Underwater Multiview Stereo Using Axial Camera Models

Felix Seegräber[1,2][0009−0002−6360−6635], Patricia Schöntag[1][0000−0003−3404−8195], Felix Woelk[2][0000−0002−7963−2654], and Kevin Köser[1][0000−0002−0974−3834]

[1] GEOMAR Helmholz Centre for Ocean Research Kiel, Kiel, Germany
[2] Kiel University of Applied Sciences, Kiel, Germany

Abstract. 3D models, generated from underwater imagery, are a valuable asset for many applications. When acquiring images underwater, light is refracted as it passes the boundary layers between water, housing and the air inside the housing due to the different refractive indices of the materials. Thus the geometry of the light rays changes in this scenario and the standard pinhole camera model is not applicable. As a result, pinhole 3D reconstruction methods can not easily be applied in this environment. For the dense reconstruction of scene surfaces the added complexity is especially challenging, as these types of algorithms have to match vast amounts of image content. This work proposes the refractive adaptation of a PatchMatch Multi-View Stereo algorithm. The refraction encountered at flat port underwater housings is explicitly modeled to avoid systematic errors in the reconstruction. Concepts derived from the axial camera model are employed to handle the high demands of Multi-View Stereo regarding accuracy and computational complexity. Numerical simulations and reconstruction results on synthetically generated but realistic images with ground truth validate the effectiveness of the approach.

Keywords: axial camera · multiview stereo · refractive projection · underwater imagery · 3d reconstruction

1 Introduction and Previous Work

Motivation In the underwater environment, 3D reconstruction is employed for various use cases in research and industry such as 3D maps of underwater habitats and geological structures, identifying and monitoring discarded munition and explosives in the Baltic and North Sea or visual support of autonomous underwater navigation. However, there are unique challenges for image acquisition and 3D reconstruction in the underwater environment one of which is the refractive distortion of light. It travels through multiple interfaces between water, glass and air because cameras are usually placed in dedicated housings with a *flat port* glass interface to protect them from water and pressure. As a result, the pinhole camera model, on which key parts of classical 3D reconstruction algorithms are based, is invalid.

Refractive Distortion in Underwater Environment For some controlled environments it is possible to mitigate a large portion of the refractive effects by calibrating the whole camera setup underwater. This allows the lens distortion parameters to absorb large parts of the refraction for fixed viewing distances [16]. But since the refractive distortion is dependent on the distance of the scene to the camera this approach introduces systematic errors [13,27]. Therefore, a more robust and flexible solution requires explicit modeling of refraction. One approach for compensating flat port refraction is to construct a more extensive compensation model, realized by calculating the distortion per pixel for a fixed distance and storing it in a lookup [19]. Undistorting refractive image data using this lookup allows to return to the pinhole camera model with little error. A major drawback of this method is that it requires very small distances between the center of projection and the housing interface in the order of a few millimeters. Otherwise, the reprojection error for objects not at the calibration distance becomes significant. More generic solutions model the physical path of light rays. It has been shown that with refraction at a flat interface, light rays intersect at a common axis, forming an axial camera model [1]. This holds for a single layer of refraction ("thin glass") and interface-camera alignment [27] also with a more complex system of two layers of refraction ("thick glass") with potential interface tilt [24] and even at a spherical interface, when the camera is not centered inside [25].

Consequently, in this work a refractive Multi-View Stereo algorithm is developed based on insights from the axial camera model. We chose to base our proposed method on the common flat port scenario, two interfaces with potential interface tilt. Agrawal et al. [1] derived the respective analytical projection under refraction. While all models allow dealing with refraction very precisely, they are not easily compatible with in-air reconstruction algorithms, that heavily rely on the characteristics of the pinhole camera model. So it is necessary to make explicit adaptions or design completely new approaches.

The proposed method follows a state-of-the-art PatchMatch approach while replacing key components with newly developed refractive counterparts. Since dense reconstruction requires large-scale matching between a very large number of pixels, computational efficiency is a major concern. Tradeoffs between accuracy and runtime have to be taken into consideration to make the method feasible.

Refractive Multi-View Stereo Reconstruction Multi-View Stereo (MVS) methods usually recover the camera poses and a sparse set of 2D-3D correspondences from image features in a Structure-from-Motion (SfM) step. This data is then used in conjunction with the input images to initialize the MVS algorithm which generates a dense point cloud or even a mesh [7]. While this work does not focus on SfM, the challenges regarding refraction are similar. Chari and Sturm [5] provide a theory for relative pose estimation with single layer refraction. Kang et al. [15] derive specific refractive constraints for single layer refractive SfM and utilize an adapted version [4] of the Patch-Based Multi-View

Stereo (PMVS) algorithm [8] for dense reconstruction. Refractive SfM enabled by refractive bundle adjustment is presented in [14]. The virtual cameras, first introduced for a bundle adjustment error during calibration in [24], inspired the virtual camera homography proposed in this work.

