# Calibration of Housing Parameters for Underwater Stereo-Camera Rigs

Anne Sedlazeck
http://www.mip.informatik.uni-kiel.de

Reinhard Koch
http://www.mip.informatik.uni-kiel.de

Institute of Computer Science
Christian-Albrechts-University of Kiel
Kiel, Germany

Institute of Computer Science
Christian-Albrechts-University of Kiel
Kiel, Germany

## Abstract

When using perspective cameras underwater, the underwater housing with its glass interface between water and air causes the light rays to change their direction due to refraction. In applications where geometrical properties of images are exploited without explicitly modeling refraction, i.e. when using the perspective pinhole camera model, this leads to a systematical error. This error is depending on the housing configuration like distance between camera and glass interface and angle between glass interface normal and optical axis. In this paper, we analyze the calibration of those parameters using a camera model explicitly considering refraction. The goal is to determine those parameters without the need of handling a calibration target underwater, which is cumbersome, if not impossible.

## 1 Introduction

Underwater imaging is becoming more and more popular as technology becomes available to explore the ocean floor at great water depths. Applications are the offshore industry, where underwater vehicles are used in order to manipulate objects of interest or for inspection purposes. The underwater vehicles themselves can use vision based-navigation algorithms in order to complement their on-board navigation units. In addition, archaeologists have great interest in recording, measuring, and reconstructing their objects of interest. Finally, different areas of ocean research like geology or biology profit from different techniques of underwater vision for measurements and reconstruction.

All those applications rely on accurate measurements stemming from underwater vision algorithms. However, there are some fundamental differences in underwater vision compared to vision in air. First, while still propagating through the water, light rays are absorbed and scattered when single photons encounter particles in the water or water molecules (Mobley [15]). This effect is depending on the light's wavelength and therefore effects the colors finally measured by the image sensor, leading to poor contrast and with it difficulties in for example feature matching routines.

Even more severe however, are effects caused by refraction (Hecht [7]). Refraction occurs when a light ray enters the underwater housing and passes from water through glass into
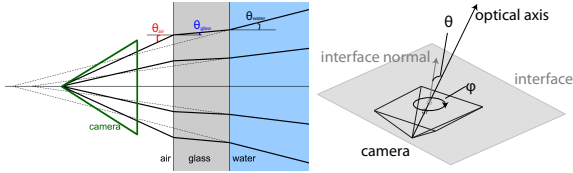
Figure 1: Left: camera with flat port underwater housing. All rays not coinciding with the interface normal are refracted twice: at the water-glass and the glass-air interface. When tracing the rays coming from the water without refraction (dashed lines), it becomes obvious that the single view point camera model is invalid. Right: the interface can be tilted with respect to the image sensor. This tilt can either be parametrized by the interface normal, or the angles $\theta$ and $\varphi$.

air (fig. 1). Those changes of media cause the ray to change its direction and therefore leads to geometrical distortions. There are two major groups of underwater housings. Flat ports are simpler to manufacture and cheaper, but also cause the more severe refraction effects. When using dome ports, ideally, the ray's direction coincides with the normal of the port in the intersection point, thus eliminating refraction effects. Often however, slight refractive effects occur due to imprecise manufacture or misalignment of camera and housing. In this paper, we will concentrate on the first case, where severe refraction occurs.

Usually, in both air and water, the perspective pinhole camera model is used for tasks like mosaicing (Eustice *et al.* [3], Trucco *et al.* [19]), reconstruction (Johnson *et al.* [9], Sedlazeck *et al.* [16]), and other measurements exploiting the implicitly contained geometry of the images. When using a perspective camera model on underwater images, the refractive effect is approximated by focal length and radial distortion (Fryer and Fraser [5], Lavest *et al.* [11]). Explicit consideration of refraction on the other hand (fig. 1) shows that the rays coming from the water are refracted twice: once at the water-glass interface and a second time at the glass-air interface, before they enter the camera. When tracing the rays coming from the water without considering refraction (dashed lines), it becomes obvious that they do not meet in one common center of projection anymore, hence the single view point camera model is invalid. Consequently, as shown by Kunz and Singh in [10], using the perspective camera model in underwater scenarios leads to a systematic error. Kunz and Singh [10] also present a calibration routine, where a checkerboard pattern is used to capture underwater checkerboard images. Unfortunately, the authors did not present any calibration results of the proposed routine.

