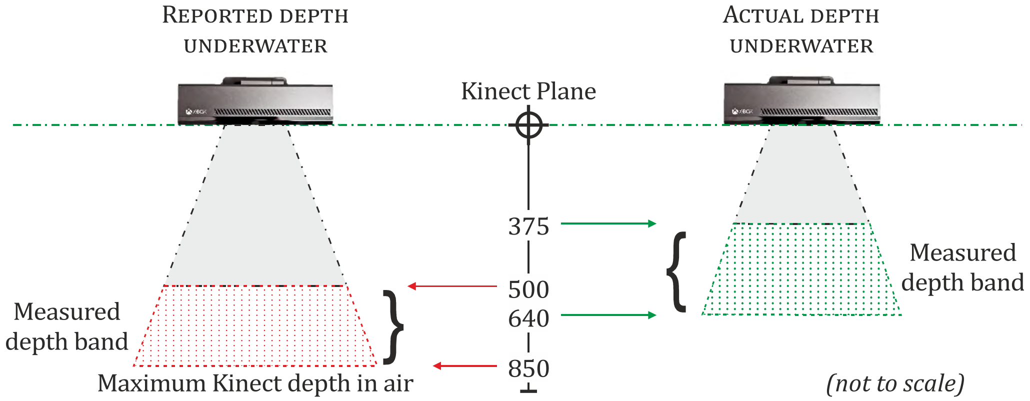


FIGURE 5. **Reported depth vs. actual depth of Kinect^{ToF} in underwater environment.**

evaluate the results after scanning them underwater. A complete dataset was acquired consisting of various objects. The data from Kinect's RGB and IR cameras was captured alongside the generated point cloud by Kinect^{ToF} and saved in Microsoft's eXtended Event File (XEF) file format that can be used with Kinect Studio application. The dataset, as well as the code developed, is publicly available[1] under GNU GPL 3.0 license.

Since Kinect^{ToF} camera uses ~800-830nm NIR wavelength, the depth measurement capability was strongly attenuated in underwater environment as NIR has a much higher absorption rate in water as compared to the visible spectrum. Experimental results showed that the Kinect^{ToF} successfully reported depth and generated accurate dense point clouds of its FoV between a minimum of 500 mm and a maximum of ~850 mm. However, the actual working depth of Kinect^{ToF} was found to be around ~350 mm to ~650 mm instead of the reported 500 mm to ~850 mm depth. The attenuated performance of Kinect^{ToF} is illustrated using Fig. 5. This difference in reported versus actual depth was due to the time of flight calculation error and was adjusted by the proposed ToF correction method described in section 3.

While the RGB camera calibration was not that problematic especially when done in clear water, the calibration of the IR camera under water was not straight forward. The infrared images acquired were of very low intensity due to the absorption of NIR. Therefore, the images had to be enhanced using simple contrast adjustment techniques before calibration. The estimated distortion parameters are given in Table 1.

Kinect Fusion has several parameters that can be modified at runtime for achieving a better working performance, such as voxel resolution, the number of voxels per axis and the integration weight. The trade-off is between the increase or decrease in reconstruction performance, quality of the mesh generated and area of the scene reconstructed.

[1] The complete dataset, design files and the code can be downloaded at the following link: http://bit.ly/UWKFusion

