Dynamic 3D Capture of Swimming Fish by Underwater Active Stereo

Ryo Kawahara

Shohei Nobuhara

Takashi Matsuyama

Graduate School of Informatics, Kyoto University

Abstract

This paper presents an underwater active stereo system that realizes 3D capture of dynamic objects in water such as swimming fish. The key idea on realizing a practical underwater 3D sensing is to model the refraction process by our pixel-wise varifocal camera model that provides efficient forward (3D to 2D) projections as well as an underwater projector-camera calibration. Evaluations demonstrate that our method achieves reasonable calibration accuracy using off-the-shelf cameras and projectors, and provides a 3D capture of real swimming fish in water.

Key words: Underwater active stereo, 3D shape reconstruction, Refraction, Calibration

1. Introduction

Image-based 3D shape modeling has been a fundamental research goal in computer vision for decades, and recent advances realized practical solutions for capturing dynamic objects such as human[1, 2, 3]. Today it provides nonconstrained and non-invasive measurements of its 3D shape and motion, and widely used as a marker-less motion capture[4, 5, 6, 7] for movie production, medical analysis, and so forth. The goal of this paper is to realize such an image-based 3D acquisition for underwater objects as shown in Figure 1. We believe that providing an automated quantitative 3D sensing of underwater objects will help advancing studies in oceanography, marine biology, aquaculture, etc.

Realizing an image-based underwater 3D capture is not a trivial problem even though many algorithms have been developed for objects in the air. The difficulty can be found in the environment and the object. Capturing images in water inevitably

Email addresses: kawahara@vision.kuee.kyoto-u.ac.jp (Ryo Kawahara), nob@i.kyoto-u.ac.jp (Shohei Nobuhara), tm@i.kyoto-u.ac.jp (Takashi Matsuyama)

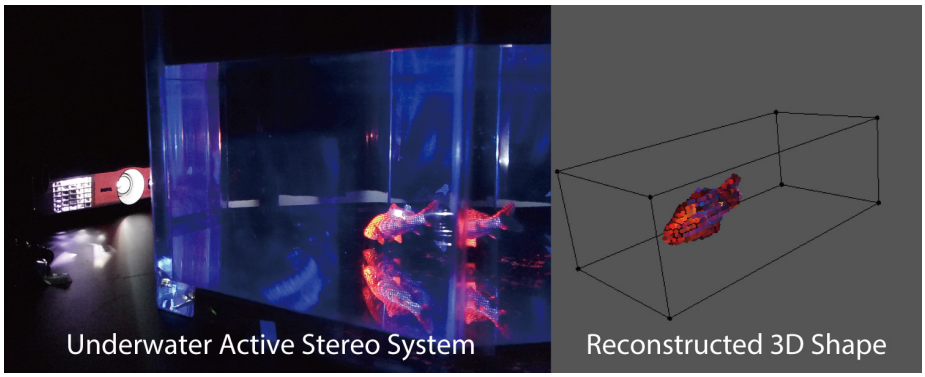


Figure 1: Dynamic 3D capture of swimming fish by underwater active stereo. Left: our capture system consisting of cameras and projectors observing the fish via flat surfaces. Right: reconstructed 3D shape.

involves refractive image distortions by housings[8, 9], and image noise and intensity attenuations by unclear media[10]. Also the objects to be captured in water are likely to show poorly-textured surfaces with transparency and specularity. These properties are not dominant in 3D capture in the air, and hence should be handled explicitly for underwater 3D capture.

The key problem addressed in this paper is the refraction by water-proof housings that lead to incorrect 3D reasoning unless modeled correctly. That is, if the refraction is ignored and the ray is wrongly modeled as a straight line passing through the camera center as in the air, 3D measurement by stereo cannot return the correct triangulation point obviously since the ray passing through the camera center and the pixel should be refracted by the housing.

The refraction is well described by Snell's law. The motivation of this paper is not to introduce another law, but to introduce a new practical representation of the refraction described by Snell's law. We show that our representation provides a faster computation of 3D-to-2D forward projection as well as a linear calibration of projector-camera systems in water.

The key idea of our approach is to introduce a virtual camera model that defines a focal length on a per-pixel basis, and is to exploit a radial structure of such pixel-wise focal lengths to realize a compact and efficient representation of the refraction. As is well known, the refraction by the housing result in a caustic structure of rays as shown in Figure 2(a), and this structure deforms depending on the relative pose of the housing and the camera. We show that our model can model cameras and projectors behind flat housings regardless of their poses, and that each of them can be converted to a virtual camera or a projector having radially-symmetric pixel-wise focal lengths (Figure 2(b)).

Based on this new camera model, this paper shows that we can realize a linear calibration of projectors and cameras in water, and can realize a practical 3D capture of underwater objects such as swimming fish.

Notice that earlier versions of this paper are partially reported in [11] and [12]. The contribution of this paper is twofold. We evaluate the accuracy of our underwater

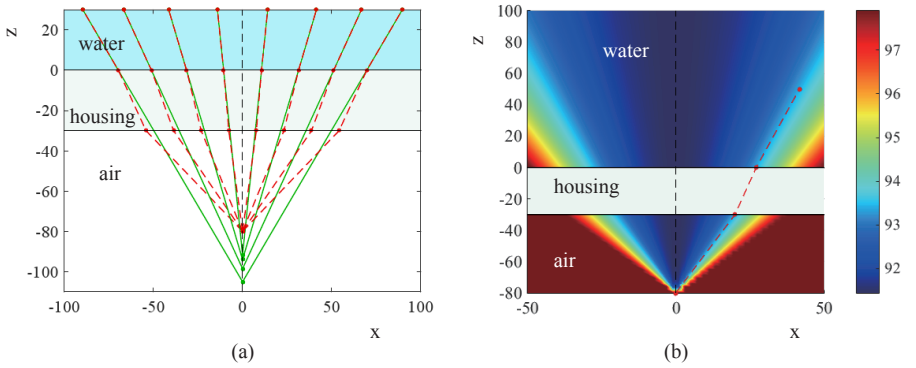


Figure 2: Refraction caused by flat housing. (a) Refraction process (red dashed lines). Ignoring the refraction results in a caustic structure of the rays originating from optical centers on a line (green lines). (b) Distribution of the pixel-wise focal length showing a radially-symmetric structure about the housing normal. The color encodes the virtual focal length for each point (Figure 6).

projector-camera system using a real dataset, and also provide a practical system that realizes underwater 3D capture by active stereo.

2. Related Works

Computer vision techniques has been realized successful image-based 3D shape acquisition of various kind of objects in the air, such as humans[2] and buildings[13, 14], and in particular algorithms utilizing active illuminations are known to be a practical solution for real environments[15, 16].

Towards realizing 3D shape estimation in water, many studies have been proposed for underwater vision[17, 18, 19, 20, 21, 22, 10, 23, 24]. Most of them do not explicitly model refractions by housings because such refractive distortions can be compensated by using dome-shaped housings. This is a straightforward solution, but dome housings require the projection center physically coincide with the dome center[23]. Flat surface housings are also popular because of the cost, and also because of the fact that underwater cameras with flat surface housings is equivalent to capturing objects in a water tank via its flat surface. In this paper, we focus on the case of flat housings.

In the context of refraction modeling, Agrawal *et al*. [25] have proposed a novel calibration technique based on the axial camera model which estimates the exact model parameters of the refraction such as the thickness and refractive indices of each media. By knowing these parameters, Snell's law allows computing the light path passing through the flat housing analytically. However, the computation of forward (3D-to-2D) projection requires solving a 12th degree equation for each projection[25], hence can be intractable if used for 3D sensing in water in practice.

Another approach to handle projection with refraction is use of the *raxel* camera model[26, 10, 27] that explicitly associates a ray in water for each camera pixel. They

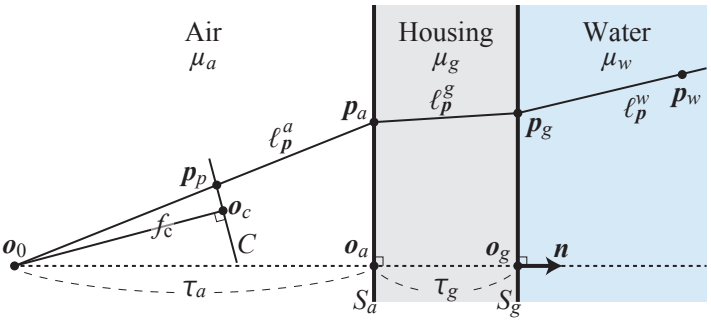


Figure 3: Measurement model of a single camera

employed a two-plane method which learns mappings between two planes in water and their projections in order to represent each 3D ray in water by connecting two distinctive points on the planes corresponding to a single 2D pixel. While this method can be applied for arbitrary housings without explicit housing-to-camera calibrations, this model does not provide a practical solution for the forward (3D-to-2D) projection.

In the context of 3D measurement for underwater object, Treibitz *et al*. have proposed a method that estimates underwater object sizes, under assumption that the camera, the flat housing and the object are all parallel[8, 9]. Sedlazeck and Koch have proposed a virtual camera model of which camera center forms the 3D caustic[28]. They have proposed an underwater structure-from-motion (SfM) technique which accounts for the refraction caused by a flat housing in computing the reprojection error minimization. However SfM requires rich textures on the target in general while underwater objects are not always well-textured.

Another underwater 3D shape reconstruction based on shape-from-silhouette (SfS) has been proposed in [29]. They avoid computing forward (3D-to-2D) projections, and implemented a real-time image-based SfS for underwater object. This method can compute the visual hull of the object as a reasonable initial guess, but their method cannot provide further refinement based on texture matching.

Compared with these studies we propose a new virtual camera model to realize an efficient forward projection and an underwater projector-camera calibration to capture poorly-textured objects in water in practice.