Less attention has been given to dense MVS incorporating refraction. [4, 15] adapt PMVS for refraction, but focus on the SfM part and are not very detailed regarding dense reconstruction. Jordt et al. [11, 12] developed a refractive plane sweep algorithm for dense reconstruction. They avoid computationally expensive refractive forward projections by back projecting and calculating the matching cost for a plane in 3D space. A more recent work [3] tackles Multi-View Stereo for arbitrarily shaped interfaces, targeted on reconstructing e.g. insects trapped in ember. Image pixels are exhaustively matched while replacing projections with solving the 4^{th} degree polynomial for single layer refracive projection. Their proposed solution is not fully operational in its current state, the main inhibitor being its extreme computational complexity. This is a general challenge for Multi-View Stereo with refraction, where the extensive pixel matching over multiple images, in combination with the more computationally expensive projections, result in infeasible runtimes.

Contributions In summary the following contributions are made: A. We introduce local virtual camera homographies between axial cameras and B. an efficient forward projection that is based on discrete line search rather than expensive solving of high-order polynomials. C. Both is used in our suggested extension of state-of-the-art patch-match multi view stereo for refractive scenarios. The effectiveness of the approach is proven by extensive evaluations on ground-truth models with real-world textures using physically-based raytracing.

2 Axial Model with Virtual Camera Approximation

The core concept of PatchMatch Multi-View Stereo is to approximate the scene surface with a collection of planes. Correctness of the planes is scored by warping pixel patches into neighboring views utilizing the homography induced by these planes [23, 26, 29]. Homographies being invalid under refraction as well as the high computational complexity of refractive projection are the main challenges for refractive PatchMatch.

Virtual Camera Homography Because it is not possible to formulate a general homography for two refractive cameras and a plane, we substitute a pinhole camera for single rays of a refractive camera. These virtual cameras, first introduced in [24] for calibration, are located at the intersection of the ray with the camera axis and aligned with the interface normal. When evaluating the photometric consistency between pixels x_i and x_j in two neighboring views, the homography of their respective virtual cameras is utilized. Since the homography is used for a whole pixel patch, an assumption made here is that the virtual

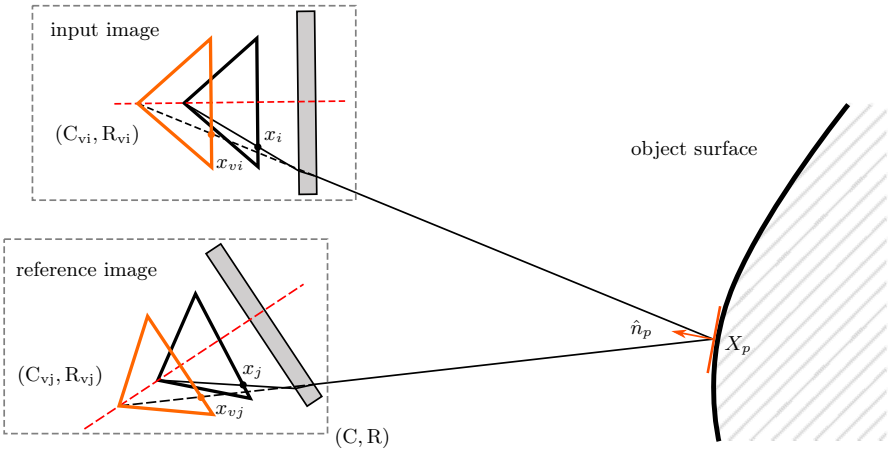

Fig. 1. Refractive PatchMatch, introducing virtual cameras to enable homography warping. The poses of the virtual cameras are given relative to their parent camera.

camera of the central pixel is a good pinhole approximation for all patch pixels. This is plausible since the refractive distortion is minimal between neighboring pixels.

Let the poses of input and reference virtual camera be $\mathrm{C_{vi}}, \mathrm{R_{vi}}$ and $\mathrm{C_{vj}}, \mathrm{R_{vj}}$ respectively. They are both known relative to their parent camera coordinate system. The input camera coincides with the world coordinate system and the reference camera pose is R, C, compare figure 1. The pose of the reference virtual camera relative to the input virtual camera is:

$$\mathrm{C_{vij}} = \mathrm{R_{vi}}(\mathrm{C} + \mathrm{R}^T\mathrm{C_{vj}} - \mathrm{C_{vi}}),\ \mathrm{R_{vij}} = \mathrm{R_{vj}}\mathrm{R}{\mathrm{R_{vi}}}^T. \tag{1}$$

Using these poses, the virtual camera homography is:

$$H_{\mathrm{vij}} = \mathrm{K_{vj}}\left(\mathrm{R_{vij}} - \frac{\mathrm{R_{vij}}\mathrm{C_{vij}}\hat{n}_{vp}^T}{\hat{n}_{vp}^T X_{vp}}\right){\mathrm{K_{vi}}}^{-1}. \tag{2}$$

$\mathrm{K_{vj}}, \mathrm{K_{vi}}$ are the camera matrices of the respective virtual cameras and $X_{vp} = \mathrm{R_{vi}}X_p - \mathrm{R_{vi}}\mathrm{C_{vi}}$, $\hat{n}_{vp} = \mathrm{R_{vi}}\hat{n}_p$ are the plain point and normal in input virtual camera coordinates.