Other works exist, where the authors calibrate a simplified model with only one refraction, omitting the shift due to the glass by assuming the glass to be very thin (Treibitz *et al.* [18]) or use underwater calibration targets like Telem and Filin in [17] or other photogrammetric methods (Li *et al.* [12]). In any case, handling a checkerboard or other calibration targets underwater is at least cumbersome and time-consuming, if not impractical.

This paper proposes a method, where based on known intrinsic parameters of the cameras of a stereo rig, the underwater housing parameters and the rig geometry are calibrated based on images captured below water, without the need for a special calibration target. In addition, as in [10] by Kunz and Singh, glass thickness is modeled explicitly because in our scenario, deep sea underwater housings with glass thicknesses in the order of several centimeters are used.

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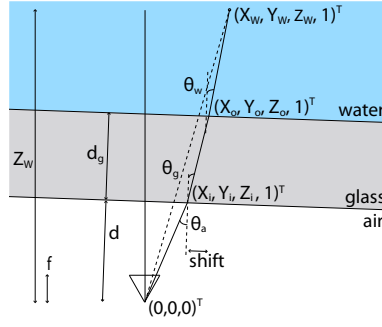
Figure 2: When unprojecting a point, the ray runs from the camera through air intersecting the inner interface plane $(X_i, Y_i, Z_i)^T$. The refracted ray in glass intersects the outer interface plane $(X_o, Y_o, Z_o)^T$, is refracted again, yielding the ray in water on which the 3D point $(X_w, Y_w,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

presented, then, the calibration routine will be introduced. After that, results on synthetic and real data will be presented, followed by a conclusion.

# 2 Refractive Camera Model

The derivation of the ray cast in the physical underwater model presented here is essentially the same as derived by Kunz and Singh in [10] with slight differences in the port parametrization. When using a flat port in front of an underwater housing, the distance to the port, the glass thickness, and the normal of the glass surface within the camera coordinate system are important parameters. Here, the inner interface plane is parametrized by $\Pi_i = (\underbrace{n_1, n_2, n_3}_{\mathbf{n}_\Pi}, -d)$ containing the normal vector and the port's distance to the origin. In addition, the outer interface plane is parametrized by the same normal vector and glass thickness $d_g$: $\Pi_o = (n_1, n_2, n_3, -(d+d_g))$ (refer to figure 2). When unprojecting [1] an image point, the goal is the computation of the point on the outer interface plane and the direction of the ray in water. First, an image point is unprojected to the ray within the camera's underwater housing $\tilde{X}_a$, using the camera's intrinsic parameters including radial distortion. This ray is in the camera coordinate system, meaning that the center of projection is in the origin. In order to find the intersection $X_i$ between ray and interface the following equation is used:

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

The intersection of the port's inner plane and the ray, parametrized by $\lambda_g$, is used to determine the point on the inner plane of the interface $X_i$. In order to compute the ray within the glass, the incidence angles $\theta_a$ and $\theta_g$ are computed. The angle $\theta_a$ between normal and

---

[1]In this paper, a camera *projects* 3D points to 2D points in the image plane, while *unproject* denotes the computation of 3D points or rays from 2D points.
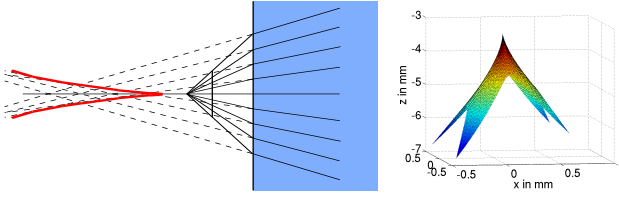
Figure 3: Caustics caused by non-single view-point camera. Left: radial image of a caustic caused by refraction at a water-air interface. Right: caustic for camera with thick glass interface and slight (ca. 2°) interface rotation. Note the asymmetric distortion.

incident ray before refraction is derived by the scalar product between the plane's normal $\mathbf{n}_\Pi$ and the ray in air, then Snell's law [2] is applied to compute $\theta_g$:

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

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

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

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

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

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

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

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

$X_w$ is still in the camera coordinate system, but using the transform of the camera with rotation and translation, the point can easily be transformed into the world coordinate system.

Figure 1 already showed that the single view point model is invalid for underwater cameras. Instead of a single view point, the rays meet in a singularity of the bundle of rays. The locus of this singularity is the caustic (fig. 3), which encompasses only a single point in case of the perspective camera - the center of projection.

