

The Pinax-Model for Accurate and Efficient Refraction Correction of Underwater Cameras in Flat-Pane Housings: MATLAB Examples

Tomasz Łuczynski, Max Pfingsthorn, Andreas Birk.

Read-Me by: Adrienne Winter, African Robotics Unit, University of Cape Town.

The Paper:

A. Agrawal, S. Ramalingam, Y. Taguchi, and V. Chari, “A theory of multi-layer flat refractive geometry,” in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 3346–3353, June 2012.

Dependencies:

Download/Add the Image Processing Toolbox to MatLab. The function “`rigidtfom3d`” that is used in `FindMap.m` is only supported from MatLab version R2022b (R2022a will not recognize this function).

Using the Code:

- Run normal camera calibration in-air with MatLab `cameraCalibrator` or `stereoCameraCalibrator` or any other calibration tool with your camera(s) in their waterproof housings.
- Run `Water_Refraction_Index_Empirical_Eqn.m` (if you have temperature and salinity readings of the water). In the paper ([link](#)) they provide resources for refraction indexes if you don't take your own readings.
- Run `Find_Optimal_d_0/Main.m`.
- Run `Find_Correction_Map/FindMap.m`.
 - You need the camera matrix K , camera distortion coefficients (both calculated during in-air camera calibration), glass/viewport thickness (d_1), glass/viewport refraction index (n_g), water refraction index (n_w), physical d_0 [mm] distance between viewport and camera lens (measured by you) and virtual d_0 [mm] (found by running `Find_Virtual_d0/Find_Virtual_d0.m`).
 - Vector of distortion coefficients $[k_1, k_2, p_1, p_2, k_3, k_4, k_5, k_6, s_1, s_2, s_3, s_4, \tau_x, \tau_y]$ of 4, 5, 8, 12 or 14 elements. k_1, k_2, k_3, k_4, k_5 , and k_6 are radial distortion coefficients; p_1 and p_2 are tangential distortion coefficients; and s_1, s_2, s_3 , and s_4 , are thin prism distortion coefficients. You only need to use the ones that are relevant to your type of camera.

- FindMap.m outputs two .txt files into the same folder it is contained in called MapX.txt and MapY.txt. These are used as inputs to a Python script that uses OpenCV to remap images taken underwater and remove the distortion added by the additional light refraction.
- Run Correct_Underwater_Images/CorrectImages.py (in any IDE – e.g. Visual Studio Code).
 - You may need to change the settings on your code editor (Visual Studio Code 2022 was used for development) to allow longer lines of text for tokenization due to the MapX.txt and MapY.txt files being large. Visual Studio Code Settings -> search for 'Tokenization' -> increase the 'max tokenization line length' until the code works.
 - Note: The Pinax model method removes camera lens distortion (camera intrinsics) and the distortion that occurs from light refracting through water, glass and air interfaces, but it does not stereo rectify the images (they use the word rectify and rectification in the paper, but they are not referring to stereo rectification). You will need to apply stereo rectification after applying this undistortion method if using a stereo camera.