3. Measurement Model

3.1. Underwater Camera

Figure 3 illustrates the measurement model of a single underwater camera. A pinhole camera C at $\boldsymbol{o}_0$ observes the underwater scene via a flat housing surface of τ_g thick at τ_a distance from $\boldsymbol{o}_0$. A point $\boldsymbol{p}_w$ in water is projected to $\boldsymbol{p}_p$ and $\boldsymbol{o}_0$ along the segment $\ell_{\boldsymbol{p}}^w$-$\ell_{\boldsymbol{p}}^g$-$\ell_{\boldsymbol{p}}^a$. The intersections on the housing boundary S_g and S_a are denoted by the point $\boldsymbol{p}_g$ and the point $\boldsymbol{p}_a$ respectively.

Here we assume the two surfaces S_a and S_g of the housing are flat and parallel, but the camera is not necessarily front-parallel to them. Instead, we employ the axial

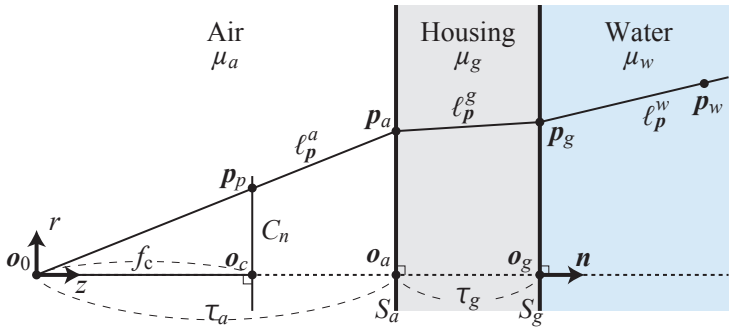


Figure 4: Axial measurement model

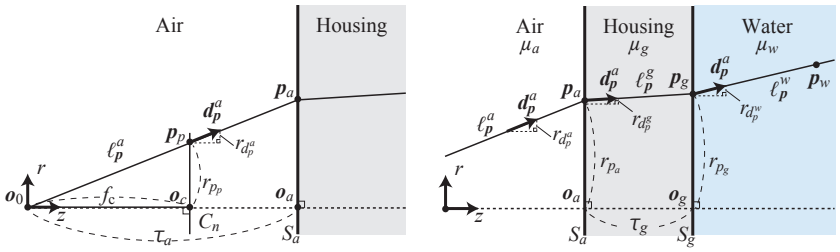


Figure 5: Light path in Axial measurement model. Left : $\ell_{\boldsymbol{p}}^a$ is derived from the observed point $\boldsymbol{p}_p$. Right : The light path $\ell_{\boldsymbol{p}}^a - \ell_{\boldsymbol{p}}^g - \ell_{\boldsymbol{p}}^w$ is derived from $\ell_{\boldsymbol{p}}^a$ with Snell's law.

camera concept proposed by Agrawal *et al*. [25] to simplify this model without loss of generality.

Consider a virtual camera C_n such that its projection center is placed at $\boldsymbol{o}_0$ and its optical axis is directed along the normal vector $\boldsymbol{n}$ of the flat housing (Figure 4). Also let C_n and C share a same intrinsic parameter A calibrated beforehand. By using the calibration proposed by [25], the poses of C and C_n w.r.t. the housing surface can be estimated explicitly. Hence the mapping from a pixel of C to a corresponding pixel of C_n sharing a same $\ell_{\boldsymbol{p}}^a$ is given by a homography matrix trivially.

Since this homography is bijective, we can use C_n instead of C without loss of generality. In addition, the light paths described using C_n have a radially symmetric structure about the Z-axis by definition. That is, $\ell_{\boldsymbol{p}}^w$-$\ell_{\boldsymbol{p}}^g$-$\ell_{\boldsymbol{p}}^a$ for each pixel of C_n is always on a single plane of refraction, and they are identical to each other for all the pixels sharing a same distance r from the image center $\boldsymbol{o}_c$ as well as $\boldsymbol{o}_a$ and $\boldsymbol{o}_g$ consequently (Figure 4). Hence we employ the $(r, z)^\top$ coordinate system hereafter.

Let r_α and z_α denote the r and z elements of vector $\boldsymbol{\alpha}$ in general. For example, point $\boldsymbol{p}_p$ is described as $\boldsymbol{p}_p = (r_{p_p}, z_{p_p})^\top$. Also let $\boldsymbol{d}_p^X = (r_{d_p^X}, z_{d_p^X})^\top$ denote the direction vector of line ℓ_p^X towards the water from the camera, where X is each medium (Figure 5). Snell's law is expressed as $\mu_a r_{d_p^a} = \mu_g r_{d_p^g} = \mu_w r_{d_p^w}$, where μ_a, μ_g, and μ_w are the refractive indices of the air, housing and water. Using Snell's law the light

path $\ell^a_{\boldsymbol{p}} - \ell^g_{\boldsymbol{p}} - \ell^w_{\boldsymbol{p}}$ is given as

$$\boldsymbol{d}^a_p = \left(r_{p_p} / \sqrt{r^2_{p_p} + f^2_c},\ f_c / \sqrt{r^2_{p_p} + f^2_c} \right)^\top, \tag{1}$$

$$\boldsymbol{p}_a = \frac{\tau_a}{f_c} \boldsymbol{d}^a_p, \tag{2}$$

$$\boldsymbol{d}^g_p = \left(\frac{\mu_a}{\mu_g} r_{d^a_p},\ \sqrt{1 - r^2_{d^g_p}} \right)^\top, \tag{3}$$

$$\boldsymbol{p}_g = \boldsymbol{p}_a + \frac{\tau_g}{z_{d^g_p}} \boldsymbol{d}^g_p, \tag{4}$$

$$\boldsymbol{d}^w_p = \left(\frac{\mu_g}{\mu_w} r_{d^g_p},\ \sqrt{1 - r^2_{d^w_p}} \right)^\top, \tag{5}$$

where f_c is the focal length of the camera. These equations allow computing direction of a ray in water $\boldsymbol{d}^w_p$ and a position $\boldsymbol{p}_g$, given a pixel $\boldsymbol{p}_p$ on the image plane. Similarly, computing $\boldsymbol{p}_p$ from $\boldsymbol{d}^w_p$ can be done by applying Snell's law inversely.

This suggests that knowing the correct direction of the projection is crucial in computing the projection of a point $\boldsymbol{p}_w$ in water. If $\boldsymbol{d}^w_p$ is available, Snell's law simply provides the analytical solution to find $\boldsymbol{p}_g$, $\boldsymbol{p}_a$, and $\boldsymbol{p}_p$. Otherwise, *i.e.*, if $\boldsymbol{d}^w_p$ is not given, it requires solving a 12th degree equation, and becomes a time-consuming process[25].

To realize a practical underwater image-based 3D acquisition, we propose a new approach that exploits the radial structure of the rays in C_n based on the raxel camera model[26].

3.2. Underwater Projector

Based on the principle of reversibility of light, an underwater projector is equivalent to an underwater camera, and hence we can model the rays emitted by the projector using the measurement model for the underwater camera (Section 3.1). Hence we first introduce our model for the camera, and then introduce a projector-specific calibration process later.

4. The Pixel-wise Varifocal Camera Model

Suppose all the model parameters in the previous section including the homography between C and C_n have been calibrated beforehand by a conventional method[25]. The goal of this section is to introduce a new virtual camera model which realizes a simple and efficient computation scheme of the projections in water by compiling the calibrated parameters into another representation.

To this end, we introduce a new virtual camera model named *pixel-wise varifocal camera model (PVCM)* defined as follows (Figure 6).

- The virtual image screen coincides with the housing surface S_g. The virtual pixel $\boldsymbol{p}_g$ is associated with a real pixel $\boldsymbol{p}_p$ of C by Eq (4) and the homography between C and C_n.
- The virtual optical axis (Z-axis) is identical to the housing normal $\boldsymbol{n}$.

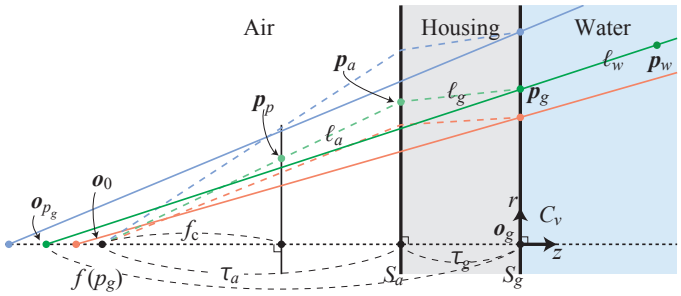


Figure 6: Pixel-wise varifocal camera model. The dashed lines illustrate the correct refractive paths while the straight lines illustrate the perspective projections in three random samples (red, green, blue).

Table 1: Pixel-ray mapping for general raxel model.

pixel	$\boldsymbol{p}_{p0}$	$\boldsymbol{p}_{p1}$	$\cdots$	$\boldsymbol{p}_{pN}$
ray	$(\boldsymbol{p}_{g0}, \boldsymbol{d}^w_{p0})$	$(\boldsymbol{p}_{g1}, \boldsymbol{d}^w_{p1})$	$\cdots$	$(\boldsymbol{p}_{gN}, \boldsymbol{d}^w_{pN})$

- The virtual pixel $\boldsymbol{p}_g$ has a pixel-wise projection center $\boldsymbol{o}_{p_g}$ defined simply by connecting ℓ_w to the virtual optical axis (Figure 6, the green straight line). The distance between the pixel-wise projection center $\boldsymbol{o}_{p_g}$ and the virtual image screen S_g is denoted as the *pixel-wise focal length* $f(\boldsymbol{p}_g)$.

As shown in Figure 6, the green dashed line is the actual path of the measurement model in Section 3 satisfying Snell's law. If we ignore the refraction and project it perspectively, the pixel-wise projection center moves from $\boldsymbol{o}_0$ to $\boldsymbol{o}_{p_g}$ along Z-axis according to the position of each pixel $\boldsymbol{p}_g$ on S_g by definition.

Here the direction of the ray ℓ_w is parameterized by the virtual focal length $f(\boldsymbol{p}_g)$, and the axial structure suggests that the virtual focal length depends only on the radial distance $r_{\boldsymbol{p}_g}$ from the virtual image center $\boldsymbol{o}_g$ to $\boldsymbol{p}_g$ (Figure 6). That is, $f(\boldsymbol{p}_g) = f(r_{\boldsymbol{p}_g})$. This fact indicates that a simple $r_{\boldsymbol{p}_g}$-$f(r_{\boldsymbol{p}_g})$ mapping (Table 2) is sufficient to describe the 3D ray ℓ_w in water emitted from the pixel $\boldsymbol{p}_g$.

