Hemispherical Refraction and Camera Calibration in Underwater Vision

Clayton Kunz and Hanumant Singh
Deep Submergence Laboratory
Woods Hole Oceanographic Institution
Woods Hole, MA 02543, USA
Email: {ckunz,hsingh}@whoi.edu

***Abstract*— We examine the challenges to underwater vision that are caused by light refracting through both hemispherical and planar pressure housing interfaces. As light travels through water, into a pressure housing, and finally into a camera, it is bent according to Snell's law, rendering the typically-used perspective camera model invalid. Through numerical simulation, we examine the degree to which the incorrect perspective model results in 2-D image distortion, and errors in 3-D scene reconstruction computed using stereo vision or structure from motion. We focus on the use of hemispherical pressure housing interfaces with imperfectly mounted cameras, and include comparisons with cameras behind planar interfaces. We also address the problem of calibrating a camera model which takes into account the physical parameters of the pressure housing interface.**

I. Introduction

High-quality digital cameras are increasingly being used on AUVs and ROVs to produce dense image sets of the sea floor. While these images are useful data products in and of themselves (especially when combined into photomosaics [1]), three-dimensional terrain models represent the next advance in underwater imaging.

Acoustic techniques can produce high-resolution depth maps [2], but without visual texture information. Optical systems can provide very high resolution (on the order of 0.045 degrees per imaged pixel) with low power consumption, and produce images that humans find easy to interpret. Techniques from photogrammetry and computer vision, including stereo vision and structure from motion [3] [4], combine several images of a single scene into a three-dimensional map. These techniques extract depth information by triangulating two or more views of a point in the world, and, with sufficiently many views, may also compute the locations and internal geometries of the cameras used to capture the images. While these techniques are well-understood in air [5], they have yet to be widely used underwater, particularly when cameras are attached to moving vehicles (though see [6] and [7]). This is partially because of the sheer difficulty in building and deploying underwater vehicles and cameras, but it is also because of the additional complications brought by the underwater environment, including lighting restrictions, optical backscatter, and refraction.

In this paper we address the errors induced by refraction, when cameras are mounted behind hemispherical or planar air/water interfaces. We begin with an examination of the underlying physics (section II). We examine both the 2-D pixel error induced by assuming that no refraction is taking place (section III), and the corresponding 3-D error induced by triangulating corresponding pixels from multiple views (section IV). We then turn to the problem of calibrating a camera rig to capture the correct physical characteristics of a system (section V), before a general discussion in section VI.

II. Background

In order to construct a 3-D scene model from a sequence of images, one must have geometric models of the camera and its motion, and the correspondences between pixels in a sequence of images. In air, a camera is typically modelled as a pinhole and an imaging plane, which yields an image formed by perspective projection. This model is nice to work with, because perspective projection is a linear operation (when projective geometry is used), and because the geometric relationship between a set of cameras imaging a single point in space induces a constraint (the epipolar constraint) that simplifies the search for a matching pixel in one image to a single straight line in any other image. (Other camera models also induce epipolar constraints, but the epipolar curves are typically not straight lines [8]).

If a camera is to work underwater, it must be enclosed in a pressure housing with a transparent interface that allows light to pass from the water into the camera's lens. This leads to refraction, as the light must pass through several media before it reaches the sensor: a ray of light is bent twice as it moves from the water into the camera, once when it transitions from water to glass, and again when it transitions from glass to air. The amount of bending is dictated by Snell's Law, and it depends on the angle between the ray of light and the surface normal of the interface, and the indices of refraction of the media: $n_1 \sin\theta_1 = n_2 \sin\theta_2$. For this paper, we assume $n_{\text{air}} = 1$, $n_{\text{water}} = 1.333$, and $n_{\text{glass}} = 1.46$. Clearly, if a ray of light passes through a change in media such that the light ray and the surface normal of the boundary are the same (i.e. $\theta_1 = 0$), then no bending takes place. In general, however, refraction causes a dispartity between apparent light direction, and true light direction, as illustrated in figure 1.