In order to compute the caustic, the mapping from image coordinates to points in 3D space (5) is differentiated. The determinant of the Jacobian matrix of this parametrization is set to zero and solved for the parameter $\alpha_w$: $\det(J(X_w(x, y, \alpha_w))) = 0$. Using $\alpha_w$ in (5) allows to compute the corresponding point on the caustic for each pixel position $(x, y)$.

In order to project a 3D point into the camera system, one can either use Fermat's law and solve an equation system with four unknowns (the x- and y coordinates of the intersections on the inner and outer interface planes) or use the unprojection described above for a numerical computation. Note, that in both cases, the projection needs an optimization and is therefore less efficient to use compared to the common perspective projection.

# 3 Calibration

Important and usually unknown parameters in the above described camera model are the interface distance $d$ and the interface normal $\mathbf{n}_\pi$. This section will introduce an algorithm for calibrating those housing parameters for two cameras in a stereo rig, each in its own housing without the need to use a calibration target below water. The intrinsics (focal length, principal point, and radial distortion) of both cameras are assumed to be known. They can be calibrated using one of the established methods in air (Bouguet [2], Heikkila and Silven [8]) with one adaptation: when computing radial distortion, the polynomial is applied when undistorting images and the iteration when distorting images. This is an interesting adaptation because only the undistortion function is used for the unprojection and not the iterative distortion. In addition, we assume the glass thickness and the refractive index of the glass to be known.

We use image pairs, captured using the stereo rig, for calibration. Between images correspondences are matched using SIFT (Lowe [13]) and an Approximate Nearest Neighbor algorithm (ANN) by Arya *et al.* [1]. The feature matching is completed by a RANSAC (Fischler and Bolles [4]) routine using the Fundamental matrix (Hartley and Zisserman [6]) for outlier detection with a tolerant threshold. An initial calibration of the rig extrinsics is usually known, e.g. the baseline is about $20cm$, and there is little or no rotation between the camera poses.

## 3.1 Error Functions

When calibrating the underwater housing parameters, it is essential to find a robust error function. We experimented with the following three error functions.

**Reprojection Error**  In bundle adjustment and other optimizations in 3D reconstruction [6], usually the reprojection error in the image is minimized. The advantage is that the error is minimized in the same space, where noise in the observations (2D image points) originates. However, in the underwater case, it is extremely time-consuming to use the projection because, as mentioned before, the projection is an optimization in itself.

**Error in 3D point space**  Instead of optimizing the reprojection error in the images, it is also possible to minimize the distance error between triangulated 3D points in space. Two corresponding points of a stereo image pair are unprojected, triangulated, and the distance between the triangulated point and the 3D point parameter is used as an error. The disadvantage of this error function's minimum is its strong sensitivity to noise in the 2D data, which is why it is not considered any further.
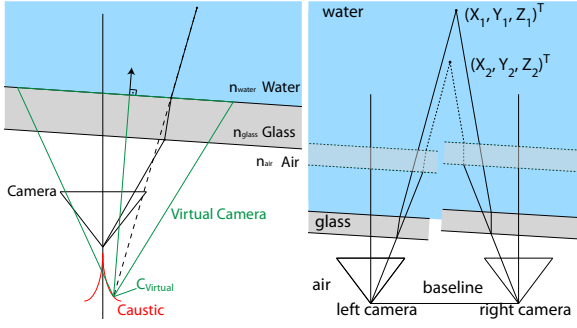
Figure 4: Left: projection of 3D point into virtual camera. Right: $d$ invariance when using two cameras only.

**Error on outer Interface Plane** In order to gain something more similar to the reprojection error within the image without having to compute the projection via optimization, we build upon the idea of Telem and Filin [17] who derive a virtual perspective projection for the 3D point into the image. Based on the intersection of the ray in water and the optical axis, the projected point is moved in the image plane according to the error caused by the camera model, without explicitly modeling the shift within the glass. However, we found that in case of rotation between interface and image sensor, the ray in water does not necessarily intersect the optical axis. Therefore, we propose to use a virtual camera with the caustic point as a the center of projection $C_v$. The virtual camera's optical axis is parallel with the interface's normal (fig. 4). 3D points can then be projected perspectively by the virtual camera, while 2D points in the image are transformed into the virtual camera as well. The point on the caustic is computed numerically as in the description above. Note, that the virtual camera needs to be computed for each 2D-3D correspondence separately because the virtual camera center, the point on the caustic is different for each 2D point. The virtual camera's rotation $\mathbf{R}_v$ is defined by a rotation axis $\mathbf{n}_{rot}$ and a rotation angle $\theta_{rot}$, derived from cross and scalar product of interface normal and optical axis of the original camera respectively:

$$\mathbf{n}_{rot} = \mathbf{n}_\pi \times (0,0,1)^T \qquad \theta_{rot} = \arccos(< (0,0,1)^T, \mathbf{n}_\pi >) \qquad (6)$$