This representation is a kind of the raxel model[26] that associates a 3D line on a per-pixel basis, but has two advantages compared with the general raxel representation. As shown in Table 1, the general raxel model associates a fully-described 3D ray for each pixel. Compared with our model (Table 2) storing focal lengths for each radial distance in a 1D array, it requires a quadric order of memory footprint than ours. Also, our representation elucidates an ordered structure of the virtual focal lengths that benefits for an efficient forward projection computation while the general raxel representation cannot exploit such structure (Section 4.3).

4.1. The Pixel-wise Focal Length

Given a pixel $\boldsymbol{p}_g = (r_{p_g}, \tau_a + \tau_g)^\top$ of the virtual camera C_v as shown in Table 2, consider representing the ray $\ell^w_{\boldsymbol{p}}$ incident at $\boldsymbol{p}_g$ as if C_v is a pinhole camera and its projection center is on the Z-axis. For this description, let us use $\boldsymbol{q}_X$ and $\ell^X_{\boldsymbol{q}}$ expression instead of $\boldsymbol{p}_X$ and $\ell^X_{\boldsymbol{p}}$ when we set the origin as $\boldsymbol{o}_g(0,0)$. Obviously its projection

Table 2: Pixel-ray mapping for our virtual camera model.

pixel	$r_{p_{g0}}$	$r_{p_{g1}}$	$\cdots$	$r_{p_{gn}}$
ray	$f(r_{\boldsymbol{p}_{g0}})$	$f(r_{\boldsymbol{p}_{g1}})$	$\cdots$	$f(r_{\boldsymbol{p}_{gn}})$

center $\boldsymbol{o}_{q_g} = (0, -f(\boldsymbol{q}_g))^\top$ is given as the intersection of the Z-axis and the line $\ell^w_{\boldsymbol{q}}$ as illustrated in Figure 6. Hence by solving $\boldsymbol{o}_{q_g} = t\boldsymbol{d}^w_q + \boldsymbol{q}_g$ using Eq (5), we have

$$\begin{pmatrix} 0 \\ -f(\boldsymbol{q}_g) \end{pmatrix} = t \begin{pmatrix} \frac{\mu_g}{\mu_w} r_{d^g_q} \\ \sqrt{1 - r^2_{d^w_q}} \end{pmatrix} + \begin{pmatrix} r_{q_g} \\ 0 \end{pmatrix}, \tag{6}$$

$$t = -\frac{\mu_w}{\mu_g} \frac{r_{q_g}}{r_{d^g_q}}, \tag{7}$$

$$f(\boldsymbol{q}_g) = \frac{\mu_w}{\mu_g} \frac{r_{q_g}}{r_{d^g_q}} \sqrt{1 - r^2_{d^w_q}}, \tag{8}$$

$$= \frac{\mu_w}{\mu_g} \frac{r_{q_g}}{r_{d^g_q}} \sqrt{1 - (\frac{\mu_g}{\mu_w} r_{d^g_q})^2}, \tag{9}$$

$$= \frac{\mu_w}{\mu_a} \frac{r_{q_g}}{r_{d^a_q}} \sqrt{1 - (\frac{\mu_a}{\mu_w} r_{d^a_q})^2}. \tag{10}$$

Once obtained $f(\boldsymbol{q}_g)$ for each radial distance of the virtual camera C_v, we can compute $\ell^w_{\boldsymbol{q}}$ for each $r_{\boldsymbol{q}_g}$ without tracing the refraction inside the housing, and can compute the forward (3D-to-2D) and backward (2D-to-3D) projection computations as follows.

4.2. Backward Projection using Pixel-wise Focal Length

The backward projection using our varifocal camera model can be done straightforwardly. If a point $\boldsymbol{q}_g$ on S_g is the projection of a 3D point $\boldsymbol{q}_w$ in water, then the viewing ray $\ell^w_{\boldsymbol{q}}$ connecting $\boldsymbol{q}_g$ and $\boldsymbol{q}_w$ is given as

$$\ell^w_{\boldsymbol{q}} : (0, -f(\boldsymbol{q}_g))^\top + t\boldsymbol{d}^w_p, \tag{11}$$

using a scale parameter t representing the depth. Notice that the mapping in Table 2 provides $f(\boldsymbol{q}_g)$ for $\boldsymbol{q}_g$, and $\boldsymbol{d}^w_p$ by connecting $\boldsymbol{o}_{q_g} = (0, -f(\boldsymbol{q}_g))^\top$ and $\boldsymbol{q}_g$.

4.3. Forward Projection using Pixel-wise Focal Length

Consider a 3D point $\boldsymbol{q}_w$ in water, and a 3D line ℓ_q passing through $\boldsymbol{q}_w$ and intersecting with S_g and Z-axis at $\boldsymbol{q}_g$ and $\boldsymbol{o}_{q_g} = (0, -f_{q_g})^\top$ as illustrated in Figure 7. Then the following proposition holds.

Proposition 4.1. *$f(\boldsymbol{q}_g)$, the pixel-wise focal length stored at $\boldsymbol{q}_g$, is equal to f_{q_g} if and only if ℓ_q is identical to the ray imaged by the camera C.*

Proof. The definition of the varifocal camera model ensures that the line passing through $\boldsymbol{q}_g$ on S_g and $\boldsymbol{o}_{q_g} = (0, -f(\boldsymbol{q}_g))^\top$ represents a 3D ray which is projected onto a single pixel of the camera C. On the other hand, the principle of the reversibility of light and

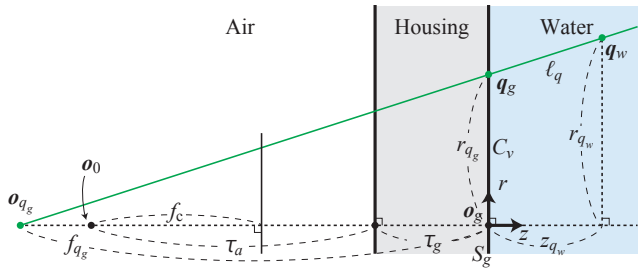


Figure 7: Forward projection by quadratically and globally convergent optimization. 3D line ℓ_q is uniquely defined by a 3D point $\boldsymbol{q}_w$ once all the model parameters $\boldsymbol{o}_0, \boldsymbol{o}_g, \tau_a, \tau_g, \mu_a, \mu_g, \mu_w$ are calibrated.

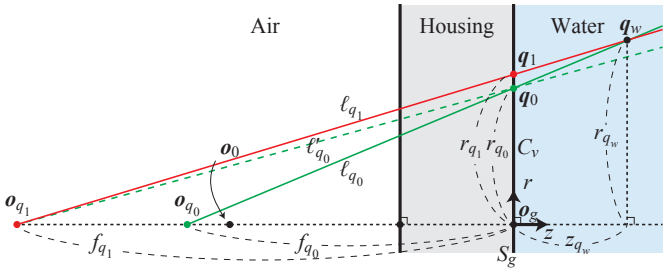


Figure 8: Forward projection by a recurrence relation. The green line illustrates the initial guess. The virtual pixel $\boldsymbol{q}_0$ has the corresponding focal length f_{q_1} and then the line is updated for $\boldsymbol{o}_{q_1}$ (red line). As a result, this recurrence relation gradually converges to the correct projection point using calibrated relationships $\boldsymbol{q} \mapsto f_q$ for each virtual pixel $\boldsymbol{q}$.

the definition of the pinhole imaging ensure that there exists only a ray incident at $\boldsymbol{q}_g$ which can be imaged by the camera C. Hence it is only the case making $f(\boldsymbol{q}_g) = f_{q_g}$ that ℓ_q is identical to the ray imaged by C. □

This proposition indicates that we can obtain the projection of $\boldsymbol{q}_w$ on S_g, the image screen of the varifocal camera, by seeking $\boldsymbol{q}_g$ which minimizes the difference between f_{q_g}, the focal length used to project $\boldsymbol{q}_w$ perspectively along ℓ_w, and $f(\boldsymbol{q}_g)$, the focal length stored at the intersection $\boldsymbol{q}_g$. To this end, we first introduce a simple method using a recurrence relation, and then introduce a faster algorithm based on the Newton's method utilizing the recurrence relation.

Forward Projection by a Recurrence Relation. As illustrated in Figure 8, suppose the 3D point $\boldsymbol{q}_w$ in question is first projected perspectively to $\boldsymbol{q}_0$ on S_g, the virtual screen of C_v, by using an initial (or tentative) focal length f_{q_0} (the green line). By the definition of the pixel-wise varifocal model, the pixel $\boldsymbol{q}_0$ has the correct focal length $f_{q_1} = f(\boldsymbol{q}_0)$ given as shown in Table 2. That is, $\boldsymbol{o}_{q_1} = (0, -f_{q_1})^\top$ is the correct virtual projection center for $\boldsymbol{q}_0$ instead of $\boldsymbol{o}_{q_0} = (0, -f_{q_0})^\top$. By iteratively applying perspective projections using $(0, -f_{q_0}), (0, -f_{q_1}), \ldots$, we have

$$r_{q_{k+1}} = \frac{r_{q_w} f(\boldsymbol{q}_k)}{(z_{q_w} + f(\boldsymbol{q}_k))}. \tag{12}$$

By Snell's law and the fact $r_{\boldsymbol{q}_k} > r_{\boldsymbol{q}_{k'}} \Leftrightarrow f(\boldsymbol{q}_k) > f(\boldsymbol{q}_{k'})$ from Eqs (1), (2), (3), (4) and (10), the following monotonicity conditions hold:

$$r_{q_1} > r_{q_0} \Rightarrow r_{q_{k+1}} \geq r_{q_k}, \quad r_{q_1} < r_{q_0} \Rightarrow r_{q_{k+1}} \leq r_{q_k}. \tag{13}$$

Also, the definition of the pixel-wise varifocal model ensures

$$r_{q_{k+1}} = r_{q_k} \Leftrightarrow f(\boldsymbol{q}_{k+1}) = f_{q_k}, \tag{14}$$

and, since $\mu_a < \mu_g$ and $\mu_a < \mu_w$,

$$\tau_a + \tau_g \leq {}^{\forall} f_{q_k}. \tag{15}$$

Since Proposition 4.1 ensures that there exists only one r_{q_k} which satisfies Eq (14), starting the recurrence from $f_{q_0} = \tau_a + \tau_g$ always converges to the correct value satisfying Eq (14) as shown in Figure 11 later.