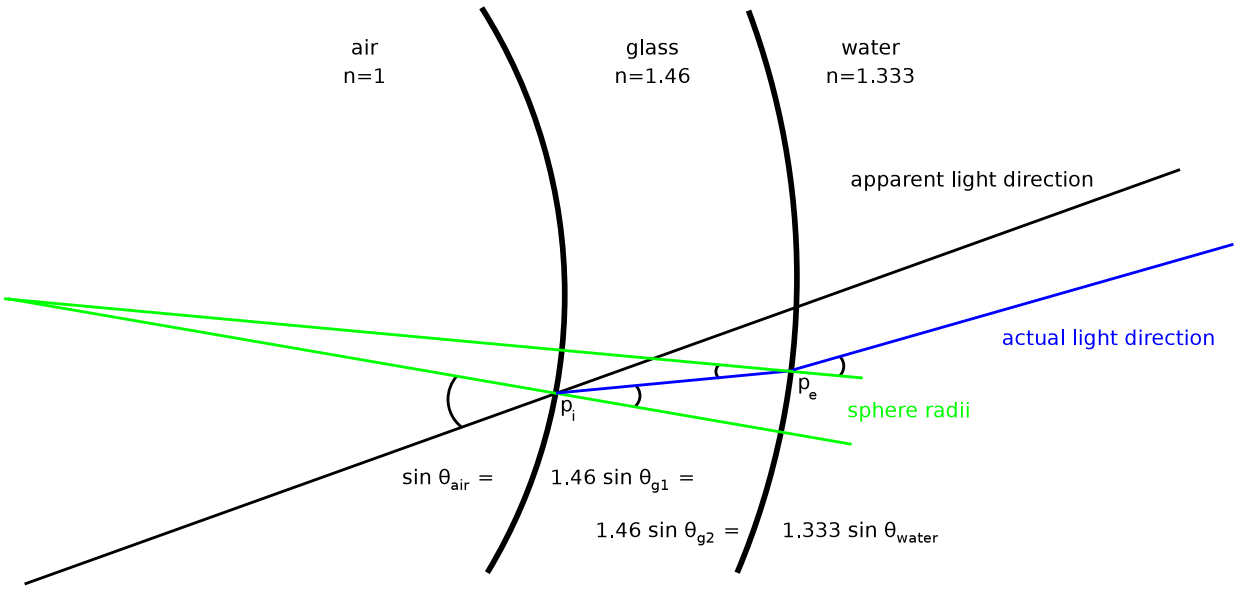

Fig. 1. Refraction of light from water to air through a hemispherical interface. The straight black line is the apparent path of a light ray, while the blue lines show the true path. The green lines show surface normals where the light moves from one medium to another. The angles are exaggerated.

Perspective Projection

Because the cameras we use underwater are well-modeled by perspective projection when they are used in air, we start with the perspective model, and extend it to encompass external refraction. The procedure for computing the mapping between points in space and their images relies on the perspective model, which we briefly describe here.

Perspective projection is the process by which a point in (3-D) space maps to a (2-D) pixel on an image plane. It models an ideal pinhole camera, which is approximated by most non-telephoto lenses used in underwater applications. The model makes use of projective geometry, to keep operations linear (see [4] for a much more thorough discussion of perspective projection and projective geometry).

Points $\mathbf{X}$ in projective 3-space ($\mathcal{P}^3$) map to pixels $\mathbf{x}$ in projective 2-space ($\mathcal{P}^2$) via a 3×4 matrix $\mathbf{H}$: $\mathbf{x} = \mathbf{HX}$. Actual pixels (x, y) on the image plane are found from homogeneous pixels $\mathbf{x}$ by scaling $\mathbf{x} = [\tilde{x}, \tilde{y}, \tilde{w}]^{\mathrm{T}}$ by so that the third coordinate is one, i.e. $[x, y] = (1/\tilde{w})[\tilde{x}, \tilde{y}]$. The matrix $\mathbf{H}$ describes the external and internal camera parameters, and can be broken down as $\mathbf{H} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$, where

$$\mathbf{K} = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

describes the internal parameters: f_x and f_y are the focal length of the system, in units of pixel sizes in the x and y pixel directions; c_x and c_y are the pixel numbers of the "principal point" where the optical axis intersects the image plane; and γ is the skew factor, which captures any nonzero angle between the x and y pixel directions.

The matrix $\mathbf{R}$ is a 3×3 rotation matrix, and $\mathbf{t}$ is a 3×1 translation vector, which together decribe the location of the camera in space. The matrix $\mathbf{T} = [\mathbf{R}|\mathbf{t}]$ transforms points referenced to some external frame to the "camera coordinate frame," in which the center of projection is at the origin, the x and y axes are parallel to the x and y directions on the image plane, and z points out from the center of projection through the lens and into space.