The role of the homography in the PatchMatch cost function is to map pixel coordinates between the two images. To match the pixel coordinates of x and x_v we move the virtual principal point, shifting the virtual image plane, until $x_v = x$. With this requirement, c_{vx}, c_{vy} can be calculated from the ray in water $\vec{X}_w$:

$$c_{vx} = x - f_{vx}x'_{vwx},\ c_{vy} = x - f_{vy}x'_{vwy}. \tag{3}$$

Here x'_{vw} is $\vec{X}_w$ transformed into the virtual camera coordinate system and normalized. The virtual focal length is set to $f_v = f$.

function *AlgorithmicRefractiveForwardProjection*$(X, \mathrm{K}, \hat{n}, L, width, height)$
  $x_p, x_n :=$ `Project(`K, X`)`, `Project(`$\mathrm{K}, \hat{n}$`)`
  $S_{start}, S_{end} :=$ `ClipLine(`$x_p, x_p - x_n, width, height$`)`
  $t := 0.5(S_{end} - S_{start})$
  $x := S_{start} + t$
  while $|t| \geq \sqrt{2}/4$ **do**
    $q :=$ `Round(`x`)`
    $C_v, \hat{X}_v := L(q)$
    $\alpha := ((X - C_v) \cdot \hat{n})/|(X - C_v)|$
    $\beta := \hat{X}_v \cdot \hat{n}$
    $forward := \alpha < \beta$
    if *forward* **then**
      $S_{start} := x$
    end
    $t := 0.5t$
    $x := S_{start} + t$
  end
  if *forward* **and** $|x - S_{end}| < 0.5$ **then**
    return *false*
  else
    return x
  end

Algorithm 1: Algorithmic Refractive Forward Projection

Virtual Camera Lookup An advantage of the virtual camera model is that it can be precalculated from the camera and interface parameters, independently of the scene. The parameters per virtual camera consist of its camera center C_v and the ray vector $\hat{X}_v = \hat{X}_w$, each in the parent camera coordinates, as well as the virtual principal point c_v. Since dimensions of the image and the lookup match, an integer pixel location x can be directly used to get the corresponding virtual camera, i.e. $L(x) = \left\{C_v(x), \hat{X}_v(x), c_v(x)\right\}$. This representation resembles a general camera model, storing a point and direction for each pixel, as proposed by [10].

Algorithmic Refractive Forward Projection Determining the reference virtual camera for the virtual camera homography, requires to find x_j by projecting X_p (c.f. figure 1). The analytical refractive forward projection involves solving a 12^{th} degree polynomial [1], which is computationally expensive. Additionally, the virtual camera lookup is only calculated to pixel precision, giving the opportunity to trade off accuracy for speed with an algorithmic approach.

The algorithmic refractive forward projection 1 exploits that the location of a refractive projection x of 3D point X can be constraint to a 2D ray $R = \{x_p + t(x_p - x_n) | t \geq 0\}$. R lies on the intersection of the plane of refraction [1] with the image plane. x_p is the "naive" pinhole projection of X and x_n the intersection of camera axis and image plane. Under real world conditions the

light is refracted towards the camera axis and x_p is always closer to x_n than x. This can be shown from Snell's law, if the refractive indices of glass, water and air adhere to the inequality $\mu_g > \mu_w > \mu_a$. R is limited to the image boundaries using the Liang–Barsky line clipping algorithm [18] yielding search line S. A point x on S can be tested by comparing $\overrightarrow{C_v(x)X}$ to $\hat{X}_v(x)$. If they are equal, x is the refractive projection of X. If they are not equal, the angles $\alpha(x)$, between $\overrightarrow{C_v(x)X}$ and $\hat{n}$ and $\beta(x)$ between $\hat{X}_v(x)$ and $\hat{n}$ can be used to limit S to one side of the currently tested point. Because the angles between $\hat{X}_v(x)$ and $\hat{n}$ steadily increase on R, if $\alpha(q) > \beta(q)$ the correct refractive projection lies on the upper line segment and vice versa. The search terminates if the search line reaches a length below $\frac{\sqrt{2}}{4}$. This is necessary, since the virtual camera lookup is only calculated to pixel precision, which means that α and β are calculated from a rounded value, resulting in a small error. To determine if a point X projects inside the image boundaries, x is compared to the end of the search line S_{end}. If the last iteration of the loop indicated that the target location is further in the direction of t and the current location hypothesis is close to the endpoint, i.e. $|x - S_{end}| < 0.5$, the projection is considered outside of the image.