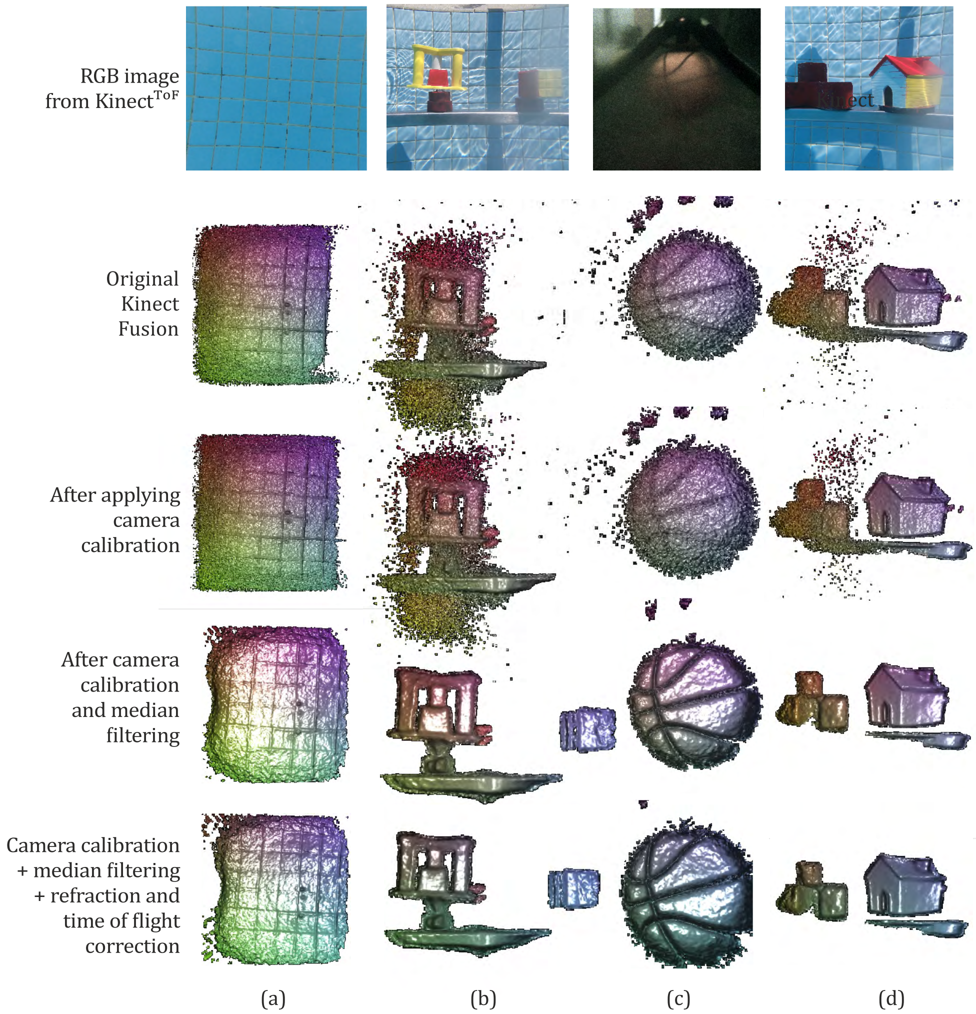

RGB image from Kinect^{ToF}
Original Kinect Fusion
After applying camera calibration
After camera calibration and median filtering
Camera calibration + median filtering + refraction and time of flight correction
(a)
(b)
(c)
(d)

FIGURE 6. Intermediate and final results of underwater 3D reconstruction of different objects using Kinect^{ToF}. (a) Swimming pool wall (b) 3D printed trophy stand (c) basket ball (d) 3D printed house model.

TABLE 1. RGB and IR camera calibration results in air and underwater.

	Parameter Name		*In air*	*Underwater*
RGB camera calibration results	Focal	(f_x)	1032.6645	1947.3593
	length	(f_y)	1033.1741	1952.8560
	Principal	(c_x)	972.3426	983.5870
	Point	(c_y)	532.6476	587.2310
	Distortion	A	0.0870	1.0543
	Coefficients	B	-0.1651	-2.0823
		C	-0.0032	0.0175
		D	-0.0034	0.01676
IR camera calibration results	Focal	(f_x)	391.096 ± 36.144	717.364895 ± 58.516
	length	(f_y)	463.098 ± 104.984	688.007761 ± 57.642
	Principal	(c_x)	243.892 ± 09.838	281.306232 ± 06.561
	Point	(c_y)	208.922 ± 58.667	300.153537 ± 32.958
	Distortion	a	0.13454	1.58070
	Coefficients	b	-0.24154	-1.82717
		c	-0.02839	0.20461
		d	-0.01516	-0.00032

As Kinect Fusion is not designed to be a memory efficient algorithm [28] but focuses more on the speed of reconstruction, the mesh being constructed cannot be very high resolution and simultaneously cover a considerable area of the scene. For our experiments, the values selected for voxel size along the X and Y axes was 256, while the voxel size along the z-axis was chosen to be 384. The voxel resolution was set at 384 v/m and an integration weight was selected to be 500. Fig. 6 shows the results of 3D mesh of a tiled swimming pool wall (Fig. 7(a)) and selected objects (Fig. 7(b, c, d)) generated from the original Kinect Fusion code, and compared to the mesh after applying camera calibration, noise filtering, ToF and refraction corrections.