In addition to the linear operation of projection, the perspective model also includes nonlinear terms to capture lens distortion. We use a 4-parameter model, which includes two terms each for radial (κ_1 and κ_2) and tangential (ρ_1 and ρ_2) distortion; this model is also used by Bouguet's MATLAB camera calibration toolbox [9], and by the OpenCV computer vision library [10], inspired by [11]. In this model, observed (distorted) pixels $(\hat{x}, \hat{y})$ are related to ideal pixels (x, y) by

$$\bar{x} = (x - c_x)/f_x$$
$$\bar{y} = (y - c_y)/f_y$$
$$r^2 = \bar{x}^2 + \bar{y}^2$$
$$\hat{x} = x + (x - c_x)(\kappa_1 r^2 + \kappa_2 r^4 + 2\rho_1\bar{y} + \rho_2(r^2/\bar{x} + 2\bar{x}))$$
$$\hat{y} = y + (y - c_y)(\kappa_1 r^2 + \kappa_2 r^4 + 2\rho_2\bar{x} + \rho_1(r^2/\bar{y} + 2\bar{y}))$$

When projecting a point to a pixel, first the linear operation $\mathbf{x} = \mathbf{HX}$ is performed, the coordinates are "de-homogeneized," and then the observed pixel is found using the above equations. When computing a ray corresponding to a pixel, the above equations are first inverted (numerically) to find the "ideal" pixel location, and then the ray is determined using

$$\mathbf{l_0} = -\mathbf{R}^{\mathrm{T}}\mathbf{t}$$
$$\mathbf{l_1} = \mathbf{R}^{\mathrm{T}} \begin{bmatrix} (x - c_x)/f_x \\ (y - c_y)/f_y \\ 1 \end{bmatrix} = \mathbf{R}^{\mathrm{T}}\mathbf{K}^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

The point $\mathbf{l_0}$ is the camera center, in world coordinates, and the vector $\mathbf{l_1}$ points in the direction of the pixel on the image plane, rotated into world coordinates. A point $\mathbf{X}$ is on the ray if $\mathbf{X} = \mathbf{l_0} + \lambda \mathbf{l_1}$ for some λ.

Underwater Cameras and Spherical Interfaces

There are two schools of thought on interface design: one can use a flat piece of acrylic or glass (basically a thick window), or one can use a hemisphere of similar material. Planar interfaces have the advantages that they are less expensive and easier to work with than hemispherical interfaces, and the disadvantages of reduced field-of-view, and increased distortion due to refraction. An examination of flat refractive geometry is presented in [12]. When a hemispherical interface is used with a pinhole camera, and the center of projection is positioned at the center of the sphere, all refractive distortion from the hemisphere is eliminated, because all light rays passing through the center of projection of the camera are orthogonal to the interior and exterior surfaces of the hemisphere.

A pinhole camera perfectly mounted to a hemispherical interface can be calibrated in air using standard techniques [13], and the calibration will still be valid in water. A perfect mount is difficult mechanically, however, particularly when performed at sea. If the camera center is translated at all from the center of the hemisphere, nonlinear distortions are induced. Moreover, because refraction takes place away from the camera's center of projection, and because the degree of bending varies from pixel to pixel, the perspective model is no longer valid. Incoming light rays no longer intersect at a single point. Perhaps more disturbingly, the image of a straight line in space is no longer straight, so the perspective epipolar constraint (the "fundamental matrix" [14]) fails.

A planar interface can be thought of as a hemispherical interface with an infinite radius. This implies that the ideal housing would position a camera infinitely far from the interface, which is clearly impractical. Sticking with the perspective model, then, will always lead to some kind of error in the mapping from points in space to pixels. The error in the planar case is also likely to be more extreme than in the hemispherical case, because the displacement from the optimal position will be greater.

Since the perspective camera model fails in these cases, it makes sense to use a physics-baesd model instead. Grossberg and Nayar's "raxel" model [15] is a nice starting point; it simply maps pixels to the rays in space which they image, which don't necessarily converge in a single point. While their model uses rays attached to an imaging system's caustic surface, in this case it is perhaps more natural to use the surface of the glass/water interface itself, and to retain the perspective model as valid inside the pressure housing.

To map a pixel to the ray it images, the point on the interior of the interface can be found using the perspective model as described above, and finding the intersection of the ray with the interior boundary of the interface. Then, we follow the light ray through the interface, bending it as necessary, to the point where it reaches the water, where it bends again. The imaged raxel is characterized by this point on the exterior of the interface, and the direction given by Snell's law:

- Pixel to interior of interface:

$$\mathbf{l_0} = [0, 0, 0]^{\mathrm{T}}$$

$$\mathbf{l_1} = \mathbf{K}^{-1}\mathbf{x}$$

$\mathbf{p_i}$ = intersection of line ($\mathbf{l_0}$, $\mathbf{l_1}$) with interior of interface

θ_{air} = angle between line and interface normal at $\mathbf{p_i}$

$$\theta_{\text{g1}} = \arcsin \frac{\sin \theta_{\text{air}}}{n_{\text{glass}}}$$