3 Refractive PatchMatch

The proposed method extends PatchMatch Multi-View Stereo for refractive images, by integrating the virtual camera homography and algorithmic refractive forward projection into the procedure. As a proof of concept the complete algorithm was integrated into OpenMVS [21] which is largely based on the PatchMatch variant described in [26] but includes several modifications. The following sections present the integration based on this implementation.

Refractive Depth Map For the application in the refractive patch match algorithm it is useful to change the interpretation of values in the depth map. The depth λ_p for a pinhole camera describes the distance between the camera center C and the scene point X along the viewing ray, related by projection: $X = \lambda_p \mathrm{K}^{-1} x'$. To retain the alignment of depth vector and viewing ray under refraction, the interpretation of depth changes to:

$$X = \mathrm{C}_v + \lambda \hat{X}_v, \tag{4}$$

the distance between the virtual camera center C_v and X. Both C_v and viewing ray $\hat{X}_v$ are contained in lookup L, which makes this calculation efficient. Both representations can be easily converted into each other. If λ_r is the refractive depth and λ_p the pinhole depth, the λ_p can be calculated from λ_r as:

$$\lambda_p = |\mathrm{C}_v + \lambda_r \hat{X}_v|. \tag{5}$$

View Selection View selection is based on [9]. It is generally compatible with refraction, but considers crowd sourced images. As this scenario is not typical

underwater, view selection scaling was disabled for refractive images, to avoid having to rescale and recalculate the virtual camera lookups.

Patch Similarity The photometric similarity between two pixel patches in input and reference image is quantified using a Normalized Cross Correlation score, as defined in for example [17]. The patch coordinates of the input image are warped onto the reference image using the current plane hypothesis of the scored pixel. This cost function is adapted by replacing the homography with the virtual camera homography H_v (2). The resulting error is combined with the following measures to a single confidence score for each pixel.

Neighborhood Smoothness To encourage a smooth surface between neighboring pixels, dedicated point and angular smoothness terms are employed. While mostly compatible, the point smoothness Δ_p has to be modified to respect the varying depth interpretation with (4) and (5):

$$\Delta_{vp}(n) = \frac{d_v(n)}{|\mathrm{C}_v + \lambda \hat{X}_v|}, \tag{6}$$

with $d_v(n) = \hat{n}_p \cdot X_{vp}(n) + |\mathrm{C}_v + \lambda \hat{X}_v|$ and $X_{vp}(n) = \mathrm{C}_v(n) + \lambda(n)\hat{X}_v(n)$. $d_v(n)$ is the distance of the current plane hypothesis to the plane point $X_{vp}(n)$ of neighboring pixel n.

Depth Map Consistency Consistency between neighboring views is evaluated using a *forward-backward reprojection error* $|x_i - x_{ij}|$ [28], [23]. x_i, x_j are the current input pixel location and the corresponding reference image location respectively. x_{ij} is determined by back projecting x_j using depth λ_j and forward projecting the resulting 3D point into the input image. To adapt this score for refraction, the pixel distance is calculated in the input virtual camera image:

$$x_{vij} = \mathrm{K}_{vi}\mathrm{R}_v(\mathrm{R}^T \lambda_j \mathrm{K}_j^{-1} x'_j + \mathrm{C} - \mathrm{C}_v). \tag{7}$$

While this introduces some additional error, due to the virtual camera only being an approximation, the distance is expected to be in the range of a few pixels, where the approximation is still good.

Plane Propagation As proposed for example in [23], depth values are propagated by intersecting the viewing ray of the current pixel with the neighboring plane. With the refractive depth map, propagated depth λ_{new} must be the distance between C_v and the neighboring plane $X_{vp}(n), \hat{n}(n)$ in direction of $\hat{X}_v$:

$$\lambda_{new} = \frac{(X_{vp}(n) - \mathrm{C}_v) \cdot \hat{n}(n)}{\hat{X}_v \cdot \hat{n}(n)}, \tag{8}$$

with $X_{vp}(n) = \mathrm{C}_v(n) + \lambda(n)\hat{X}_v(n)$. This way λ_{new} aligns with the refractive depth interpretation of the current pixel, i.e. $X_p = \mathrm{C}_v + \lambda_{new}\hat{X}_v$.

Merging Depth Maps Depth Maps are merged and converted to point clouds by interpolating between the plane points and normals, using the respective confidence score of each point as a weight. 3D points are calculated utilizing the refractive depth interpretation (4). To find corresponding pixel positions in neighboring views, the algorithmic refractive forward projection is used.

4 Evaluation

In this section the algorithmic refractive forward projection and the virtual camera homography are evaluated regarding accuracy and sensitivity to potential error sources. If not noted otherwise, the camera and interface parameters used in the following subsections are: 80° field of view (FOV), centered principal point, distance to interface $d_a = 2$cm, interface thickness $d_g = 1.4$cm with no interface tilt.