However, the rate of the convergence becomes slower and slower by iteration, because the lines by $\boldsymbol{o}_{q_k}$ and $\boldsymbol{o}_{q_{k+1}}$ (the green and the red lines of Figure 8) become nearly parallel. To overcome this difficulty, we propose a method based on the Newton's algorithm which utilizes this recurrence relation.

Forward Projection by a Quadratically and Globally Convergent Optimization. We can describe the 3D point $\boldsymbol{q}_w$ in question as

$$r_{q_w} = \frac{r_{d_p^a}}{z_{d_p^a}}\tau_a + \frac{r_{d_p^g}}{z_{d_p^g}}\tau_g + \frac{r_{d_p^w}}{z_{d_p^w}} z_{q_w}. \tag{16}$$

By using Eqs (1), (3) and (5), we can rewrite this as

$$r_{q_w} = \frac{\mu_w r_{d_p^w} \tau_a}{\sqrt{\mu_a^2 - \mu_w^2 r_{d_p^w}^2}} + \frac{\mu_w r_{d_p^w} \tau_g}{\sqrt{\mu_g^2 - \mu_w^2 r_{d_p^w}^2}} + \frac{r_{d_p^w} z_{q_w}}{\sqrt{1 - r_{d_p^w}^2}}. \tag{17}$$

From this equation, we can formulate the forward projection computation as a problem finding $r_{d_p^w}$ which makes the following $E(r_{d_p^w})$ be zero.

$$E(r_{d_p^w}) = r_q - \frac{r_{d_p^w} z_q}{\sqrt{1 - r_{d_p^w}^2}} - \frac{\mu_w r_{d_p^w} \tau_a}{\sqrt{\mu_a^2 - \mu_w^2 r_{d_p^w}^2}} - \frac{\mu_w r_{d_p^w} \tau_g}{\sqrt{\mu_g^2 - \mu_w^2 r_{d_p^w}^2}}. \tag{18}$$

The best $r_{d_p^w}$ which makes $E(r_{d_p^w}) = 0$ can be computed by the Newton's method efficiently, and moreover, it converges globally regardless of the initial value.

Proof. The theorem on Newton's method for a convex function ensures that if the objective function is twice continuously differentiable, increasing, convex and has a zero, then the zero is unique, and the Newton's method will converge to it from any initial value[30].

In case of Eq (18), the first and the second derivatives of $E(r_{d_p^w})$ are given as

$$\begin{aligned}\frac{dE(r_{d_p^w})}{dr_{d_p^w}} =& \frac{z_q}{E_1} + \frac{z_q\, r_{d_p^w}^2}{E_1^3} + \frac{\tau_g \mu_w}{E_2} \\ &+ \frac{\tau_g\, \mu_w^3\, r_{d_p^w}^2}{E_2^3} + \frac{\tau_a \mu_w}{E_3} + \frac{\tau_a\, \mu_w^3\, r_{d_p^w}^2}{E_3^3}, \end{aligned} \tag{19}$$

$$\begin{aligned}\frac{d^2E}{dr_{d_p^w}^2} =& \frac{3\, z_q\, r_{d_p^w}}{E_1^3} + \frac{3\, z_q\, r_{d_p^w}^3}{E_1^5} + \frac{3\, \tau_g\, \mu_w^3\, r_{d_p^w}}{E_2^3} \\ &+ \frac{3\, \tau_g\, \mu_w^5\, r_{d_p^w}^3}{E_2^5} + \frac{3\, \tau_a\, \mu_w^3\, r_{d_p^w}}{E_3^3} + \frac{3\, \tau_a\, \mu_w^5\, r_{d_p^w}^3}{E_3^5}, \end{aligned} \tag{20}$$

where $E_1 = 1/\sqrt{1 - r_{d_p^w}^2}$, $E_2 = 1/\sqrt{{\mu_g}^2 - {\mu_w}^2\, r_{d_p^w}^2}$, and $E_3 = 1/\sqrt{{\mu_a}^2 - {\mu_w}^2\, r_{d_p^w}^2}$.

Since $r_{d_p^w}$ is non-negative by definition, $\frac{d^2E(r_{d_p^w})}{dr_{d_p^w}^2} \geq 0$ holds and $E(r_{d_p^w})$ is a convex function. Obviously $E(r_{d_p^w})$ is twice continuously differentiable, increasing, and has a zero for $r_{d_p^w} \geq 0$, then the Newton's method converges globally. □

In addition, while this global convergence allows us to start finding $r_{d_p^w}$ from any value in $[0, \mu_w/\mu_g]$, notice that the recurrence relation of Eq (12) can provide a reasonable initial guess of $r_{d_p^w}$ by projecting first by a tentative virtual focal length with a smaller computational cost as shown in Table 3 later.

5. Camera and Projector Calibration in Water

Up to this point, we introduced our underwater camera model that allows a compact and efficient representation of 3D-to-2D forward projection. This section introduces a practical method for calibrating underwater cameras by PVCM (pixel-wise varifocal camera model) and underwater projectors by PVPM (pixel-wise varifocal projector model) to realize our underwater active stereo system.

Notice that our calibration requires reference objects of known geometry in water, such as a chessboard. That is, as long as such reference objects are available, our calibration can be conducted in water.

5.1. Camera Calibration in Water

Modeling a single underwater camera by PVCM itself is a trivial process. By calibrating the relative pose of the camera w.r.t. the housing based on a conventional method using a reference object in water[25], we can compute the table associating the virtual focal length and the radial distance (Table 2) as described earlier.

Based on this single PVCM calibration, we propose a linear extrinsic calibration for multiple underwater cameras.

5.1.1. Linear Extrinsic Calibration of Underwater Cameras

Suppose we have two pixel-wise varifocal cameras C_v and C'_v. The goal of the extrinsic calibration is to estimate the relative pose R, T of these cameras from a set of corresponding points in their images. Since our virtual camera is a kind of axial models, its extrinsic calibration can be seen as a special form of the one for axial cameras[31].

Given a pixel $\boldsymbol{p}_{p'}$ in the real image, we can obtain the corresponding position $\boldsymbol{p}_g$ on S_g without loss of generality as illustrated by Figures 3 and 6. Therefore, given a pair of corresponding points, we can represent the 3D point in water as

$$\begin{aligned} \boldsymbol{q}_w = t_{q_w}\boldsymbol{d}^w_p + \boldsymbol{q}_g &= \lambda_{q_w}\boldsymbol{d}^w_p + \boldsymbol{o}_{q_g}, \\ &= R(\lambda'_{q_w}\boldsymbol{d}'^w_p + \boldsymbol{o}'_{q_g}) + T, \\ \Leftrightarrow \lambda_{q_w}\boldsymbol{d}^w_p - \lambda'_{q_w}R\boldsymbol{d}'^w_p &= R\boldsymbol{o}'_{q_g} + T - \boldsymbol{o}_{q_g}, \end{aligned} \tag{21}$$

where λ_{q_w} and λ'_{q_w} denote the unknown depths of the 3D point from $\boldsymbol{o}_{q_g}$ and $\boldsymbol{o}'_{q_g}$.

This equation indicates that $\boldsymbol{d}^w_p$, $R\boldsymbol{d}'^w_p$ and $R\boldsymbol{o}'_{q_g} + T - \boldsymbol{o}_{q_g}$ are on a single plane. In other words, they satisfy:

$$\boldsymbol{d}^{\boldsymbol{w}\top}_{\boldsymbol{p}}\left(\left(R\boldsymbol{o}'_{q_g} + T - \boldsymbol{o}_{q_g}\right) \times \left(R\boldsymbol{d}'^w_p\right)\right) = 0. \tag{22}$$

By rewriting this as an element-wise formula, we have

$$\begin{aligned} & l_w^\top E_v l'_w = 0, \\ & l_w = \begin{pmatrix} x_{d^w_p} & y_{d^w_p} & z_{d^w_p} & f_{q_g}x_{d^w_p} & f_{q_g}y_{d^w_p} \end{pmatrix}^\top, \\ & l'_w = \begin{pmatrix} x'_{d^w_p} & y'_{d^w_p} & z'_{d^w_p} & f'_{q_g}x'_{d^w_p} & f'_{q_g}y'_{d^w_p} \end{pmatrix}^\top, \\ & E_v = \begin{pmatrix} r_{31}y_t - r_{21}z_t & r_{32}y_t - r_{22}z_t & r_{33}y_t - r_{23}z_t & -r_{12} & r_{11} \\ r_{11}z_t - r_{31}x_t & r_{12}z_t - r_{32}x_t & r_{13}z_t - r_{33}x_t & -r_{22} & r_{21} \\ r_{21}x_t - r_{11}y_t & r_{22}x_t - r_{12}y_t & r_{23}x_t - r_{13}y_t & -r_{32} & r_{31} \\ -r_{21} & -r_{22} & -r_{23} & 0 & 0 \\ r_{11} & r_{12} & r_{13} & 0 & 0 \end{pmatrix}, \end{aligned} \tag{23}$$

where r_{ij} is the (i, j) element of R and $T = (x_t, y_t, z_t)^\top$. $(x_{d^w_p}, y_{d^w_p}, z_{d^w_p})$ and $(x'_{d^w_p}, y'_{d^w_p}, z'_{d^w_p})$ represent x, y, z elements of $\boldsymbol{d}^w_p$ and $\boldsymbol{d}'^w_p$ expressed in their camera coordinate systems respectively.

Since l and l' are given by each of corresponding pairs, we can linearly estimate 17 unknown elements of E_v up to a scale, by using 16 or more corresponding pairs. Once E_v is given, R and T can be obtained linearly from E_v consequently.