$\mathbf{q_i}$ = surface normal at $\mathbf{p_i}$ rotated by θ_{g1}

- Interior of interface to exterior of interface:

$(\mathbf{l_2}, \mathbf{l_3})$ = line defined by $\mathbf{p_i}$ and $\mathbf{q_i}$

$\mathbf{p_e}$ = intersection of line ($\mathbf{l_2}$, $\mathbf{l_3}$) with exterior of interface

θ_{g2} = angle between line and interface normal at $\mathbf{p_e}$

- Exterior of interface into water:

$$\theta_{\text{water}} = \arcsin \frac{\sin \theta_{\text{g2}}}{n_{\text{water}}}$$

$\mathbf{q_e}$ = surface normal at $\mathbf{p_e}$ rotated by θ_{water}

These calcuations yield a description of the imaged raxel in the camera's local coordinate frame. The ray starts at $\mathbf{p_e}$ and points in the direction of $\mathbf{q_e}$.

There are many ways to rotate vectors in 3-D. In our implementation, we compute θ_{air} by taking vector dot products, and then compute the direction of the ray as it is bent by rotating the local normal vector by θ_{g1} about the axis given by the cross product between the local normal vector and the incoming ray. The rotation itself can be carried out using unit quaternions. Care must be taken in degenerate cases (for example, if θ_{air} is very near zero, or if the center of projection is very close to the interface), but these are easy to catch. The orientation of the vector exiting the interface is computed similarly.

The image formation model, mapping points in space to their images, inverts the process described above; because of the trigonometry involved we compute the inverse numerically. Once the correct point on the interior of the interface is found, the imaged pixel is computed using the perspective projection model.

A full model of imaging systems with hemispherical or planar air/water interfaces must take into account the relationship between the camera and the interface, as well as the properties of the interface itself (sphere radius, and glass thickness). It is worth noting that symmetries reduce the number of degrees of freedom (DOF) in both cases from 6 to 3. In the hemispherical case, camera rotation can be thought of as an extrinsic property, while in the planar case, camera translation parallel to the interface, as well as rotation about the optical axis, can be thought of as extrinsic properties. In other words,

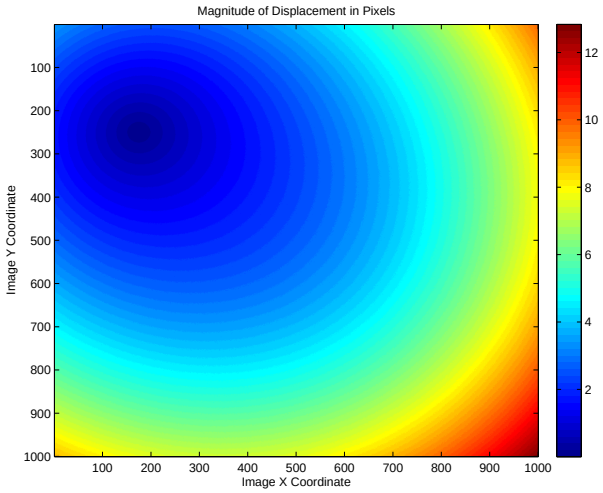

Fig. 2. Pixel distortion for an example set up, using a 1-megapixel camera with a 90-degree field of view. This image simulates 2.6 mm of camera displacement from the center of a hemisphere with radius 8 cm and thickness 2.54 cm. Simulated object distance averages 3 meters from the sphere, with standard deviation of 50 cm.

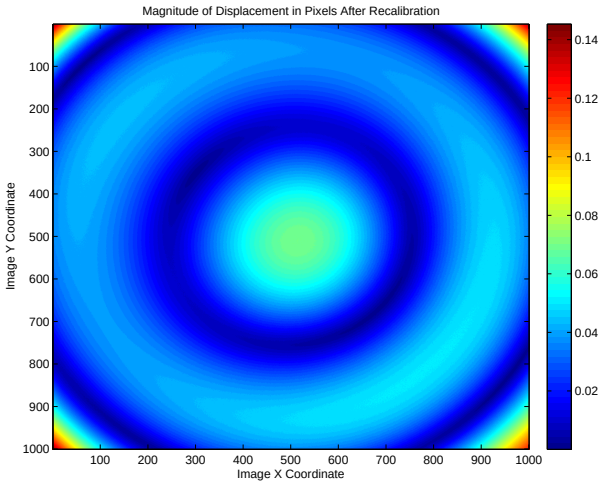

Fig. 3. Distortion after optimal recalibration in water (root mean squared error is 0.036 pixels), using perspective model with lens distortion.

these motions of a camera inside a pressure housing do not affect the magnitude of the bending of light rays, rather just the direction of the bending, and so they can be modelled as motions of the pressure housing itself.