Algorithmic Refractive Forward Projection To evaluate the accuracy, a pixel position x is refractively back projected into 3D point X, which is then reprojected into image point x_r, using the algorithmic refractive forward projection. The Euclidean distance $e = |x - x_r|$ between the two image points forms a reprojection error. Figure 2 shows the distribution of e, calculated over a 1000×1000 image, subpixel positions reprojected in 0.1 pixel steps. Three parameters were varied: A higher FOV results in increased incident angles, amplifying the refractive distortion. The distance of the camera to the interface d_a has a strong effect on the distribution of virtual camera locations along the camera axis. Increasing d_a moves the intersections of the rays with the camera axis, i.e. the virtual camera centers, further away from the real camera center, increasing the overall spread along the axis. This effect is also noted in [19], which they call an increase of the "focus section". Finally, tilting the interface normal strongly influences the overall geometry of the system, making it a good candidate as a potential error source. Even with strong variation in these parameters the overall distribution of the error stays similar with a mean of roughly a quarter pixel.

The runtime of the algorithmic projection is dependent on the search line length and therefore on the image size. Nevertheless, tests show that it is still faster than a reference implementation of the analytical version [19] by factors of about 18 to 34 for image sizes of 2 MP to 10 MP.

Virtual Camera Homography The assumption for the virtual camera homography is that a virtual camera is a good approximation even for neighboring rays. To test this, two stereo camera setups are simulated: A generic stereo setup with 1m baseline and 4m plane distance and a setup with a translated ($x = 0.5$m, $z = -1$m) reference camera to limit potential symmetries in the transformation. Figure 3 shows schematics of the arrangement.

Patches were calculated as: 9×9 window size, skipping every second pixel, resulting in 5×5 patches. All image sizes in this section were enlarged by a

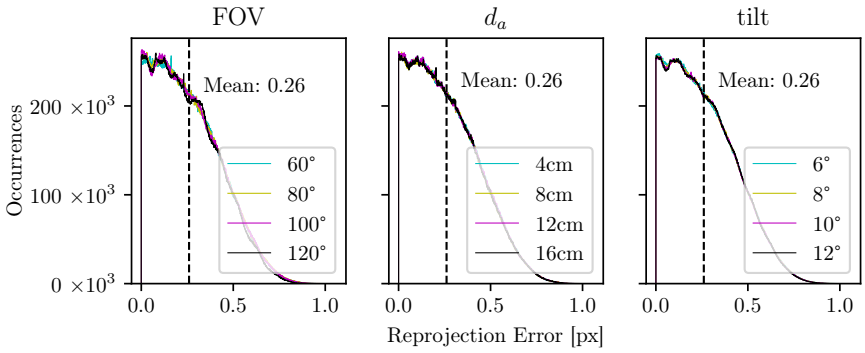

Fig. 2. Reprojection error distributions for different camera setups. For a 1000×1000 image FOV (*left*), interface distance d_a (*middle*) and interface tilt(*right*) are varied.

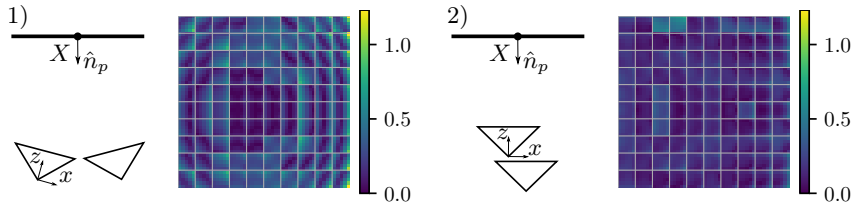

Fig. 3. Setups 1 and 2 for the virtual homography evaluation. The images show the patch pixel error e_H for a 10×10 image in the respective setup. Each cell represents the patch at the specific pixel position, concatenated to form the whole image. The cell contents are the errors of each patch pixel location forming the 5×5 patch.

4 pixel border to compensate for the patch size. To calculate the error for a pixel q in patch B, centered on pixel x, q is warped to the reference image using the virtual camera homography of x, yielding $H_v(q)$. The true value for q is determined by backprojecting it into space, intersecting with the plane and projecting refractively into the reference image to get the true reference position q_r. The distance $e_H = |H_v(q) - q_r|$ is the error for a single pixel in a patch, while $e_{BH} = \frac{1}{n} \sum_{q \in B} |H_v(q) - q_r|$, with n being the number of pixels in B, is the mean error of the whole patch. Note that to determine the reference position q_r in exact values the analytic refractive projection derived in [1], with the implementation from [19] is used.