The 3D point in water $\mathsf{X}_w$ is first transformed into the camera's local coordinate system and then into the virtual coordinate system by, the camera's and virtual camera's rotations $\mathbf{R}_{cam}$, $\mathbf{R}_v$ and centers of projection $C_{cam}$ and $C_v$ respectively:

$$\mathsf{X}_{localCam} = (\mathbf{R}_{cam}^T | - \mathbf{R}_{cam}^T C_{cam}) \mathsf{X}_w \qquad (7)$$
$$\mathsf{X}_v = (\mathbf{R}_v^T | - \mathbf{R}_v^T C_v) \mathsf{X}_{localCam}.$$

In addition, the 2D point in the image is unprojected onto the outer interface $\mathsf{X}_o$ by the method described in the section above. It is then transformed into the virtual camera coordinate system as well:

$$\mathsf{X}_{ov} = (\mathbf{R}_v^T | - \mathbf{R}_v^T C_v) \mathsf{X}_o \qquad (8)$$

Both points are then projected using the focal length of the virtual camera $f_v$ in figure 4 (on the left):

$$f_v = (< \tilde{\mathsf{X}}_w, \mathbf{n}_\pi >) \, \| \, (C_v - \mathsf{X}_o) \, \|_2 \qquad (9)$$
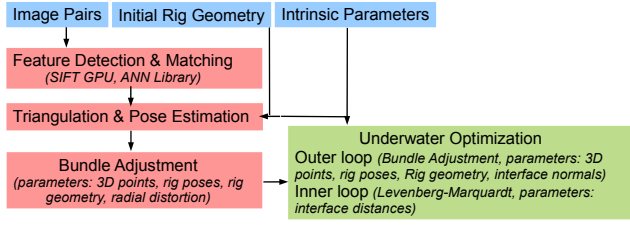
Figure 5: overview for system with multiple images.

The error for a set of $N$ 2D-3D point correspondences is the sum of squared distances between unprojected 2D image point $\mathsf{X}_{ov}$ and projected 3D point $\mathsf{X}_v$ in the virtual camera:

$$error_{outerInt} = \sum_{i \in 1,\ldots,N} \| \mathsf{X}_{ov_i} - \mathsf{X}_{v_i} \|_2^2 \tag{10}$$

The virtual rotation, which is derived directly from the interface normal to be calibrated, leads to very high errors in case of erroneous normal parameters and therefore to robustness against noise in the observation data. At the same time it is largely invariant against erroneous camera-interface distances.

## 3.2 Calibration Algorithms

### 3.2.1 Stereo Image Pair

The above error functions were utilized in different calibration scenarios. When computing 3D points from correspondences in one stereo pair, it is possible to estimate the normal between interface and image sensor using Bundle Adjustment (McGlone [14]) optimizing 3D points and normals with the virtual camera error. Note, that in this simple case, the rig geometry is not optimized, only the normal parameters.

The interface distance $d$ cannot be optimized in this simplified scenario because, as can be seen in figure 4, a fitting 3D point can be found for each different interface distance.

### 3.2.2 Multiple Image Pairs

In order to be able to optimize the interface distance, we use several image pairs and the corresponding rig transformations. Initial rig transformations are assumed to be known, but can be calibrated with a SfM approach (refer to Hartley and Zisserman [8]). The initial rig transformations are optimized with the classic perspective pinhole camera model by approximating refractive effects by focal length and radial distortion. The known functional correlation between the focal length in water and the focal length in air $1.3333 f_{air} = f_{water}$ (see Fryer and Fraser [5], Lavest *et al.* [11]) is used as an approximate intrinsic camera calibration. In this equation, 1.333 is the refractive index of water, which is not optimized in our approach, since the index of refraction changes only about 2% in the whole relevant range of ocean water bodies (Mobley [15]). The resulting intrinsic calibration is used to refine the initial set of cameras and the sparse 3D point cloud using the adjustment method with the
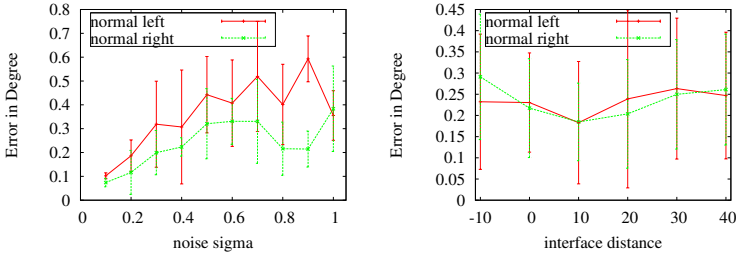
Figure 6: Results from stereo image based normal estimation. Left: interface distance and camera poses were correct, added noise varied. Right: error in normal estimation with erroneous interface distances, ground truth was 20*mm*. The algorithm takes about 5 minutes for optimization on a common desktop PC.