III. 2-D Pixel Error

Given that a camera perfectly mounted inside a sphere can be calibrated in air, it is useful to get a feel for how severely distorted images will be when the mount is not perfect. Because the amount of distortion depends on both the camera motion inside the sphere (three degrees of freedom) and the distance to the objects being viewed (most generally one degree of freedom for each imaged pixel), it is difficult to visualize the distortion. It is straightforward to simulate the distortion for a typical case, however, as shown in figure 2. The figure was generated by measuring the pixel displacements between an ideal camera with known parameters

$$\mathbf{K} = \begin{bmatrix} 500 & 0 & 500 \\ 0 & 500 & 500 \\ 0 & 0 & 1 \end{bmatrix}$$

and no lens distortion viewing a "noisy plane" 3 meters away and parallel to the image plane, and the same camera viewing the same points, but placed behind an hemispherical interface and accounting for refraction. In this particular set up, the hemisphere is 2.54 centimeters thick, with radius 8 centimeters, and the camera is located at $(0.0013, 0.001, -0.002)$ meters relative to the center of the sphere. This is the pixel error in the "point-to-pixel" computation described above when using these parameters.

It is not likely that someone interested in underwater photogrammetry would calibrate their camera in air and assume that the mount inside the pressure housing would be perfect. Typically, one instead calibrates the camera in water, using something like the MATLAB calibration toolbox. The perspective model does not account for refraction, however, so this "wet calibration" will still result in distorted images. Figure 3 shows the error using the recalibrated perspective model. In this simulated case, because there is perfect knowledge of the 3-D location that each pixel images, it is possible to do an "optimal" calibration, as described in [4]. The recovered model has

$$\mathbf{K} = \begin{bmatrix} 496.6660 & 0.0279 & 496.1866 \\ 0 & 496.6598 & 497.0628 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

$$(\kappa_1, \kappa_2, \rho_1, \rho_2) = (-0.0029, 0.0003, -0.0012, -0.0016)$$

The model is good enough to reduce distortion to far less than one pixel width on average, but the distortion still depends on the distance from camera to the object, and so will increase as the camera moves closer to or further from the observed scene.

To visualize how the camera displacement affects image distortion using this optimal in-water calibration, we ran the same simulation described above 800 times, starting with the simulated camera positioned at the center of the hemisphere, and each time moving the camera 25 micrometers further from the center, along the same direction vector used above. For each resulting distortion map, we pulled the distortions along the main diagonal, and stacked them, producing figure 4. This way, the y axis of the figure cooresponds to camera displacement, the x axis corresponds to pixel number, and the color corresponds to displacement, as above. Clearly, as the camera moves further from the center of the hemisphere, the effectiveness of the perspective camera model with lens distortion at compensating for refraction diminishes.

For comparison, we repeated the simulations with a planar interface, varying the camera location from zero to 2 centimeters from the interface, while at the same time varying camera rotation from zero to 2 degrees about the line $x = y$ in the

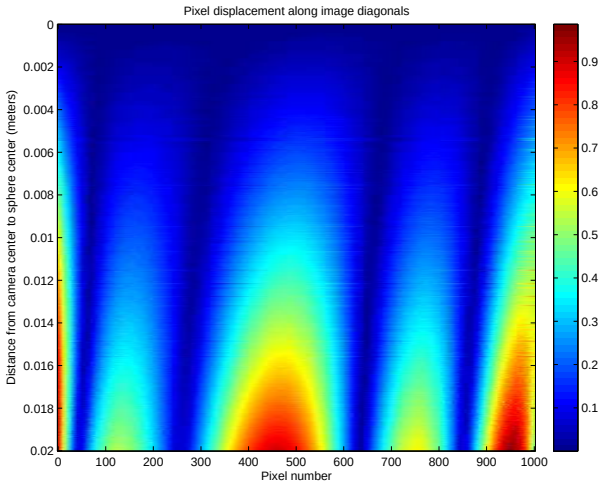

Fig. 4. Pixel distortion as a function of camera displacement from the center of the sphere. Each line in this image is taken from the diagonal of an image like that shown in figure 3. The shape of the surface depends on the direction of camera displacement, but the magnitude increases as the camera moves from the center of the sphere, regardless of the direction it moves in.

