Simultaneous 3D Reconstruction for Water Surface and Underwater Scene

Supplementary Material

Paper ID: 1398

1 Video for Fig. 3 in the Paper

Please see the supplementary video[1] named “Fig3.mp4”.

2 Video for Fig. 6 in the Paper

Please see the supplementary video named “Fig6.mp4”.

3 Video for Fig. 7 in the Paper

Please see the supplementary video named “Fig7.mp4”.

4 Forward Projection through Water Surface

As mentioned in Section 5.2 in the paper, the implementation of novel view synthesis requires to project the underwater scene points to the image plane of the evaluation camera through the estimated water surface. As shown in Fig. 1, for each underwater scene point $\mathbf{P}_i$, given the recovered point set $\mathbf{S}$ of the water surface and the camera center $\mathbf{O}$ of the evaluation camera, we aim to estimate the projection (x_i, y_i). Such a forward projection is non-linear because of light refraction at the water surface. Nevertheless, the projection (x_i, y_i) can be easily obtained if the corresponding interface point $\mathbf{X}$ is known.

Previous works [1, 3, 4] in underwater camera calibration propose an iterative procedure to locate $\mathbf{X}$ when the interface can be parametrized (*e.g.* it is flat or cylindrical). Here we present a modification of the method in [3] to locate $\mathbf{X}$ for each underwater point $\mathbf{P}_i$. Algorithm 1 shows our modified algorithm for this task, which differs from the previous method in [3] in two aspects. Firstly, since the interface is assumed to be cylindrical in [3], they use the standard ray-cylinder intersection procedure to intersect a ray with the interface. In comparison, our reconstructed water surface cannot be simply parametrized using a cylinder. We instead apply the ray tracing-based method presented in Section 4.2 in the paper. Secondly, to estimate the normal of an interface point, their method again utilizes the cylindrical parametrization, whereas our modification is based on the local quadratic surface fitting as discussed in Section 4.1 in the paper.

[1] All videos in this supplementary material play at 30fps.

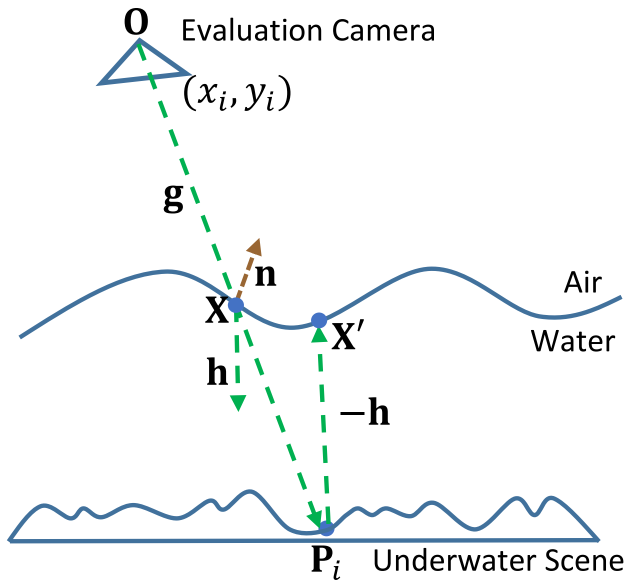

Figure 1: 2D illustration of forward projection through the water surface.

Algorithm 1 Iterative Forward Projection for Point $\mathbf{P}_i$

Input: point set $\mathbf{S}$ of the water surface, underwater scene point $\mathbf{P}_i$, the intrinsic and extrinsic parameters of the evaluation camera, threshold $\epsilon = 10^{-6}$ and $T = 200$

Output: projection (x_i, y_i)

1: initialize $\mathbf{X}$ as the intersection between the water point set $\mathbf{S}$ and ray $\overrightarrow{\mathbf{OP}_i}$
2: **repeat**
3:     compute ray direction $\mathbf{g} := \overrightarrow{\mathbf{OX}}$
4:     estimate the normal $\mathbf{n}$ of $\mathbf{X}$ by fitting a local quadratic surface
5:     compute ray direction $\mathbf{h}$ using Snell's law, given $\mathbf{g}$ and $\mathbf{n}$
6:     shoot a ray from $\mathbf{P}_i$ along direction $-\mathbf{h}$, and this ray intersects with the water surface at $\mathbf{X}'$
7:     **if** the distance $\|\mathbf{X} - \mathbf{X}'\|_2^2 < \epsilon$ **then**
8:         go to Step 13
9:     **else**
10:         $\mathbf{X} := (\mathbf{X} + \mathbf{X}')/2$
11:     **end if**
12: **until** the number of iterations is greater than T
13: project $\mathbf{X}$ to the image plane of the evaluation camera

---

Specifically, we start with connecting point $\mathbf{P}_i$ with the evaluation camera center $\mathbf{O}$ and initializing $\mathbf{X}$ as the intersection between the water surface $\mathbf{S}$ and ray $\overrightarrow{\mathbf{OP}_i}$. We then compute ray $\mathbf{g}$ by connecting $\mathbf{X}$ and $\mathbf{O}$, and estimate the normal $\mathbf{n}$ of $\mathbf{X}$. By Snell's law, the refracted ray $\mathbf{h}$ is computed as:

$$\mathbf{h} = \eta\mathbf{g} + \left(-\eta\mathbf{n}\cdot\mathbf{g} - \sqrt{1 - \eta^2\left(1 - (\mathbf{n}\cdot\mathbf{g})^2\right)}\right)\mathbf{n}, \tag{1}$$

where η is the ratio of the refractive indices of air and water. We set $\eta = \frac{1}{1.33}$ in our implementation. We then shoot a ray from the underwater point $\mathbf{P}_i$ along the negative direction of $\mathbf{h}$ and estimate the intersection $\mathbf{X}'$ between the shot ray and the water surface. If the distance between $\mathbf{X}$ and $\mathbf{X}'$ is larger than a threshold ϵ, we compute the average point $(\mathbf{X} + \mathbf{X}')/2$ as the new value of $\mathbf{X}$ and iterate until their distance is small enough. Finally, we project $\mathbf{X}$ to the image plane of the evaluation camera using the conventional linear projection model [2].

References

[1] J. Belden. Calibration of multi-camera systems with refractive interfaces. *Experiments in fluids*, 54(2):1463, 2013.

[2] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision.* Cambridge university press, 2003.

[3] L. Kudela, F. Frischmann, O. Yossef, A. Uzan, S. Kollmannsberger, Z. Yosibash, and E. Rank. Image-based mesh generation of tubular geometries under circular motion in refractive environments. *Machine Vision and Applications*, Mar 2017.

[4] C. Mulsow. A flexible multi-media bundle approach. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci*, 38:472–477, 2010.