There are several noticeable differences visible in the resulting meshes. Firstly, median filtering was effective in removing the noise from the depth images acquired and enabled a much cleaner mesh. Features of objects scanned were much clearer as compared to unfiltered data. In some cases, like Fig. 7(a) and (b), after applying camera calibration and median filtering, Kinect Fusion could generate a slightly larger mesh and recover some objects that were lost due to the noisy data such as the Rubik's cube model. The proposed refraction correction also improved the generated mesh, and the effect could be seen much more clearly when a flat tiled wall was reconstructed. Fig. 7 shows the front and side view

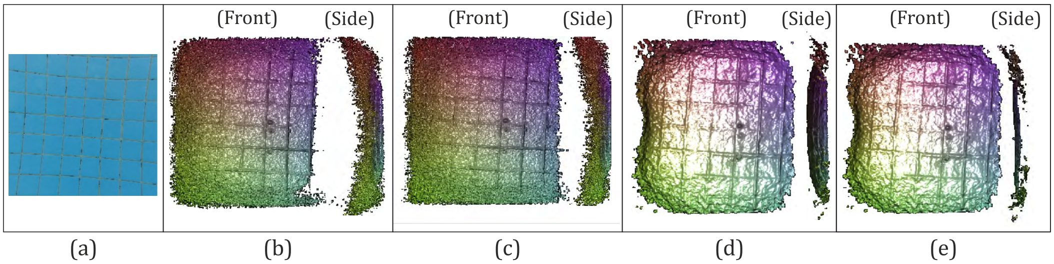


FIGURE 7. Front and side view of 3D meshes of the flat tiled swimming pool wall reconstructed from unaltered Kinect Fusion as well as all intermediate steps of proposed correction algorithm. (a) RGB image from Kinect (b) original Kinect Fusion mesh (c) applying camera calibration (d) applying camera calibration and median filtering (e) applying camera calibration, median filtering, refraction correction and ToF correction.

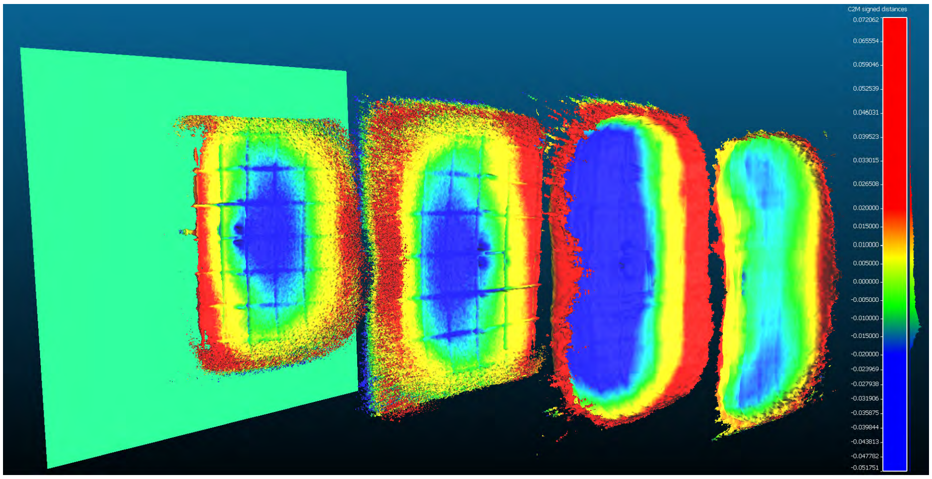


FIGURE 8. Alignment error maps of 3D reconstructed mesh of a submerged swimming pool wall compared with an ideal plane, showing the refraction correction results. green represents 0 mm error, red represents $\geq +20$ mm error, blue represents ≥ -20mm error. (From Left to right: Ideal reference plane, original Kinect Fusion mesh, after camera calibration, after median filtering, ToF and refraction corrected mesh).