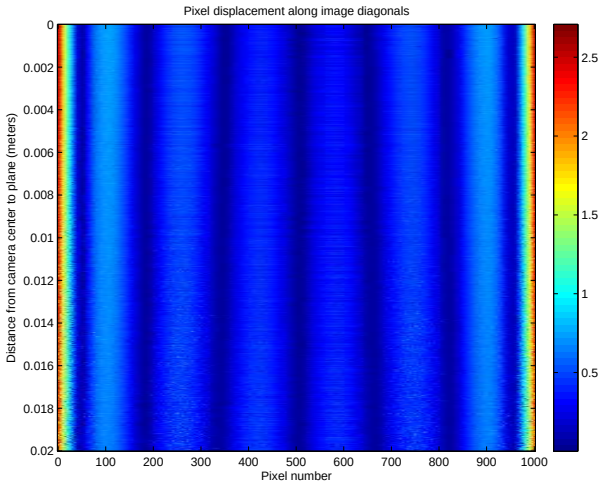

Fig. 5. Pixel distortion along the diagonal of each of 800 images taken at different camera-plane distances. The error increases dramatically near the image corners, but does not significantly change as the camera moves relative to the planar interface.

image plane. The summary image is shown in 5. It is worth noting that the error does not change as much over the course of the simulation as it did in the case of the spherical interface. This may be because the dominant source of error is induced by displacement alone, which produces a symmetric distortion that can be handled to a reasonable extent by lens distortion terms. The error jumps significantly near the corners of the images (seen by the bands near the left and right edges of the error figure), which shows the limitations of using only four terms for radial and tangential distortion.

A more direct comparison between planar and hemispherical interfaces is shown in figure 6. It is worth noting that the overall RMS distortion decreases as the camera moves further from the planar interface with no rotation. This appears to be

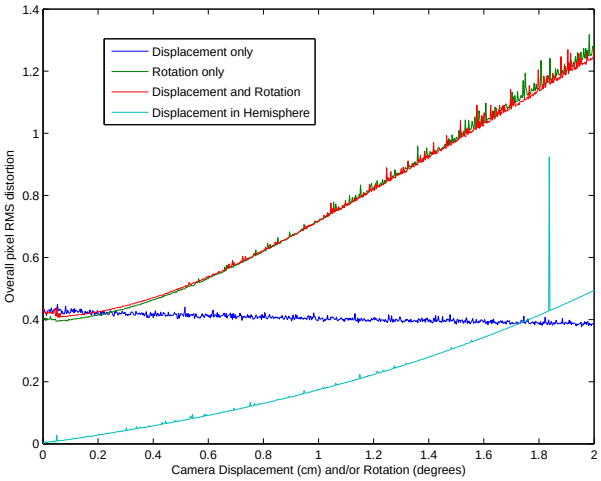

Fig. 6. Pixel distortion in four different cases. The first three cases are for planar interfaces. In the first case, the camera is translated away from the interface from zero to two centimeters. In the second case, the camera is fixed at one centimeter from the interface, and is rotation from zero to two degrees. In the third case, which is hard to distinguish from the second, both rotation and translation happen simultaneously. The fourth case shows displacement only for a camera with a hemispherical interface.

in conflict with the results of [12], but in fact is not: in this case we're examining actual pixel displacements and using a model that includes lens distortion, rather than examining the size of the system's caustic surface. Rotation has a much more detrimental effect on refraction-induced distortion with a planar interface, because of its inherent asymmetry. For reasonable camera displacements, the hemispherical interface always yields less error than the planar interface.

IV. 3-D Reconstruction Error

Even though unmodelled refraction causes the perspective epipolar constraint to fail, it is still possible to find correspondences between multiple views of the same scene, and then to triangulate rays from the corresponding pixels into depth estimates. When using the standard perspective model, the depth estimates will be incorrect, because the pixel-to-ray mapping is wrong. Even if one were to compensate for all 2-D error by adding higher-order lens distortion terms, the 3-D mapping will still be incorrect, because the rays that are actually imaged by the system do not intersect in a single point, which is assumed by the perspective model.

To quantify the 3-D reconstruction error, we take the simulations described above one step further. We compare the estimated locations of 3-D points computed given perfect image-to-image correspondences, and using the physics-based models and the optimal (but incorrect) "wet-calibrated" perspective models described above:

- Use the physics-based model to compute which points on a simulated plane 3 meters from the camera are imaged (pixel-to-ray step).
- Use the physics-based model to compute which pixels image these points when the camera has moved 0.75