Notice that Eq (22) has a trivial solution $R\boldsymbol{o}'_{q_g} + T - \boldsymbol{o}_{q_g} = 0$. This indicates $R\boldsymbol{o}'_{q_g} + T = \boldsymbol{o}_{q_g}$ and the two virtual optical centers coincide. In this case two lines ℓ_w and ℓ'_w should either be intersecting at the virtual optical center or be identical to each other. The former case suggests that the two lines never intersect in water and violates the assumption where the cameras captured a 3D point in water. The latter case suggests that the two cameras are identical. Hence this trivial solution should be rejected.

Besides, the rotation matrix estimated linearly may not be an $SO(3)$ matrix. To enforce the orthogonality constraint, we modify the matrix as $R' = UV^\top$ where U and V are the left and the right singular matrices of R[32].

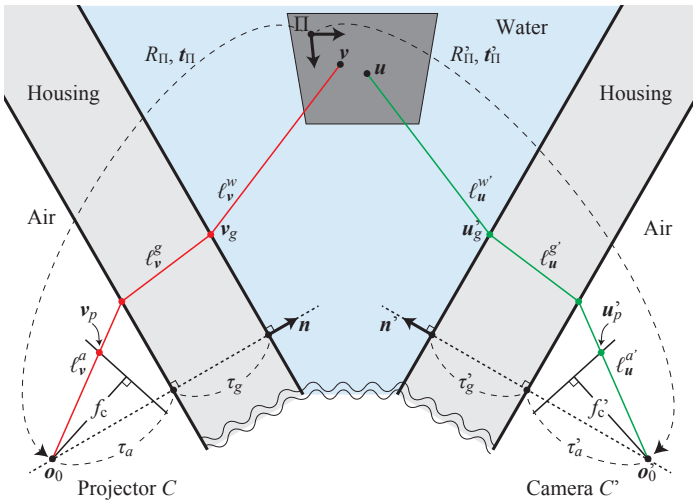


Figure 9: Underwater projector-camera calibration. A projector C and a camera C' observe a point $\boldsymbol{u}$ on a plane Π in water via flat housings.

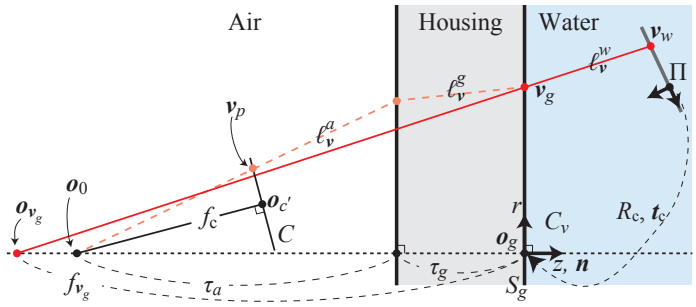


Figure 10: Pixel-wise varifocal camera / projector model

5.2. Calibration of Projector-Camera System in Water

This section describes our underwater projector-camera calibration. Notice that this is originally presented in [12], but we include this in order to keep this paper self-contained.

As illustrated in Figure 9, suppose we have an underwater projector C and an underwater camera C'. Π is a flat calibration panel in water. Here we assume that a 2D pattern U of a known geometry is printed on Π, and the camera C' is calibrated as PVCM ${C_v}'$ beforehand. Notice that we use X' to denote a parameter of PVCM which corresponds to X of PVPM, and $x_{\boldsymbol{X}}$, $y_{\boldsymbol{X}}$, and $z_{\boldsymbol{X}}$ to denote the x, y, and z element of a vector $\boldsymbol{X}$ respectively, and $\boldsymbol{d}_X^Y$ to denote the direction of the line ℓ_X^Y.

Similarly to PVCM, we model the rays emitted by the projector using pixel-wise varifocal lengths. We denote this as pixel-wise varifocal projector model (PVPM) hereafter (Figure 10). This section first introduces a PVPM calibration and then the extrin-

sic calibration of a PVCM-PVPM pair.

Our calibration is based on estimating the 3D geometry of patterns projected onto the plane Π. As done in the air[33], we use a 2D pattern U printed on Π as well as a 2D pattern V projected by C onto Π, to provide a set of 2D-3D correspondences for the projector to calibrate it as a PVPM. Our calibration consists of the following steps.

Step 1. Camera ${C_v}'$ pose estimation w.r.t. Π by capturing pattern U.

Step 2. Estimation of 3D geometry of a pattern V projected by C on Π using ${C_v}'$.

Step 3. PVPM calibration of C_v and its pose estimation w.r.t. Π using 2D-3D correspondence of pattern V.

As a result, both the camera and the projector poses are estimated w.r.t. Π.

In what follows, we denote parameter X in underwater camera coordinate system by ${}^{\{{C_v}'\}}X$ and X in the plane coordinate system by ${}^{\{\Pi\}}X$ in order to clarify the coordinate system.

Step 1. Pose Estimation of PVCM using Planner Pattern in Water. Estimation of the camera pose ${R_c}'$ and ${\boldsymbol{t}_c}'$ w.r.t. Π can be done by using the flat refraction constraint[25]. That is, the direction $\boldsymbol{d}_{\boldsymbol{u}}^{w'}$ of the ray $\ell_{\boldsymbol{u}}^{w'}$ to a known point $\boldsymbol{u}$ of U is identical to the vector from the incident point ${\boldsymbol{u}_g}'$ to ${\boldsymbol{u}_w}'$, where the known point $\boldsymbol{u} = (x_{\boldsymbol{u}}, y_{\boldsymbol{u}}, 0)^\top$ is described as ${\boldsymbol{u}_w}'$ in the coordinate system of PVCM ${C_v}'$ (Figure 9).

$$
\begin{aligned}
& {}^{\{{C_v}'\}}\boldsymbol{d}_{\boldsymbol{u}}^{w'} \times \left({}^{\{{C_v}'\}}{\boldsymbol{u}_w}' - {}^{\{{C_v}'\}}{\boldsymbol{u}_g}' \right) = \boldsymbol{0}, \\
\Leftrightarrow\ & {}^{\{{C_v}'\}}\boldsymbol{d}_{\boldsymbol{u}}^{w'} \times \left(\left({R_c}' {}^{\{\Pi\}}\boldsymbol{u} + {\boldsymbol{t}_c}' \right) - {}^{\{{C_v}'\}}{\boldsymbol{u}_g}' \right) = \boldsymbol{0}, \\
\Leftrightarrow\ & \begin{pmatrix} x_{\boldsymbol{u}}[\boldsymbol{d}_{\boldsymbol{u}}^{w'}]_\times & y_{\boldsymbol{u}}[\boldsymbol{d}_{\boldsymbol{u}}^{w'}]_\times & [\boldsymbol{d}_{\boldsymbol{u}}^{w'}]_\times \end{pmatrix} \begin{pmatrix} {\boldsymbol{r}_{c,1}}' \\ {\boldsymbol{r}_{c,2}}' \\ {\boldsymbol{t}_c}' \end{pmatrix} = [\boldsymbol{d}_{\boldsymbol{u}}^{w'}]_\times {\boldsymbol{u}_g}',
\end{aligned}
\tag{24}
$$

where $\boldsymbol{r}_{X,i}$ denotes the ith column vector of R_X, and $[\boldsymbol{X}]_\times$ denotes the 3×3 skew-symmetric matrix defined by a 3D vector $\boldsymbol{X}$. Since this equation provides three constraints for 9 unknowns ${\boldsymbol{r}_{c,1}}'$, ${\boldsymbol{r}_{c,2}}'$, and ${\boldsymbol{t}_c}'$, we can solve this system of equations linearly by using at least three points. Once ${\boldsymbol{r}_{c,1}}'$ and ${\boldsymbol{r}_{c,2}}'$ are obtained, ${\boldsymbol{r}_{c,3}}'$ is given by their cross product.

Step 2. Estimation of 3D Geometry of Projected Pattern. Suppose the projector C casts a known pattern V onto Π which consists of feature points such that their projections can be identified in the captured image of ${C_v}'$ even under refractions. Let $\boldsymbol{v}$ denote a 3D feature point on Π projected from a pixel $\boldsymbol{v}_p$ of projector C. The goal here is to estimate ${}^{\{\Pi\}}\boldsymbol{v} = (x_{\boldsymbol{v}}, y_{\boldsymbol{v}}, 0)^\top$ from its projection ${\boldsymbol{v}_w}'$ in the camera ${C_v}'$ image in order to establish 2D-3D correspondences between 2D projector pixels $\boldsymbol{v}_p$ and 3D points $\boldsymbol{v}$ on Π.

Since $\boldsymbol{v}$ is on $\ell_{\boldsymbol{v}}^{w'}$, we can represent its 3D position to ${C_v}'$ with a scale parameter ${\lambda_{\boldsymbol{v}}}'$ as

$$
{}^{\{\Pi\}}\boldsymbol{v} = {R_c}'^\top ({}^{\{{C_v}'\}}{\boldsymbol{v}_w}' - {\boldsymbol{t}_c}') = {R_c}'^\top ({\lambda_{\boldsymbol{v}}}' {}^{\{{C_v}'\}}\boldsymbol{d}_{\boldsymbol{v}}^{w'} + {\boldsymbol{o}_{\boldsymbol{v}_g}}' - {\boldsymbol{t}_c}'). \tag{25}
$$

Here we know that $z_{\boldsymbol{v}} = 0$ because of the fact that V is on Π, and it is trivial to determine the other unknown parameters ${\lambda_{\boldsymbol{v}}}'$, $x_{\boldsymbol{v}}$ and $y_{\boldsymbol{v}}$.

Step 3. Calibration of Pixel-wise Varifocal Projector Model using 2D-3D Correspondences. Given a set of correspondences between 2D projector pixels $\boldsymbol{v}_p$ and 3D points $\boldsymbol{v}$ on Π in the previous step, the pose of the real projector R_Π and $\boldsymbol{t}_\Pi$ w.r.t. Π, and its housing parameters can be calibrated by the conventional method[25]. Once obtained these parameters, we can build a table representing the virtual projector focal length as done for PVCM.