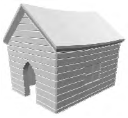
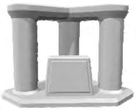
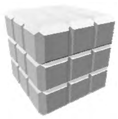
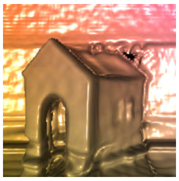
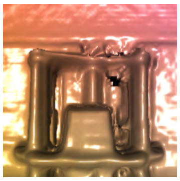
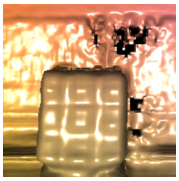
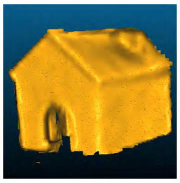
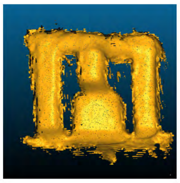
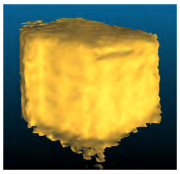
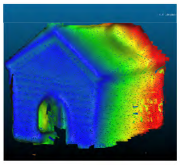
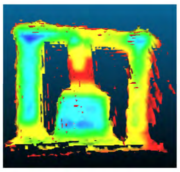
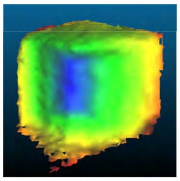
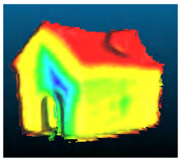
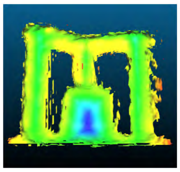
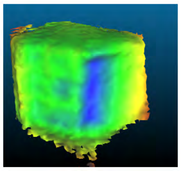
of the 3D meshes, of the flat tiled swimming pool wall and all intermediate steps of the proposed method.

Effect of camera calibration on the depth data is visible in the front view of Fig. 8(c) where the tile lines were straighter after incorporating the lens distortion correction. The concave nature of the uncorrected data is visible in the side view of Fig. 8(d), after the noisy depth data was filtered to generate a cleaner surface. The result of the proposed refraction correction algorithm is visible inside view of Fig. 8(e) where the concave nature of the mesh was adjusted, and the generated surface was significantly flatter. Effect of ToF correction was simply the reduction in reconstruction distance of the overall mesh, as it was evenly applied on the entire depth data. There was no visible or measurable effect of ToF correction on the quality of the mesh generated.

To quantitatively analyze the effect of the proposed refraction correction algorithm, alignment error maps of generated mesh were compared with the scanned mesh with an ideal plane mesh as shown in Fig. 8. green color in the heat map represents 0 to ±5 mm error. Red represents $\geq +20$ mm error whereas blue represents ≥ -20mm error.

After applying median filter, quality of the mesh increased significantly; the error decreased and noise was removed. The significant inward bulge in the center (blue) due to refraction was reduced after applying refraction correction. The mean error of the mesh after applying refraction correction was 1.3 mm with a standard deviation of 14 mm error. The slightly higher standard deviation was due to the large errors along the edges and scattered mesh points as we move away from the center of the depth image. Table 2 gives the alignment error heat maps and error histogram Gaussian distribution of the several objects whose 3D model were available to be used as ground truth. These objects were 3D printed and then scanned in air as well as underwater. The corrected underwater mesh was then compared to the original 3D model and the 3D mesh scanned in air, to generate an error histogram that is represented as a heat map on the 3D mesh for visual clarity. As can be seen in the results, we were able to achieve a mean error of ±6 mm with an average stand deviation of 3 mm, thereby confirming that the reconstructed meshes were a close approximation of the object being scanned. It is worth noting that Kinect Fusion is designed to work on dense point clouds, as it uses the entire depth frame for alignment using ICP. Because of NIR absorption and refraction in water, there was a significant loss depth data and the point cloud returned was sparse in nature. If the point cloud is sparse, then

TABLE 2. Error heat maps and gaussian distribution of error histogram of various objects scanned underwater. Objects scanned are compared to original 3D CAD model as well as with the 3D printed model scanned with Kinect^{ToF} in the air.