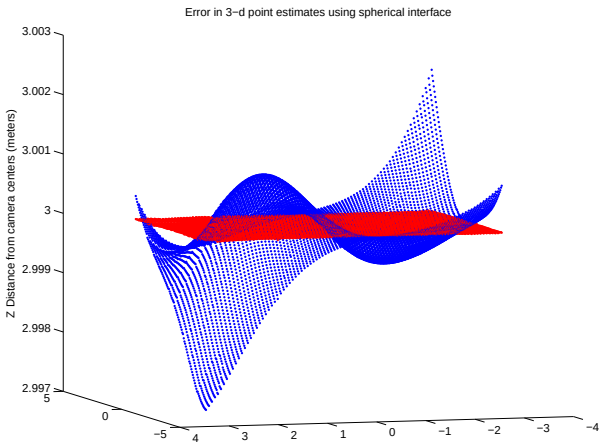

Fig. 7. Reconstruction error for known corresponding points between a pair of images. The simulated camera motion is 75cm, and the distance to the imaged plane is 3m. The correct reconstruction is in red, while the reconstruction without modelling refraction is in blue.

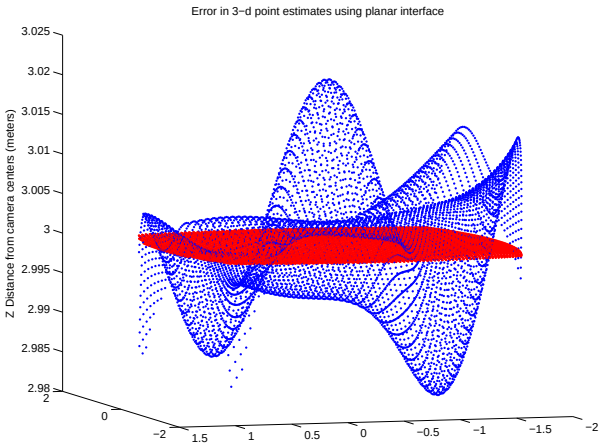

Fig. 8. Reconstruction error for known corresponding points between a pair of images taken behind a planar air/water interface. The camera motion is the same as in figure 7. Only pixels visible by both cameras are shown.

meters (point-to-pixel step). Some of these pixels will be outside the camera's field of view.

- Use the perspective model to compute rays for these corresponding pixels (pixel-to-ray step with perspective model).
- Triangulate these rays to compute depth estimates. We do this by finding the 3-D point that has the minimum sum of squared distances to the two rays.

We repeated these steps using both a hemispherical interface and a planar interface, with the same glass thickness and sphere radius used above. For the hemispherical case, the camera was located at $(0.0013, 0.001, -0.002)$ meters relative to the sphere center, as before, for the planar case, the camera was 5 millimeters behind the plane, rotated one degree about the $x = y$ line. The spherical reconstruction error is shown in figure 7, and the planar reconstruction error is shown in figure 8. There is less error in the hemispherical case (generally less than a millimeter) than in the planar case (on the order of about a centimeter). In fact, the error induced by unmodelled refraction is likely to be less than that caused by imperfect estimates of camera motion in this particular case of a single stereo pair. The shape of the error surface is perhaps the most interesting result of the simulation – clearly it is not an easily-modeled function, and if one were to attempt to build a large 3-D model from several overlapping sets of images, the accumulated error caused by the unmodelled refraction would quickly lead to inconsistencies in the reconstruction.

V. Calibrating with Refraction

Using the perspective projection camera model with refraction added as a "second layer" makes modelling the entire imaging system much simpler than using a completely general imaging model. Camera calibration can be done in two steps: the first is a traditional in-air perspective calibration, and the second adds the terms accounting for refraction. In both the planar and hemispherical cases, there are three degrees of freedom to account for camera motion inside the pressure housing, and one degree of freedom to account for the thickness of the interface. The hemispherical case also has a degree of freedom for the hemisphere's radius. In practice, the parameters of the interface can be measured directly, and they do not change, so only the three degrees of freedom accounting for camera motion need to be determined.

Fortunately, determining these remaining degrees of freedom is fairly straightforward, using an ordinary (though waterproof) planar checkerboard calibration target with known dimensions. These targets are frequently used for in-air perspective calibration, so requiring one for the second phase of calibrating for refraction is not onerous. The benefit of this technique is that no external measurements need to be known in advance, except for the internal geometry of the checkerboard pattern. In particular, the location of the checkerboard in space relative to the camera is solved for by the optimization, rather than measured by hand. A single image of the target presents nine degrees of freedom that must be found, of which three are relevant for the model (the remaining six capture the location of the target relative to the camera-centric coordinate frame). The image will preferably contain checkerboard corners that are close to the image edges, as this is where the distortion is most prevelant, so pixel estimation errors will be minimized.