Figure 3 shows e_H, the error for each patch pixel "exploded" for a 10×10 image: The 5×5 error values for each patch are concatenated forming a cell in the image grid. Each grid cell represents the patch of the respective pixel, the cell contents the error value of the specific patch pixel. The central value in each patch corresponds to the approximated and then rounded refractive projection which was used to look up the reference virtual camera for H_v. In most cases,

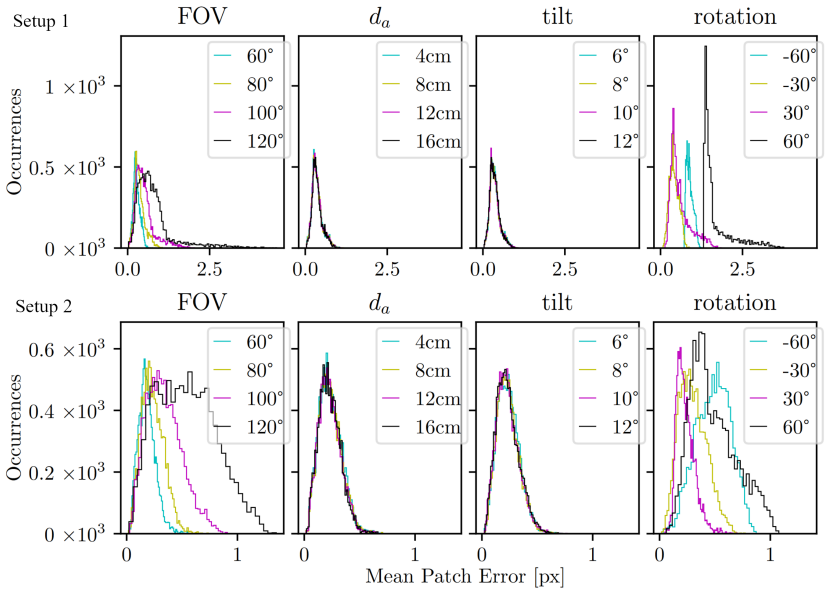

Fig. 4. Mean patch error e_{BH} for setup 1 (*top*) and 2 (*bottom*), 100×100 images. The overall mean of each series from left to right, top to bottom is for the top row.

this rounding error seems to dominate the overall patch error. The average patch error, i.e. the average patch color in figure 3, is mostly close to the central pixels color. Interestingly, for some patches the central pixel does not hold the lowest value in the patch, indicating that the rounding error cancels out the virtual camera error for some patch pixels.

The resulting mean patch errors, for 100×100 images in setup 1 and 2, are shown in figure 4. The same values were used for both cameras in the setup, assuming a Multi-View Stereo scenario with a single camera device. In addition to the parameters varied before, the homography plane normal is rotated around the y-axis for an additional series. For most of the variations, the projection error is a significant contributor to the mean patch error, visible as the distinct patterns in the images. For high field of views or strong plane angles, the error caused by the virtual camera approximation becomes the dominant factor. The more the refractive rays deviate from pinhole rays, the stronger the error seems to be.

Refractive Dense Reconstruction The complete dense reconstruction algorithm, described in section 3, was tested on synthetic images. The images were rendered using the Blender toolbox published in [20]. The 3D model [2] was chosen to depict a geological scene which could potentially be found underwater. Only the geometric effects of refraction were considered. Other water related effects, such as scattering and attenuation, were not modeled. The pre-

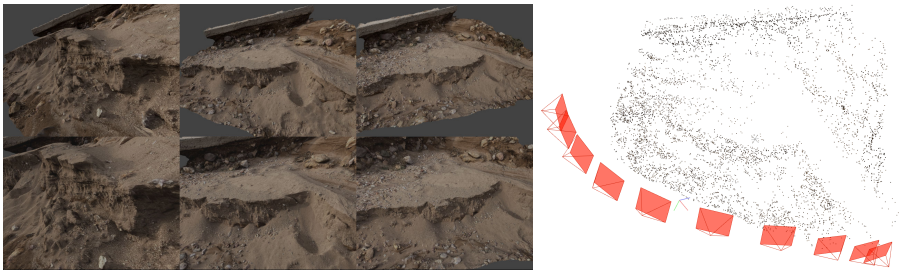

Fig. 5. Excerpt from the synthetic test images (left): Images were rendered from the same poses, with (*bottom row*) and without (*top row*) refraction. Right: Poses and sparse point cloud.

requisite sparse point cloud and camera pose information were generated by the SfM pipeline in COLMAP [22]. Because there is currently no readily available SfM solution which supports refraction, each image was rendered once with and without an refractive flatport interface. Sparse point cloud and pose information were generated from in-air SfM using the unrefracted images. They were then combined with the refracted versions of the images to form the input for the refractive dense reconstruction. A subset of the rendered images, recovered poses and sparse point cloud is shown in figure 5. Images were rendered in 3.1 MP resolution, 2048×1536 pixels. The 10 poses were manually chosen such that most of the image is filled by the rendered model. Since the FOV is effectively reduced by the refraction, the unrefracted images cover more area and contain some background at the borders.