Notice that the 3D points $\boldsymbol{v}$ are not necessarily from a single Π. In fact by capturing the panel Π with different poses in water, they can cover a larger area of the scene and contribute to improve the accuracy and robustness of the parameter estimation as pointed out in [25].

6. Underwater Active Stereo

Up to this point, we have introduced our underwater projector-camera calibration that can handle projections in water in a compact and efficient manner. This section introduces our underwater active stereo system that utilizes projectors as reverse cameras as done in the Kinect sensor[34] in order to improve the stereo matching for poorly-textured underwater objects by attaching artificial texture onto the target surface.

Apart from the texture, the main difficulty of underwater stereo is its depth-dependent image distortion caused by refraction. It deforms the epipolar line according to the pixel-wise depth, and invalidates stereo methods that work on 2D image domain by template matching with slanted local supports[35]. Also stereo methods that work on 3D domain by projecting small patches to 2D images[2] can be intractable due to the refraction computation.

In this section we introduce two practical systems that can be realized by our underwater projector-camera calibration and efficient 3D-to-2D projection.

6.1. Structured Lighting in Water

For static objects, our underwater projector-camera calibration allows implementing a structured light technique. Suppose calibrated underwater projectors cast structured light patterns such as Gray codes to the object. By capturing the object using underwater cameras, we can establish dense per-pixel correspondences by decoding the pattern images[33]:

1. Project structured light patterns to the object.
2. Capture the images and establish correspondences between projector and camera pixels.
3. Estimate the depth by triangulating the rays in water computed by PVCM and PVPM.

This provides an accurate 3D geometry since it does not need to *estimate* the correspondences, but requires the object to be kept static.

6.2. Underwater Active Stereo using Random Dot Pattern

As done in the Kinect sensor[34], casting a random dot pattern onto the object can enhance the stereo matching quality for poorly-textured surfaces in practice, and allows a oneshot capture of dynamic object in the scene.

Our underwater projector-camera calibration allows projectors to be used as reverse cameras, and our efficient forward projection allows explicit handling of the refraction by projecting 3D sample points (voxels) to images for photo-consistency evaluation as done in space carving[36]:

1. Capture the background image for each camera without the object beforehand.
2. Capture the foreground image for each camera with the object while projecting a known pseudo-random-dot pattern from projectors.
3. Estimate the object silhouette for each camera by the background subtraction.
4. Apply the shape-from-silhouette[37, 29] to estimate the maximum object volume by the visual hull.
5. Carve photo-inconsistent voxels from the visual hull until all the outmost voxels be photo-consistent.

Notice that projectors contribute to the photo-consistency evaluation in **5.** as reverse cameras by utilizing the knowledge about the projected pattern.

7. Evaluation

7.1. Efficiency of Forward projection

This section evaluates our forward projection computation in terms of efficiency, since the proposed model does not improve the accuracy by definition.

Rate of convergence. To evaluate the rate of convergence of our iterative methods for the forward projection in Section 4.3, Figure 11 shows the projection error E_p against the number of iterations k. By using a synthesized data set, the reprojection error is defined as

$$E_p = |P'(\hat{r}_q) - P'(r_{q_k})|, \tag{26}$$

where $\hat{r}_q$ is the ground-truth and r_{q_k} is the value returned by the algorithm at the k-th iteration in C_v. $P'(r_{q_k})$ denotes the pixel position in the original image of C' corresponding to r_{q_k} in C_v. Notice that $P'(\cdot)$ is employed only for evaluating E_p in pixels, and is not required for the forward projection to C_v.

From these results, we can observe that (1) the rates of convergence of the recurrence relation and the Newton-based ones are linear and quadratic respectively, and (2) our Newton-based algorithm with 3 times iterations achieve a sub-pixel accuracy.

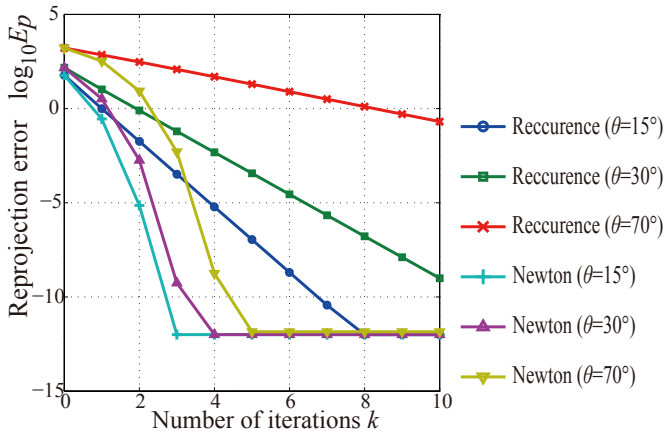


Figure 11: Comparison of the rate of convergence. Notice that errors are lower bounded by 10^{-12}, the default precision of the floating-point computations in our implementation.

Table 3: Average computational costs of single forward projections

	Analytical[25]	By Recurrence	By Newton
Runtime	1.39 msec	0.14 msec	0.27 msec
FLOPS	1512	113	250

Computational efficiency. Table 3 reports computational costs of our methods computing up to the subpixel accuracy and the state-of-the-art solving the 12th degree of equation analytically[25]. They are the average values of 6400 forward projections run in Matlab on an Intel Core-i5 2.5GHz PC. The FLOPS are counted using Lightspeed Matlab toolbox[38].

From these results, we can conclude that our method runs much faster than the analytical method while maintaining the sub-pixel accuracy. In other words, the our method achieves the equivalent accuracy as other methods in practical image-based analysis.

Discussion. The linear rate of convergence by the recurrence relation method and the quadratic rate by the Newton-based method (Section 4.3) shown by Figure 11 do not immediately indicate that the Newton-based method is always the better option, because of the difference on their computation costs for a single iteration reported in Table 3. In particular, if we represent the pixel-wise focal lengths $f(r_g)$ by a LUT of a certain resolution of r_g, then the recurrent computation of Eq (12) can be done just by obtaining the value $f(r_{q_i})$ from it. Hence if the total computation cost is limited, then combining these two methods by updating with the recurrence relation method first and then by switching to the Newton-based one for fine tuning can be a better option.

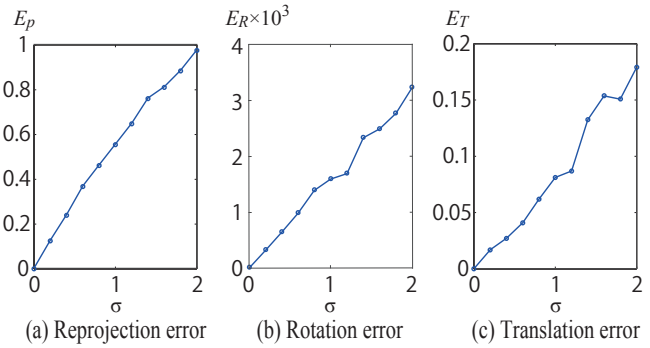


Figure 12: Quantitative evaluations of our underwater projector-camera calibration

7.2. Calibration

7.2.1. Quantitative Evaluation using Synthesized Data

Figure 12 shows calibration errors under different noise levels. Given a set of 160 points synthesized in water, we randomly select 16 points for each trial, and add Gaussian noise with zero-mean and standard deviation $\sigma = 0.1, 0.2, \ldots, 2.0$ to their 2D projections. The three plots report the average errors of 100 trials at each noise level. Here the estimation error of R is described as the quaternion distance to the ground truth, and the estimation error of T is defined as the RMS error normalized by $|T|$. These results indicate that our linear method performs robustly against observation noise. Notice that since the projector serves as a reverse camera, this applies both to projectors and cameras.

7.2.2. Quantitative Evaluation using Real Data

Figure 13 shows our environment. We used four SXGA cameras (Pointgrey CMLN-13S2C-CS) and one 1080p projector (BenQ MH680) around an octagonal water tank of 900 mm diameter. The capture target is a flat panel having two chess patterns on it: a colored pattern in water and a black pattern in the air. Their relative pose is calibrated by capturing them in the air beforehand.

The two cameras $Camera1, Camera2$ and the projector observe the object (colored chess pattern) in water via the flat acrylic tank surface of about 30 mm thick. This is optically-equivalent to having an underwater projector and cameras with flat housings of the same thickness. $Camera3$ and $Camera4$ are used as reference cameras to provide the ground truth of the colored chess-pattern 3D geometry in water by capturing the black chess pattern in the air and their relative pose. The ground truth of the camera poses are calibrated beforehand, by capturing reference objects in the air[39] for evaluation purpose.

Underwater Cameras. Based on the calibration given in Figure 13, Figure 14(a) shows the estimated 3D geometry of 40 chess corners in water on five panels at different distances: the distance between the nearest and the farthest panels was roughly 400 mm.

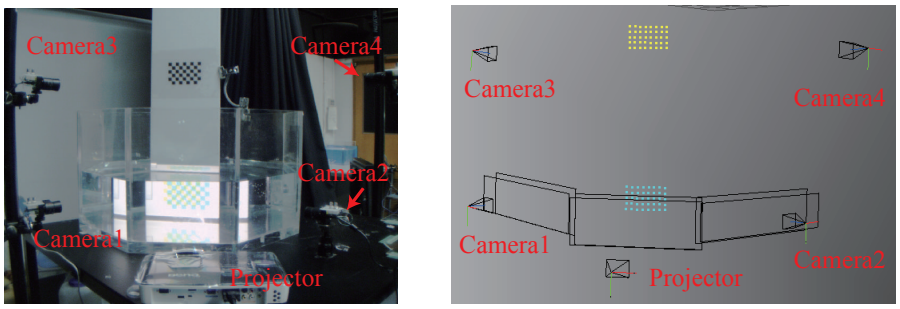


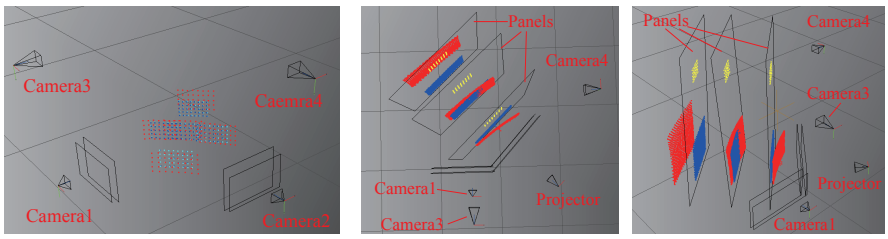
Figure 13: Evaluation environment of underwater projector-camera system. (a) Two cameras and one projector observing the target (colored chess pattern) in water via a flat housing, and two cameras capturing the reference object (black chess pattern) in the air to provide the ground truth position of the colored chess pattern based on their relative posture calibrated beforehand. (b) Calibration result of evaluation system. Camera1 and Camera2 define our underwater camera system. Camera1 and Projector define our underwater projector-camera system. Camera1, Camera2, and Projector are calibrated only by detecting the colored chess pattern (cyan points) from their images.