Description		3D Printed House	3D Printed Trophy Stand	3D Printed Rubik's Cube
Ground truth	3D model of object			
	3D mesh scanned in air			
3D mesh scanned underwater				
Underwater mesh vs 3D model	Alignment error			
	Error histogram statistics	Min error 0.0 Max error 0.00335 Avg. error 0.00076 Std. dev (σ) 0.00076	Min error 0.0 Max error 0.10200 Avg. error 0.00652 Std. dev (σ) 0.00735	Min error 0.0 Max error 0.03302 Avg. error 0.00737 Std. dev (σ) 0.00533
Underwater mesh vs air scan	Alignment error			
	Error histogram statistics	Min error 0.0 Max error 0.00335 Avg. error 0.00076 Std. dev (σ) 0.00076	Min error 0.0 Max error 0.10929 Avg. error 0.00033 Std. dev (σ) 0.00684	Min error 0.0 Max error 0.02655 Avg. error 0.00430 Std. dev (σ) 0.00399

calculating the local minimum of each frame for alignment is difficult, therefore the tracking was frequently lost while scanning. The importance of point cloud in the first frame of the mesh reconstruction process was increased significantly since ICP algorithm aligns the frames acquired in succession to the first frame of the reconstructed scene.

At the time of writing of this work, there has been no other detailed performance analysis or refraction correction method proposed for fully submerged Kinect^{ToF} depth data acquisition. The closest research on data acquired by fully submerged RGB-D camera is by Digumarti et al using the Intel RealSense, which is a structured light sensor like Kinect^{SL}. Furthermore, there was no data (point clouds, 3D meshes or raw dataset) provided by the authors for doing a qualitative or quantitative comparison of the output of our proposed method. Therefore, a direct quantitative comparison with their method was not possible. However, major differ-

TABLE 3. Comparison of proposed work with Digumarti et al.

	Digumarti et al.	Proposed work
Sensor	Intel RealSense (Structured Light)	Kinect^{ToF} (Time of Flight)
Fusion technique	InfiniTAM	Kinect Fusion
Refraction correction	Ray-tracing based	Ray-casting based
Camera calibration	Yes	Yes
Frame rate	1 fps	5 - 10 fps
Scanning distance	Up to 200 mm	350 mm - 650 mm

ences between their work and our proposed method are given in Table 3. We were able to achieve a frame rate of upto 10 fps on a system with Core-i7, 8GB Ram, Nvidia GTX 765m graphics card and 256GB SSD storage.

V. CONCLUSION

Commercial depth cameras have recently been tested to work underwater providing high-resolution 3D meshes but at reduced depth performance. Our proposed work analyzes the performance of Kinect$^{\text{ToF}}$ sensor underwater, in a specially designed housing that provides waterproofing without affecting the NIR transmission of the sensor. We show that Kinect$^{\text{ToF}}$ is successfully able to acquire point cloud data up to ~600 mm but with significant noise. The noise is removed by applying a standard median filter on the depth data. Since Kinect$^{\text{ToF}}$ is designed to work in open air, the depth data does not incorporate the slower speed of light in the water, for which we propose a simple time of flight correction method to correct the distances reported by Kinect sensor. Camera calibration results of Kinect$^{\text{ToF}}$ underwater have been presented, and the effect of calibration on the 3D mesh generated is also discussed. We also propose a fast and intuitive refraction correction method that caters for the effect of refraction of NIR underwater due to the housing and water and can reconstruct 3D meshes with a mean error of ±6mm up to 10 fps. The scanning performance of our proposed work can be significantly enhanced by developing a parallelized implementation using GPU. The acquired underwater dataset including RGB, IR and depth data, proposed method codes, as well as the design of the waterproof Kinect$^{\text{ToF}}$ casing has been made publicly available for further research. The objectives of this research were to evaluate the substitution of expensive and specialized optical depth sensors such as lidars with economical and easily available sensors for small-scale solutions and activities for 3D scene reconstruction and 3D scanning underwater. Areas such as coral reef mapping and underwater SLAM for a robotic solution in shallow waters can benefit from the results achieved by this research.

REFERENCES

[1] A. Hogue and M. Jenkin, "Development of an underwater vision sensor for 3D reef mapping," in *Proc. Int. Conf. Intell. Robots Syst. (IROS)*, 2006, pp. 5351–5356.

[2] G. Söhnlein, S. Rush, and L. Thompson, "Using manned submersibles to create 3D sonar scans of shipwrecks," in *Proc. IEEE OCEANS*, Sep. 2011, pp. 1–10.

[3] E. Lachat, H. Macher, T. Landes, and P. Grussenmeyer, "Assessment and calibration of a RGB-D camera (Kinect v2 sensor) towards a potential use for close-range 3D modeling," *Remote Sens.*, vol. 7, no. 10, pp. 13070–13097, 2015.