Two reconstructions were run using the images and sparse point cloud as inputs. One with the refracted images and the adapted refractive dense reconstruction procedure and one using the unmodified OpenMVS CPU implementation as a reference. The usual method of comparing the generated depth maps with the ground truth depth maps was not possible. Generating depth maps under refraction is currently not supported with the used renderer. Instead, the reconstruction quality was quantified by comparing the generated point clouds with the original model mesh. As the poses were extracted from the rendered images, the relation to the original model was lost. To compare the point clouds, they were aligned and registered with the model mesh using CloudCompare [6]. The tool also offers the functionality to calculate the distance between the cloud points and the model mesh, which is shown in Figure 6. The model units are unknown, but from measures taken of its content it is plausible that they are meters. The variance of the unrefracted and refracted points is 2.58×10^{-6} and 3.72×10^{-6} respectively. The absolute mean distance is 0.88×10^{-3} for the unrefracted and 1.28×10^{-3} for the refracted points.

The reconstruction results for this set of images is very promising. The density and accuracy is comparable to its in-air counter part. Experiments on real underwater images, incorporating all noise and calibration errors existent in real world data, should be conducted to properly validate the algorithm.

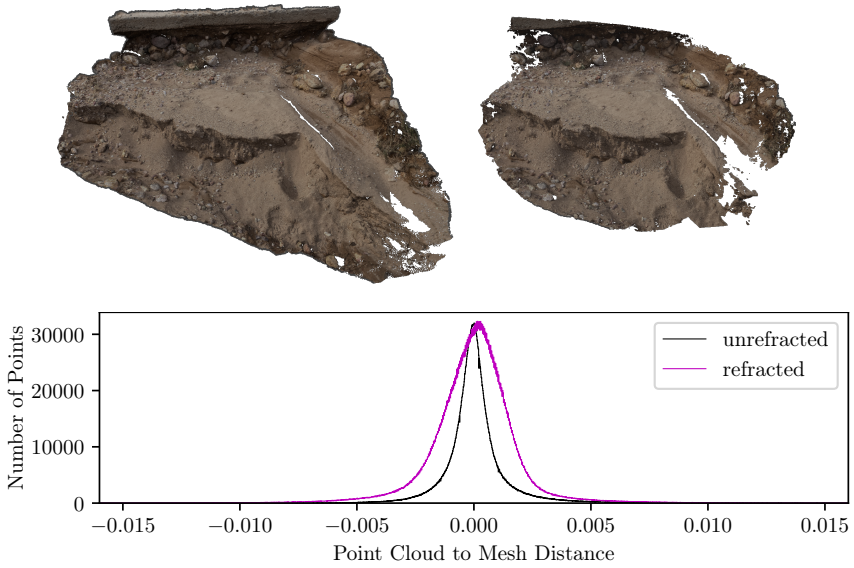

Fig. 6. Reconstructed point clouds (*top*) and point cloud model mesh distance (*bottom*). The refracted point cloud (*top right*) is cut off compared to the unrefracted one (*top left*) because the refraction effectively narrows the FOV. The clouds contain 4,200,659 (unrefracted) and 4,635,708 (refracted) 3D points.

5 Conclusion

We presented a novel approach to refractive Multi-View Stereo. It allows 3D reconstruction of underwater images which suffer from complex refractive distortion due to flat port housings. The proposed method extends a state-of-the-art PatchMatch algorithm with concepts derived from the refractive axial camera model. The refractive matching of pixel patches is enabled by pixel precise pinhole camera approximations stored in a lookup. Utilizing this lookup allows the application of a newly developed efficient algorithmic refractive forward projection. This retains feasible runtimes while maintaining sufficient accuracy to reconstruct high quality dense point clouds. Numerical simulations and a complete refractive dense reconstruction with our method show that the algorithmic refractive projection is accurate to subpixel precision. Density and quality of the generated point cloud is comparable between the refractive and in-air algorithm. Further speed ups can be assumed if an optimized implementation, e.g. using GPUs, is considered in the future. Other potential for future work is in improvement of the virtual camera approximation. Investigating e.g. focal length modification or interpolation between virtual cameras could increase accuracy.