The blue points are the ground truth calculated by the cameras in the air ($Camera3$, $Camera4$). The cyan ones are the points by our underwater camera system ($Camera1$, $Camera2$). The average error of these 200 points was 2.43 mm. The red ones are points calculated by assuming the perspective projection without refraction, and its average error was 31.11 mm.

Underwater Projector and Camera. Figures 14(b) and (c) show the estimated chess corner positions on three panels at every 200 mm. The blue dots denote the points by our underwater projector and camera system ($Camera1$, $Projector$) using the structured-light method (Section 6). The red dots denote the points by assuming the perspective projection without refraction. Notice that we used only $Camera1$ and the projector as a reverse camera in order to evaluate the underwater projector-camera calibration accuracy sorely. These figures qualitatively visualize that our method better reconstructs the 3D points of different distances from the camera and the projector.

The quality of the calibration is assessed by measuring the distance from the panel to the estimated 3D points. Here the geometry of the each panel surface is estimated using the 3D positions of the chess corners in the air captured by $Camera3$ and $Camera4$. The average errors of the blue points on the three panels were, from near to far, 1.90 mm, 1.59 mm, and 4.01 mm respectively. Those of the red ones were 34.36 mm, 9.53 mm, and 89.06 mm respectively.

From these results we can conclude that our method realized a practical underwater projector-camera calibration in a reasonable accuracy for a wide range of distance from the cameras.



(a) (b) (c)

Figure 14: Evaluation result. (a) 3D points estimated by the underwater camera pair. The blue points are ground truth of colored chess patterns provided by capturing black chess patterns by Camera3 and Camera4 in the air. The cyan points are estimated by Camera1 and Camera2 with our refraction modeling, and the red points are estimated by Camera1 and Camera2 with perspective projection without refraction. (b) 3D points estimated by the underwater projector-camera pair. The panel planes are defined by the yellow points calibrated by Camera3 and Camera4 in the air as the ground truth. The blue points are estimated by our underwater projector and camera system. The red points are estimated by assuming perspective projection without refraction. (c) Top view of (b).

7.3. Underwater 3D Shape Estimation

To demonstrate the performance of the proposed camera model, this section shows two 3D reconstruction results: one for an underwater camera system corresponding to Section 5.1, and the other for an underwater projector-camera system corresponding to Section 5.2.

Underwater Camera System. Figure 15 illustrates our underwater camera system. Given 8 images as shown in Figure 16 left and 2.5 mm voxel resolution, a space carving[36] with our refraction modeling returns a 3D shape shown in Figure 16 right. The reconstruction cost about 1 minute in Matlab on an Intel Core-i5 2.5GHz PC. This result demonstrates the validity of our refraction modeling for underwater cameras qualitatively.

Underwater Projector-Camera System. Figure 18 shows a result of our dynamic 3D shape capture of a swimming goldfish, in order to demonstrate the performance of our underwater projector-camera calibration. We used the system as illustrated in Figure 17 and the same PC used above. Each of the projectors casts pattern in different color channels (red and blue) for avoiding interference. The system ran at 15 fps in recording, and took about 30 sec per frame to reconstruct the 3D shape by our underwater space carving using 4 mm voxel resolution.

The three columns in the left of Figure 18 show the captured images, and the three columns in the right show rendered images of the reconstructed 3D shapes by our method and by the conventional space carving with perspective projection[36]. As the left column of the rendered images shows, we can virtually observe the object appearance even from the top-side of the object where the real camera does not exist. This well demonstrates the accuracy of our 3D shape estimation quality. On the other hand,

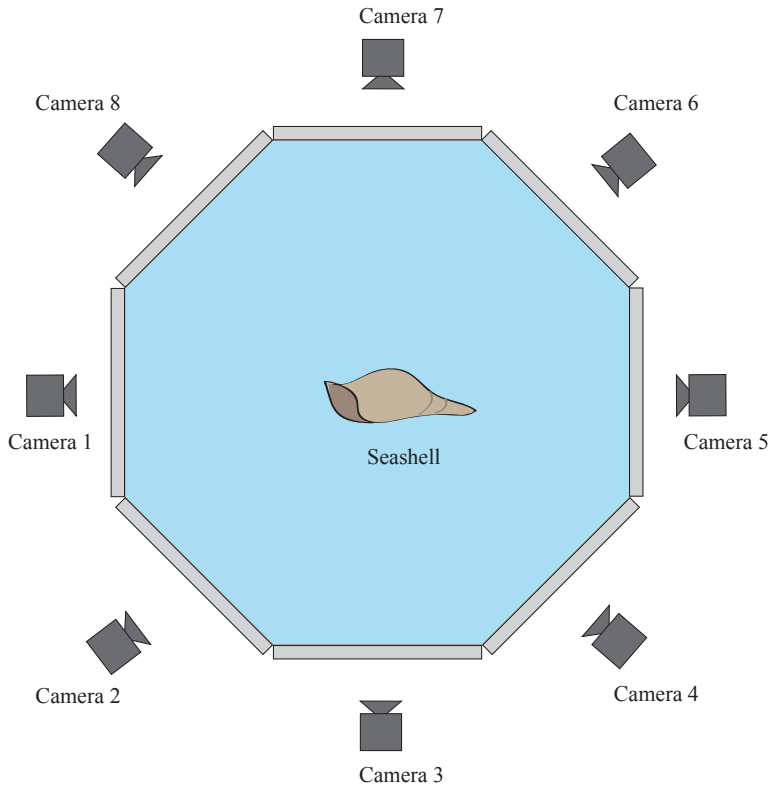


Figure 15: Setup for 3D shape capture of *seashell* with 8 cameras surrounding the same octagonal tank used in Figure 13 without projector.

the conventional space carving cannot produce a comparable result since it ignores refraction and results in poor 3D estimations due to wrong photo-consistency evaluation. These points prove the concept of our image-based full 3D shape reconstruction of underwater dynamic objects.

Notice that the blue colors are a result of projected pattern that can be eliminated if implemented with IR projectors and cameras for example.

8. Conclusion

Towards realizing a 3D shape capture of underwater objects, this paper proposed an underwater projector-camera system which explicitly handles the refraction caused by the flat housings. The evaluations using synthesized and real datasets demonstrated the robustness and accuracy of our underwater projector-camera calibration quantitatively. We demonstrated our proposed algorithm by implementing an underwater multi-view

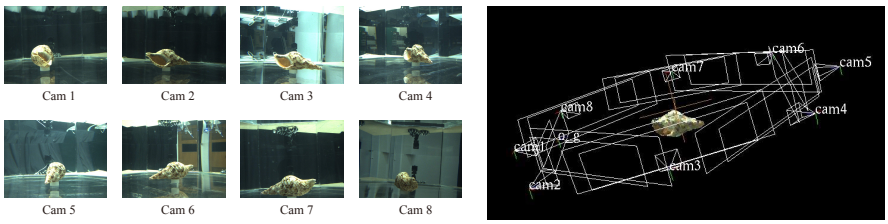


Figure 16: Result of 3D shape estimation of *seashell*.

projector-camera system that captured the dynamic 3D shape of a swimming goldfish successfully.

We believe this work brings us one step closer to realizing a practical 3D sensing of underwater objects. Our future work includes real-time full 3D capture, 3D motion estimation, and extension to semi-transparent targets. Furthermore, extending our method to allow self-calibrating the cameras and projectors without known reference objects in water will help applying our method in environments where calibration parameters change dynamically due to pressure[24].

Acknowledgments

This research is partially supported by JSPS KAKENHI 26240023 and 15J07706.

References

[1] J. Starck, A. Hilton, G. Miller, Volumetric stereo with silhouette and feature constraints, in: Proc. BMVC, 2006, pp. 1189–1198.

[2] Y. Furukawa, J. Ponce, Accurate, dense, and robust multi-view stereopsis, in: Proc. CVPR, 2007, pp. 1–8.

[3] T. Matsuyama, S. Nobuhara, T. Takai, T. Tung, 3D Video and Its Applications, Springer Publishing Company, Incorporated, 2012.

[4] T. B. Moeslund, A. Hilton, V. Krüger, A survey of advances in vision-based human motion capture and analysis, CVIU 104 (2) (2006) 90–126.

[5] L. Ballan, G. M. Cortelazzo, Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes, in: Proc. 3DPVT, 2008.

[6] S. Corazza, L. Mündermann, E. Gambaretto, G. Ferrigno, T. P. Andriacchi, Markerless motion capture through visual hull, articulated icp and subject specific model generation, IJCV 87 (1-2) (2010) 156–169.

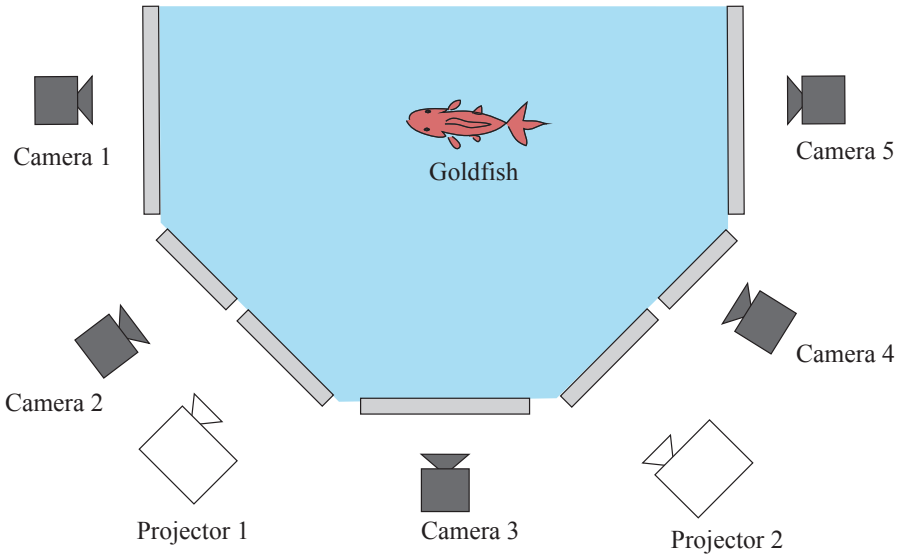


Figure 17: Setup for dynamic 3D shape capture of *goldfish*. The cameras, the projectors, and the tank are the same ones used in Figure 13.