Only one image of the target is necessary to calibrate the refraction terms in the model, though multiple views of the target will likely produce better results given the interdependencies between the unknown parameters. The best estimate of the 6-DOF target location for each view depends upon all of the other parameters in the system. This leads to a nested optimization approach, in which the outer optimization attempts to solve the 3-DOF camera location, and the inner optimizations each solve for the target locations given the camera location estimate. Because the image formation process (point-to-pixel operation) is modelled numerically, the optimizations to determine the refraction parameters and target locations are

also carried out numerically (using Levenberg-Marquardt, for example). Each inner optimization step is numerically well-behaved, but the outer optimization needs a reasonable starting guess, because the overall pixel reprojection errors tend to be small. We intend to refine our understanding of the calibration process in future work.

VI. Conclusions

When taking pictures underwater, some degree of refraction is inevitable. We have shown the degree to which refraction leads to image distortion, and subsequent errors in 3-D estimation. For a typical setup, 2-D distortion can be largely compensated for with nonlinear lens distortion terms in the perspective projection model, and 3-D error is small compared to that induced by errors in camera motion estimation and pixel correspondence estimation. But these attempts to minimize refraction-induced distortion will ultimately fail when several images are used together for large-scale photomosaicking and structure from motion modelling. Moreover, we have shown that calibrating an underwater camera including terms that model the pressure housing interface and its refraction effects is not prohibitively difficult.

Perhaps the best reason *not* to use a camera model which accounts for refraction is that a large amount of software exists which relies on the perspective model. The underlying math, and in particular the perspective epipolar constraint, makes working with cameras modelled by perspective projection easier. On the other hand, stereo correspondence algorithms which use the epipolar constraint to limit search will miss matching pixels, especially near the image edges where the error in the perspective model is greatest. One alternative is to use an orthographic, rather than perspective camera model, though these are only approximately valid, and only when using lenses with long focal lengths and narrow fields of view. With an orthographic camera, a planar interface is preferable to a hemispherical interface, because all light rays are assumed to be parallel to each other, and orthogonal to the image plane. Most underwater applications demand the use of wide-angle lenses, however, to capture the maximum amount of seafloor area per image as is practical.

The next step of this research will be to use the physics-based model on underwater imagery to create large-scale photomosaicks and 3-D models using structure from motion. The first step of this process will be a more careful analysis of calibration. We also intend to further refine our understanding of reconstruction error, by individually examining the influences of errors in subpixel correspondence estimation, and in camera motion estimation. Finally, we wish to further reconcile the numerical simulations made here with the more analytical approach of [12] into a general understanding of the underwater imaging process.

References

[1] N. Gracias, S. van der Zwaan, A. Bernardino, and J. Santos-Victor, "Mosaic based navigation for autonomous underwater vehicles," *IEEE J. Ocean. Eng.*, vol. 28, no. 4, pp. 609–624, Oct. 2003.

[2] H. Singh, L. Whitcomb, D. Yoerger, and O. Pizarro, "Microbathymetric mapping from underwater vehicles in the deep ocean," *Computer Vision and Image Understanding*, vol. 79, no. 1, pp. 143–161, 2000.

[3] C. C. Slama, C. Theurer, and S. W. Henriksen, Eds., *Manual of Photogrammetry*. American Society of Photogrammetry, 1980.

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

[5] M. Pollefeys, "3D from image sequences: Calibration, motion and shape recovery," in *Handbook of mathematical models in computer vision*, N. Pargios, Y. Chen, and O. Faugeras, Eds. New York: Springer, 2006.

[6] R. M. Eustice, O. Pizarro, and H. Singh, "Visually augmented navigation for autonomous underwater vehicles," *IEEE J. Oceanic Eng.*, accepted, To Appear.

[7] O. Pizarro, "Large scale structure from motion for autonomous underwater vehicle surveys," Ph.D. dissertation, MIT / Woods Hole Oceanographic Institution, 2004.

[8] R. Gupta and R. I. Hartley, "Linear pushbroom cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 9, Sept. 1997.

[9] J.-Y. Bouguet, "Camera calibration toolbox for MATLAB," http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.

[10] http://www.intel.com/technology/computing/opencv/, "Open source computer vision library."

[11] J. Heikkilä and O. Silvén, "A four-step camera calibration procedure with implicit image correction." in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, June 1997, pp. 1106–1112.

[12] T. Treibitz, Y. Y. Schechner, and H. Singh, "Flat refractive geometry," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Anchorage, Alaska, June 2008.

[13] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, 2000.

[14] R. Hartley, "Estimation of relative camera positions for uncalibrated cameras," in *Proc. European Conf. Computer Vision*, 1993, pp. 579–587.

[15] M. D. Grossberg and S. K. Nayar, "The raxel imaging model and ray-based calibration," *Int. J. of Computer Vision*, vol. 61, no. 2, pp. 119–137, 2005.