References

1. Agrawal, A., Ramalingam, S., Taguchi, Y., Chari, V.: A theory of multi-layer flat refractive geometry. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE (jun 2012). https://doi.org/10.1109/cvpr.2012.6248073
2. Carena, S.: Foreset beds, spain (3d model) (2022), https://skfb.ly/otPRA
3. Cassidy, M., Melou, J., Queau, Y., Lauze, F., Durou, J.D.: Refractive multi-view stereo. In: 2020 International Conference on 3D Vision (3DV). IEEE (nov 2020). https://doi.org/10.1109/3dv50981.2020.00048
4. Chang, Y.J., Chen, T.: Multi-view 3d reconstruction for scenes under the refractive plane with known vertical direction. In: 2011 International Conference on Computer Vision. IEEE (nov 2011). https://doi.org/10.1109/iccv.2011.6126262
5. Chari, V., Sturm, P.: Multiple-view geometry of the refractive plane. In: Procedings of the British Machine Vision Conference 2009. British Machine Vision Association (2009). https://doi.org/10.5244/c.23.56
6. Cloud Compare Developers: Cloudcompare (2023), http://www.cloudcompare.org/
7. Furukawa, Y., Hernández, C.: Multi-view stereo: A tutorial. Foundations and Trends® in Computer Graphics and Vision **9**(1-2), 1–148 (2015). https://doi.org/10.1561/0600000052
8. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(8), 1362–1376 (aug 2010). https://doi.org/10.1109/tpami.2009.161
9. Goesele, M., Snavely, N., Curless, B., Hoppe, H., Seitz, S.M.: Multi-view stereo for community photo collections. In: 2007 IEEE 11th International Conference on Computer Vision. IEEE (2007). https://doi.org/10.1109/iccv.2007.4408933
10. Grossberg, M., Nayar, S.: A general imaging model and a method for finding its parameters. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001. IEEE Comput. Soc (2001). https://doi.org/10.1109/iccv.2001.937611
11. Jordt, A., Köser, K., Koch, R.: Refractive 3d reconstruction on underwater images. Methods in Oceanography **15-16**, 90–113 (apr 2016). https://doi.org/10.1016/j.mio.2016.03.001
12. Jordt-Sedlazeck, A., Jung, D., Koch, R.: Refractive plane sweep for underwater images. In: Lecture Notes in Computer Science, pp. 333–342. Springer Berlin Heidelberg (2013). https://doi.org/10.1007/978-3-642-40602-7_36
13. Jordt-Sedlazeck, A., Koch, R.: Refractive calibration of underwater cameras. In: Computer Vision – ECCV 2012, pp. 846–859. Springer Berlin Heidelberg (2012). https://doi.org/10.1007/978-3-642-33715-4_61
14. Jordt-Sedlazeck, A., Koch, R.: Refractive structure-from-motion on underwater images. In: 2013 IEEE International Conference on Computer Vision. IEEE (dec 2013). https://doi.org/10.1109/iccv.2013.14
15. Kang, L., Wu, L., Yang, Y.H.: Two-view underwater structure and motion for cameras under flat refractive interfaces. In: Computer Vision – ECCV 2012, pp. 303–316. Springer Berlin Heidelberg (2012). https://doi.org/10.1007/978-3-642-33765-9_22
16. Lavest, J.M., Rives, G., Lapresté, J.T.: Underwater camera calibration. In: Lecture Notes in Computer Science, pp. 654–668. Springer Berlin Heidelberg (2000). https://doi.org/10.1007/3-540-45053-x_42
17. Lewis, J.: Fast normalized cross-correlation. Ind. Light Magic **10** (10 2001)

18. Liang, Y.D., Barsky, B.A.: A new concept and method for line clipping. ACM Transactions on Graphics **3**(1), 1–22 (jan 1984). https://doi.org/10.1145/357332.357333
19. Łuczyński, T., Pfingsthorn, M., Birk, A.: The pinax-model for accurate and efficient refraction correction of underwater cameras in flat-pane housings. Ocean Engineering **133**, 9–22 (mar 2017). https://doi.org/10.1016/j.oceaneng.2017.01.029
20. Nakath, D., She, M., Song, Y., Köser, K.: An optical digital twin for underwater photogrammetry. PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science (Jan 2022). https://doi.org/10.1007/s41064-021-00190-9
21. OpenMVS authors: OpenMVS (2022), https://github.com/cdcseacave/openMVS
22. Schönberger, J.: COLMAP (2022), https://colmap.github.io/
23. Schönberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: Computer Vision – ECCV 2016, pp. 501–518. Springer International Publishing (2016). https://doi.org/10.1007/978-3-319-46487-9_31
24. Sedlazeck, A., Koch, R.: Calibration of housing parameters for underwater stereo-camera rigs. In: Procedings of the British Machine Vision Conference 2011. British Machine Vision Association (2011). https://doi.org/10.5244/c.25.118
25. She, M., Nakath, D., Song, Y., Köser, K.: Refractive geometry for underwater domes. ISPRS Journal of Photogrammetry and Remote Sensing **183**, 525–540 (2022). https://doi.org/https://doi.org/10.1016/j.isprsjprs.2021.11.006, https://www.sciencedirect.com/science/article/pii/S092427162100304X
26. Shen, S.: Accurate multiple view 3d reconstruction using patch-based stereo for large-scale scenes. IEEE Transactions on Image Processing **22**(5), 1901–1914 (may 2013). https://doi.org/10.1109/tip.2013.2237921
27. Treibitz, T., Schechner, Y., Kunz, C., Singh, H.: Flat refractive geometry. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(1), 51–65 (jan 2012). https://doi.org/10.1109/tpami.2011.105
28. Zhang, G., Jia, J., Wong, T.T., Bao, H.: Recovering consistent video depth maps via bundle optimization. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE (jun 2008). https://doi.org/10.1109/cvpr.2008.4587496
29. Zheng, E., Dunn, E., Jojic, V., Frahm, J.M.: PatchMatch based joint view selection and depthmap estimation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. IEEE (jun 2014). https://doi.org/10.1109/cvpr.2014.196