[4] F. Järemo Lawin, "Depth data processing and 3D reconstruction using the Kinect v2," M.S. thesis, Dept. Electr. Eng., Linköping Univ., Linköping, Sweden, 2015.

[5] A. Corti, S. Giancola, G. Mainetti, and R. Sala, "A metrological characterization of the Kinect V2 time-of-flight camera," *Robot. Auto. Syst.*, vol. 75, pp. 584–594, Jan. 2016.

[6] H. Gonzalez-Jorge *et al.*, "Metrological comparison between Kinect I and Kinect II sensors," *Measurement*, vol. 70, pp. 21–26, Jun. 2015.

[7] M. G. Diaz, F. Tombari, P. Rodriguez-Gonzalvez, and D. Gonzalez-Aguilera, "Analysis and evaluation between the first and the second generation of RGB-D sensors," *IEEE Sensors J.*, vol. 15, no. 11, pp. 6507–6516, Nov. 2015.

[8] M. Bueno, L. Díaz-Vilariño, J. Martínez-Sánchez, H. González-Jorge, H. Lorenzo, and P. Ariasa, "Metrological evaluation of KinectFusion and its comparison with microsoft Kinect sensor," *Measurement*, vol. 73, pp. 137–145, Sep. 2015.

[9] C. Dal Mutto, P. Zanuttigh, and G. M. Cortelazzo, *Time-of-flight cameras and Microsoft Kinect (TM)*. New York, NY, USA: Springer, 2012.

[10] H. Gonzalez-Jorge, B. Riveiro, E. Vazquez-Fernandez, J. Martínez-Sánchez, and P. Arias, "Metrological evaluation of microsoft Kinect and ASUS Xtion sensors," *Measurement*, vol. 46, no. 6, pp. 1800–1806, Jul. 2013.

[11] H. Sarbolandi, D. Lefloch, and A. Kolb, "Kinect range sensing: Structured-light versus time-of-flight Kinect," *Comput. Vis. Image Understand.*, vol. 139, pp. 1–20, Oct. 2015.

[12] N. M. DiFilippo and M. K. Jouaneh, "Characterization of different microsoft Kinect sensor models," *IEEE Sensors*, vol. 15, no. 8, pp. 4554–4564, Aug. 2015.

[13] L. Shao, J. Han, P. Kohli, and Z. Zhang, *Computer Vision and Machine Learning With RGB-D Sensors*. Switzerland: Springer, 2014.

[14] M. Hildebrandt, "Development, evaluation and validation of a stereo camera underwater SLAM algorithm," Ph.D. dissertation, Dept. Electr. Eng., Linköping Univ., Linköping, Sweden, 2014.

[15] K. Khoshelham, "Accuracy analysis of kinect depth data," in *Proc. ISPRS Workshop Laser Scanning*, 2011, vol. 38. no. 5, p. W12.

[16] H. Lu *et al.*, "Depth map reconstruction for underwater Kinect camera using inpainting and local image mode filtering," *IEEE Access*, vol. 5, pp. 7115–7122, 2017.

[17] C.-L. Tsui, D. Schipf, K.-R. Lin, J. Leang, F.-J. Hsieh, and W.-C. Wang, "Using a Time of Flight method for underwater 3-dimensional depth measurements and point cloud imaging," in *Proc. IEEE OCEANS*, Apr. 2014, pp. 1–6.

[18] A. Dancu, M. Fourgeaud, Z. Franjcic, and R. Avetisyan, "Underwater reconstruction using depth sensors," in *Proc. SIGGRAPH Asia Tech. Briefs*, 2014, p. 2.

[19] T. Butkiewicz, "Low-cost coastal mapping using Kinect v2 time-of-flight cameras," in *Proc. Oceans-St. John's*, Sep. 2014, pp. 1–9.

[20] S. T. Digumarti, G. Chaurasia, A. Taneja, R. Siegwart, A. Thomas, and P. Beardsley, "Underwater 3D capture using a low-cost commercial depth camera," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2016, pp. 1–9.

[21] A. Maccarone *et al.*, "Underwater depth imaging using time-correlated single-photon counting," *Opt. Exp.*, vol. 23, no. 26, pp. 33911–33926, 2015.

[22] K. Shifrin, *Physical Optics of Ocean Water*. New York, NY, USA: AIP, 1988.