[7] Y. Liu, J. Gall, C. Stoll, Q. Dai, H.-P. Seidel, C. Theobalt, Markerless motion capture of multiple characters using multiview image segmentation, TPAMI 35 (11) (2013) 2720–2735.

[8] T. Treibitz, Y. Y. Schechner, H. Singh, Flat refractive geometry, in: Proc. CVPR, 2008.

[9] T. Treibitz, Y. Y. Schechner, C. Kunz, H. Singh, Flat refractive geometry, TPAMI 34 (2012) 51–65.

[10] S. Narasimhan, S. Nayar, B. Sun, S. Koppal, Structured light in scattering media, in: Proc. ICCV, Vol. I, 2005, pp. 420–427.

[11] R. Kawahara, S. Nobuhara, T. Matsuyama, A pixel-wise varifocal camera model for efficient forward projection and linear extrinsic calibration of underwater cameras with flat housings, in: In Proc. of ICCV 2013 Underwater Vision Workshop, 2013, pp. 819–824.

[12] R. Kawahara, S. Nobuhara, T. Matsuyama, Underwater 3d surface capture using multi-view projectors and cameras with flat housings, IPSJ Transactions on Computer Vision and Applications 6 (2014) 43–47.

[13] K. Ikeuchi, T. Oishi, J. Takamatsu, R. Sagawa, A. Nakazawa, R. Kurazume, K. Nishino, M. Kamakura, Y. Okamoto, The great buddha project: digitally

archiving, restoring, and analyzing cultural heritage objects, IJCV 75 (2007) 189–208.

[14] D. Thomas, A. Sugimoto, A flexible scene representation for 3d reconstruction using an RGB-D camera, in: Proc. ICCV, 2013, pp. 2800–2807.

[15] H. Kawasaki, R. Furukawa, R. Sagawa, Y. Yagi, Dynamic scene shape reconstruction using a single structured light pattern, in: Proc. CVPR, 2008, pp. 1–8.

[16] T. Furuse, S. Hiura, K. Sato, 3-D shape measurement method with modulated slit light robust for interreflection and subsurface scattering, in: Proc. PROCAMS, 2009, pp. 1–2.

[17] Y. Schechner, N. Karpel, Clear underwater vision, in: Proc. CVPR, Vol. 1, 2004, pp. I–536–I–543 Vol.1.

[18] P. Corke, C. Detweiler, M. Dunbabin, M. Hamilton, D. Rus, I. Vasilescu, Experiments with underwater robot localization and tracking, in: Proc. ICRA, 2007, pp. 4556–4561.

[19] D. M. Kocak, F. R. Dalgleish, F. M. Caimi, Y. Y. Schechner, A Focus on Recent Developments and Trends in Underwater Imaging, Marine Technology Society Journal 42 (2008) 52–67. doi:10.4031/002533208786861209.

[20] M. Alterman, Y. Y. Schechner, P. Perona, J. Shamir, Detecting motion through dynamic refraction, TPAMI 35 (1) (2013) 245–251.

[21] M. Alterman, Y. Schechner, Y. Swirski, Triangulation in random refractive distortions, in: Proc. ICCP, 2013, pp. 1–10.

[22] C. Beall, F. Dellaert, I. Mahon, S. Williams, Bundle adjustment in large-scale 3d reconstructions based on underwater robotic surveys, in: IEEE OCEANS, 2011, pp. 1–6.

[23] C. Kunz, H. Singh, Hemispherical refraction and camera calibration in underwater vision, in: OCEANS 2008, 2008, pp. 1–7.

[24] C. Kunz, H. Singh, Stereo self-calibration for seafloor mapping using auvs, in: IEEE/OES Autonomous Underwater Vehicles, 2010, pp. 1–7.

[25] A. Agrawal, S. Ramalingam, Y. Taguchi, V. Chari, A theory of multi-layer flat refractive geometry, in: Proc. CVPR, 2012, pp. 3346–3353.

[26] M. D. Grossberg, S. K. Nayar, The raxel imaging model and ray-based calibration, IJCV 61 (2) (2005) 119–137.

[27] J. Gregson, M. Krimerman, M. B. Hullin, W. Heidrich, Stochastic tomography and its applications in 3d imaging of mixing fluids, in: Proc. ACM SIGGRAPH, 2012, pp. 52:1–52:10.

[28] A. Jordt-Sedlazeck, R. Koch, Refractive structure-from-motion on underwater images, in: Proc. ICCV, 2013, pp. 57–64.

[29] T. Yano, S. Nobuhara, T. Matsuyama, 3d shape from silhouettes in water for online novel-view synthesis, IPSJ Transactions on Computer Vision and Applications.

[30] D. Kincaid, W. Cheney, Numerical Analysis: Mathematics of Scientific Computing, Pure and Applied Undergraduate Texts Series, American Mathematical Society, 2002.

[31] H. Li, R. Hartley, J.-H. Kim, A linear approach to motion estimation using generalized camera models, in: Proc. CVPR, 2008, pp. 1–8.

[32] G. H. Golub, C. F. Van Loan, Matrix Computations (3rd Ed.), Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[33] D. Moreno, G. Taubin, Simple, accurate, and robust projector-camera calibration, in: Proc. 3DIMPVT, 2012, pp. 464–471.

[34] Z. Zhang, Microsoft kinect sensor and its effect, IEEE Multimedia 19 (2) (2012) 4–10.

[35] P. Heise, S. Klose, B. Jensen, A. Knoll, Pm-huber: Patchmatch with huber regularization for stereo matching, in: Proc. ICCV, 2013, pp. 2360–2367.

[36] K. N. Kutulakos, S. M. Seitz, A theory of shape by space carving, in: Proc. ICCV, 1999, pp. 307–314.

[37] A. Laurentini, How far 3D shapes can be understood from 2D silhouettes, TPAMI 17 (2) (1995) 188–195.

[38] T. Minka, Lightspeed matlab toolbox, `http://research.microsoft.com/en-us/um/people/minka/software/lightspeed/`.

[39] Z. Zhang, A flexible new technique for camera calibration, TPAMI 22 (11) (2000) 1330–1334.

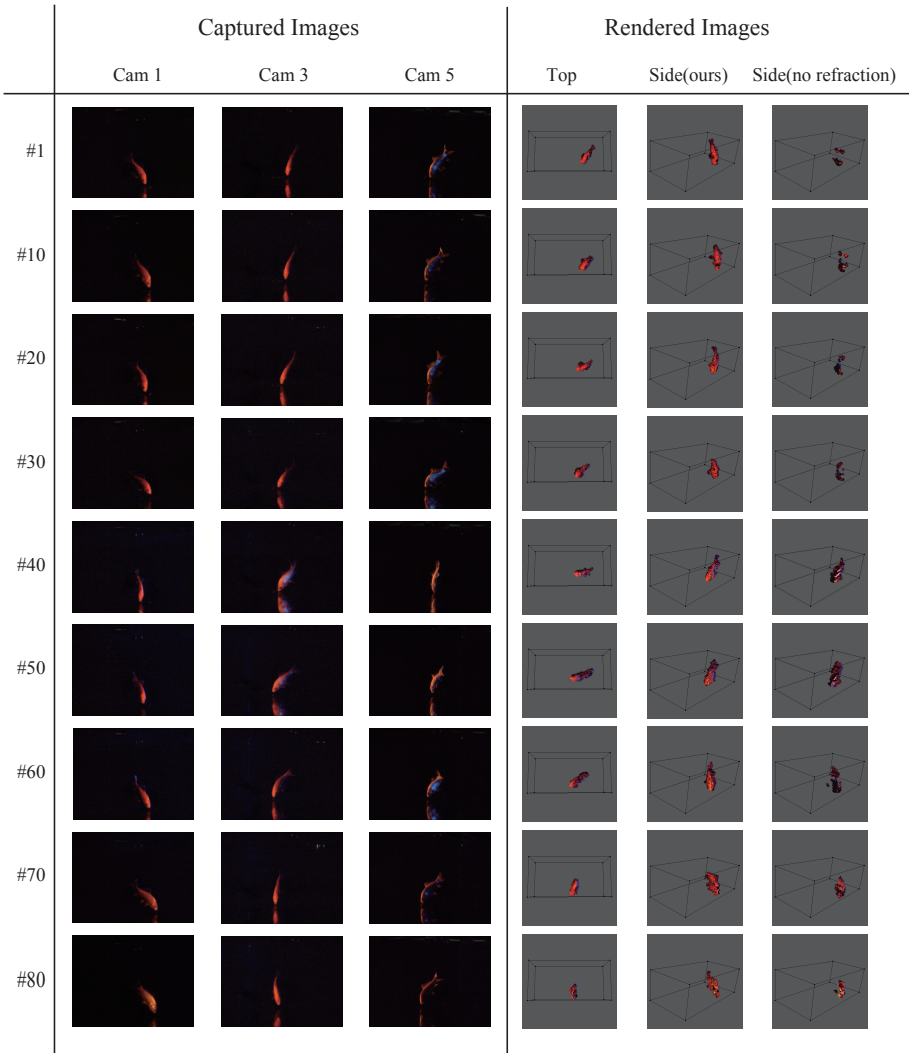


Figure 18: Result of 3D shape estimation of *goldfish*. Each row shows images of the frame indicated in the left most column.