perspective camera model. Parameters are the rig poses (translation and rotation), the relative transform of the slave camera in the rig, radial distortion, and the 3D points. This step yields a good approximation of the scene that can now be used to find the parameters for the underwater housing. Experiments with the above listed error functions lead to a system with nested loops: the outer loop minimizes the virtual camera error functions, while the inner loop optimizes the interface distance using the reprojection error. Parameters and usage of error functions are summarized in fig 5.

# 4 Results

The algorithm was implemented in C++ using the BIAS Image Library [2]. In this section results on synthetic and real data will be presented.

## 4.1 Synthetic Data

Synthetically generated sets of 3D points with the corresponding 2D points allow to evaluate the method's ability to deal with normal distributed noise added to the 2D observations. This was tested in two different scenarios. First, estimating the interface normals from one stereo image pair was simulated and the normals were initialized with $(0,0,1)^T$, while the true normal was rotated by the angles $\varphi_{left} = 40°$, $\theta_{left} = -2°$, $\varphi_{right} = -10$, and $\theta_{right} = -1°$ (see fig. 1 right) and the results in figure 6 show that the estimation works.

In a second scenario, multiple image pairs were used for optimizing interface normal, interface distance, and camera poses as in figure 5 except that the geometric data was generated and then optimized using the perspective Bundle Adjustment routine. The results are shown in figure 7: the initial interface distances were 10*mm*, the true interface distances were 20*mm*. Note, that part of the error caused by using the perspective camera model is absorbed by the camera poses and not only by focal length and radial distortion, so after working perspectively, optimizing the camera poses along with the housing parameters is essential.
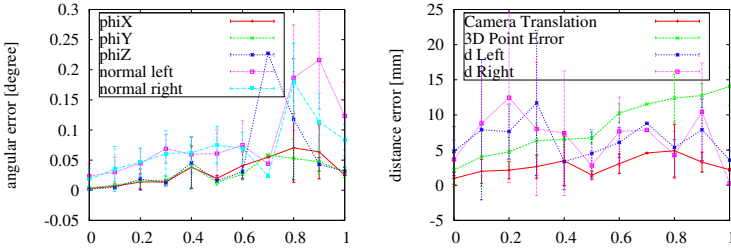
Figure 7: Results for synthetic tests on multiple image pairs. In order to simulate initialization after perspective Structure from Motion, the data was optimized using the perspective Bundle Adjustment. After that, the underwater optimization was started with initial interface normals $(0, 0, 1)^T$ and interface distance $10mm$, ground truth was $20mm$. The run-time of the algorithm was in the order of 3 hours on a common desktop PC. However, there is still room for optimization.



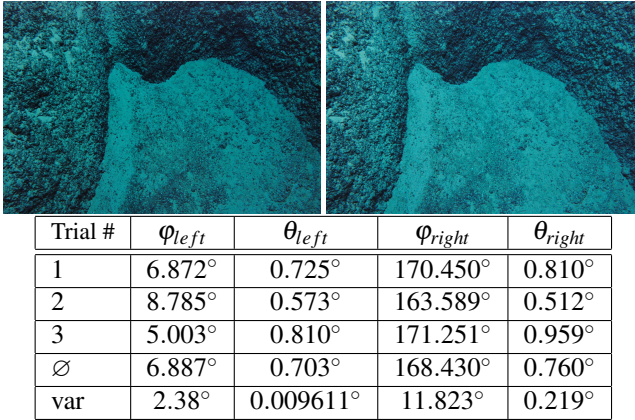| Trial # | $\varphi_{left}$ | $\theta_{left}$ | $\varphi_{right}$ | $\theta_{right}$ |
|---------|---------|---------|---------|---------|
| 1 | 6.872° | 0.725° | 170.450° | 0.810° |
| 2 | 8.785° | 0.573° | 163.589° | 0.512° |
| 3 | 5.003° | 0.810° | 171.251° | 0.959° |
| ∅ | 6.887° | 0.703° | 168.430° | 0.760° |
| var | 2.38° | 0.009611° | 11.823° | 0.219° |

Figure 8: Top row: exemplary input images. Bottom: results of normal estimation on real images.