[23] J. A. Curcio and C. C. Perry, "The near infrared absorption spectrum of liquid water," *J. Opt. Soc. Amer. A*, vol. 41, no. 5, pp. 302–304, May 1951.

[24] A. Jordt, K. Köser, and R. Koch, "Refractive 3D reconstruction on underwater images," *Methods Oceanography*, vols. 15–16, pp. 90–113, Apr./Jul. 2016.

[25] A. Jordt, "Underwater 3D reconstruction based on physical models for refraction and underwater light propagation," Ph.D. dissertation, Dept. Faculty Eng., Kiel Univ., Germany, 2014.

[26] A. Anwer, S. S. A. Ali, A. Khan, and F. Mériaudeau, "Real-time underwater 3D scene reconstruction using commercial depth sensor," in *Proc. IEEE Int. Conf. Underwater Syst. Technol., Theory Appl. (USYS)*, Dec. 2016, pp. 67–70.

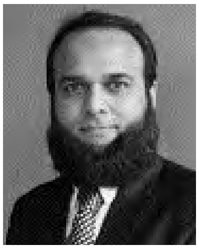
[27] R. A. Newcombe *et al.*, "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Reality*, Oct. 2011, pp. 127–136.

[28] S. Izadi *et al.*, "KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera," in *Proc. 24th Annu. ACM Symp. User Interface Softw. Technol.*, 2011, pp. 559–568.

[29] *Optical & Transmission Characteristics—Plexiglass.com*. Accessed on Mar. 24, 2016. [Online]. Available: http://www.plexiglas.com/export/sites/plexiglas/.content/medias/downloads/sheet-docs/plexiglas-optical-and-transmission-characteristics.pdf

[30] *GML C++ Camera Calibration Toolbox*. Accessed on Feb. 10, 2017. [Online]. Available: http://graphics.cs.msu.ru/en/node/909

ATIF ANWER received the bachelor's degree in mechatronics engineering from the National University of Sciences and Technology, Islamabad, Pakistan, in 2004. He is currently pursuing master's degree in electrical and electronics engineering by research from Universiti Teknologi Petronas, Malaysia. He has a background in robot vision, embedded systems, mechanical design, additive manufacturing, and control system design and implementation. He is also associated with the Centre for Intelligent Signal and Imaging Research. His interests include mobile robotics, robot and machine vision, and underwater robotics.



SYED SAAD AZHAR ALI (M'16–SM'17) received the B.E. degree in electrical engineering from NED University, Karachi, Pakistan, and the master's and Ph.D. degrees in nonlinear control from the King Fahd University of Petroleum and Minerals. He was with Air University and Iqra University prior to being engaged with the Center of Intelligent Signal and Imaging Research, Universiti Teknologi Petronas. He leads the Visual Surveillance and Monitoring Group. Recently, he has been involved in neurosignal processing. He has authored over 60 peer-reviewed publications, including four books/chapters. His research focus has been on intelligent control, signal processing, underwater robotics with the emphasis on image enhancement and 3-D scene reconstruction. He is the PI for several funded research projects.

AMJAD KHAN received the bachelor's degree in electronic engineering from Iqra University, Karachi, Pakistan, in 2014. He is currently pursuing M.Sc. degree in electrical and electronics engineering by research from Universiti Teknologi Petronas, Malaysia. He has served the industry for many years under different capacities. He is with the Centre for Intelligent Signal and Imaging Research. His interests include image processing and visual surveillance.

FABRICE MÉRIAUDEAU was born in Villeurbanne, France, in 1971. He received the master's degree in physics from Dijon University, France, the Engineering degree (Hons.) in material sciences in 1994, and the Ph.D. degree in image processing with Dijon University in 1997. He held a post-doctoral position for a year with the Oak Ridge National Laboratory. He was the Director of the Le2i (UMR CNRS), which has more than 200 staff members, from 2011 to 2016. He has coordinated an Erasmus Mundus Master in the field of computer vision and robotics from 2006 to 2010 and was the Vice President for International Affairs with the University of Burgundy from 2010 to 2012. Since 2016, he has been with Universiti Teknologi Petronas as a Professor. He is currently a Professeur des Universites. He has authored and co-authored over 150 international publications and holds three patents. His research interests focused on image processing for non-conventional imaging systems (UV, IR, and polarization) and more recently on medical/biomedical imaging.