## 4.2 Synthetically Rendered Images

Using a simulator, we can render images compliant with the above described camera model. This allows us to test the performance of the method when using images, thus having feature matching errors to cope with. The results in figure 6 on the right show that the normal can be estimated based on feature correspondences even if the interface distance is in the order of centimeters from the true interface distance.

## 4.3 Real Data

Normal estimation has also been tried on real images taken using a stereo rig mounted on a ROV. Since the ground truth is unknown, only robustness in estimation can be judged. In order to gain enough correspondences from the images, several pairs were used for feature matching. Different such sets of images were used to compute the normals shown in figure 8 on the right. Even though the accuracy of the camera rig calibration and the interface

distance are unknown, the interface normals could be estimated.

# 5 Conclusion and Future Work

We have presented a system for calibrating the underwater housings of a stereo rig using a camera model explicitly modeling refraction. It allows to determine the normals of the interface with respect to the optical axis. In addition, the interface distance can be calibrated when using several image pairs captured from different poses in space with overlapping view ports. Assuming the glass thickness and index of refraction of the underwater housing to be known, this allows calibrating the underwater housings of the stereo rig without using a checkerboard underwater. In the future, we hope to improve robustness and run time of the system and investigate extensions towards a complete Structure-from-Motion pipeline with explicit consideration of refraction.

# References

[1] Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM*, 45:891–923, November 1998. ISSN 0004-5411.

[2] J. Y. Bouguet. *Visual methods for three-dimensional modelling*. PhD thesis, California Institute of Technology Pasadena, CA, USA, 1999.

[3] R. Eustice, H. Singh, and J. Howland. Image registration underwater for fluid flow measurements and mosaicking. In *OCEANS 2000 MTS/IEEE Conference and Exhibition*, volume 3, pages 1529 –1534 vol.3, 2000.

[4] M. Fischler and R. Bolles. RANdom SAmpling Consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981. ISSN 0001-0782.

[5] J. G. Fryer and C. S. Fraser. On the calibration of underwater cameras. *The Photogrammetric Record*, 12, 1986.

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision (Second Edition)*. Cambridge University Press, second edition, 2004.

[7] Eugene Hecht. *Optik*. Oldenburg Verlag Muenchen Wien, 2005.

[8] Janne Heikkila and Olli Silven. A four-step camera calibration procedure with implicit image correction. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1106, 1997. ISSN 1063-6919.

[9] M. Johnson-Roberson, O. Pizarro, S.B. Williams, and I.J. Mahon. Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys. *Journal of Field Robotics*, 27, 2010.

[10] C. Kunz and H. Singh. Hemispherical refraction and camera calibration in underwater vision. In *OCEANS 2008*, pages 1 –7, 15-18 2008.

[11] Jean-Marc Lavest, Gérard Rives, and Jean-Thierry Lapresté. Underwater camera calibration. In *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part II*, pages 654–668, 2000.

[12] R. Li, C. Tao, and W. Zou. An underwater digital photogrammetric system for fishery geomatics. In *Intl. Archives of PRS*, volume Vol.XXXI, pages pp.319–323, 1996.

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[14] J. C. McGlone, editor. *Manual of Photogrammetry*. ASPRS, 5th edition, 2004.

[15] Curtis D. Mobley. *Light and Water: Radiative Transfer in Natural Waters*. Academic Press, 1994.

[16] A. Sedlazeck, K. Koser, and R. Koch. 3d reconstruction based on underwater video from rov kiel 6000 considering underwater imaging conditions. In *Proc. OCEANS '09. OCEANS 2009-EUROPE*, pages 1–10, May 11–14, 2009. doi: 10.1109/OCEANSE. 2009.5278305.

[17] Gili Telem and Sagi Filin. Photogrammetric modeling of underwater environments. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(5):433 – 444, 2010. ISSN 0924-2716.

[18] T. Treibitz, Y.Y. Schechner, and H. Singh. Flat refractive geometry. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR 2008*, pages 1–8, 2008.

[19] E. Trucco, A. Doull, F. Odone, A. Fusiello, and D. Lane. Dynamic video mosaicing and augmented reality for subsea inspection and monitoring. In *In Oceanology International, United Kingdom*, 2000.