

Towards Monocular Shape from Refraction

Antonin Sulc¹

antonin.sulc@uni-konstanz.de

Imari Sato²

imarik@nii.ac.jp

Bastian Goldluecke¹

bastian.goldluecke@uni-konstanz.de

Tali Treibitz³

ttreibitz@univ.haifa.ac.il

¹ Computer Vision and Image Analysis

University of Konstanz

Konstanz, Germany

² National Institute of Informatics

Tokyo, Japan

³ Charney School of Marine Sciences,

University of Haifa,

Haifa, Israel

Abstract

Refraction is a common physical phenomenon and has long been researched in computer vision. Objects imaged through a refractive object appear distorted in the image as a function of the shape of the interface between the media. This hinders many computer vision applications, but can be utilized for obtaining the geometry of the refractive interface. Previous approaches for refractive surface recovery largely relied on various priors or additional information like multiple images of the analyzed surface. In contrast, we claim that a simple energy function based on Snell's law enables the reconstruction of an arbitrary refractive surface geometry using just a single image and known background texture and geometry. In the case of a single point, Snell's law has two degrees of freedom, therefore to estimate a surface depth, we need additional information. We show that solving for an entire surface at once introduces implicit parameter-free spatial regularization and yields convincing results when an intelligent initial guess is provided. We demonstrate our approach through simulations and real-world experiments, where the reconstruction shows encouraging results in the single-frame monocular setting.

1 Introduction

When we look at objects through a refractive surface like water or glass, the object changes appearance, and its image is distorted (see Fig. 1). This distortion is caused by refraction that happens when light propagates to a medium that transmits light at a different speed. When the ray's path is obstructed with a refractive material, it causes a change of the light ray's trajectory on the interface based on its surface geometry.

This situation happens, for example, when looking into a body of water from above with a drone, or as a lifeguard at a pool. Reconstructing the surface can help compensate for its effects to see through it, and several applications can exploit the surface shape itself, such as sea condition monitoring or refractive object inspection.

While refraction can confuse many computer vision algorithms, the bending of the light also provides hints towards the shape of the surface. In the problem of reconstruction of such surfaces, each surface point has two unknowns: its location and its normal. Previously it was mostly solved in settings with multiple views [13, 16, 18] or multiple frames [21].

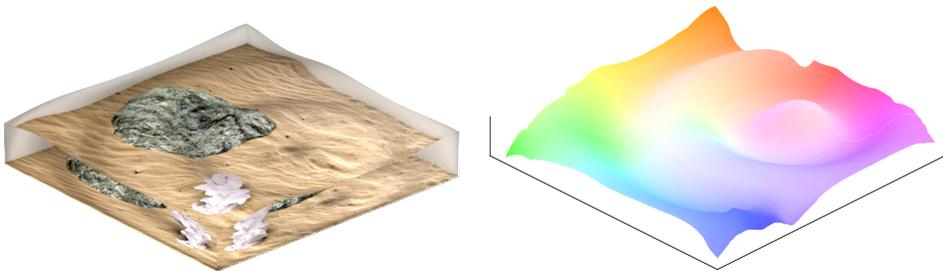


Figure 1: [Left] A simulated scene with a refractive surface. We can observe a significant distortion of the background from the refraction, which is a function of the surface geometry. We exploit this relation and estimate the 3D geometry of the refractive surface from its image, knowing the background. [Right] An example of reconstruction using our approach.

We consider a simpler setup with a single frame monocular camera with a known arbitrary background.

We ask, given an object point and its refracted image, can we determine the light path between them without knowing anything about the surface or its location? While for a single point this problem is ill-posed, we show that solving this problem for all points at once imposes implicit spatial regularization and yields reconstruction of the full 3D of the refractive surface.

We propose an energy formulation such that we seek a surface that explains the refracted projection of the background object using Snell’s law. To overcome the ambiguity, we optimize the whole surface in a single energy, which allows us a calculation of the necessary normal field. Furthermore, our energy works directly in world coordinates and minimizes the geometric errors such that we can obtain a full 3D reconstruction of the sought surface.

Our contributions are:

- We present a method that can estimate a complete 3D geometry of a refractive surface from a single distorted image with knowledge of its background texture and arbitrary depth using a standard off-the-shelf camera.
- Our method relies only on the physics of refraction, and the energy formulation is parameter-free.

The rest of the paper is organized as follows: In Sec. 2 we summarize existing works. In Sec. 3, we show the preliminaries necessary to understand our approach. In Sec. 4 we show how we optimize the unknown refractive surface via our proposed energy and in Sec. 5 we demonstrate several experiments and evaluations on synthetic and real-world scenes.

2 Related Work

A comprehensive analysis of the problem of refractive geometry in computer vision can be found in [1]. Here we review works related to our method.

Multi-view Approaches. Ben-Ezra *et al.* [2] presented a parametric method for shape and pose recovery of transparent objects from a known observer’s motion. The seminal work of Kutulakos *et al.* [3] formulates conditions about the number of views and reference points under which an arbitrarily shaped specular or refractive scene can be reconstructed. They show that a surface of a refractive object can be retrieved from two viewpoints and one

reference point. Chang *et al.* [10] model the distortion from refraction as a depth-dependent function and reconstruct a refracted scene from multiple views. Han *et al.* [9] showed a shape estimation method based on altering a background pattern of an object immersed in water. The work of Morris *et al.* [14] introduces the reconstruction of dynamic refractive surfaces with an unknown refractive index by exploiting the refractive disparity in the multi-frame stereo setting. In [10] refraction through a dynamic surface was treated as a random event happening between the object of interest and the camera. An optimization framework for 3D reconstruction of a transparent object viewed from different viewpoints on a turntable is shown in [13].

A multi-view position-normal consistency was suggested in [16, 17, 18]. The constraint imposes consistency on the normal field between the different views. In [17] this constraint was used to estimate a refractive surface in a stereo setting with a known background texture and depth. In [18] this constraint was used for 3D shape reconstruction of a wavy water surface together with the refracted underwater scene from multiple views.

Radiometric Approaches. Chari *et al.* [8] showed that radiometric clues in addition to the geometric distortion can help for refractive object reconstruction. The method of Ihrke *et al.* [11] used a level set approach to reconstruct a dynamic dyed liquid using a fluorescent substance. In [7] spectral dependency of water is used to reveal object geometry immersed in water from the absorption coefficient by applying Beer-Lambert law.

Monocular Reconstruction. Pioneering work in monocular refractive surface reconstruction was introduced in [15]. They presented an approach for non-rigid transparent shape reconstruction by statistical and geometrical clues from surface motion over a known pattern using tens of frames. The work of Wetzstein *et al.* [2] uses specially designed patterns that can encode angular information of the scene and estimate surface geometry from a single image with the additional angular information provided by the pattern.

A recent learning-based method [20] uses a large set of synthetic images to train a model which can reconstruct complex refractive objects from a single image with a known distant background. Our method recovers the surface using the physics of refraction and does not rely on any training data. Thapa *et al.* [21] use a series of images under orthographic projection to learn the fluid dynamics of water and estimate a height map of a dynamic surface from images with a known background.

Similarly to our method, [19] they present an optimization approach in a monocular setting. However, they reconstruct height-field under an orthographic projection and require a known flat-patterned background. The parallel viewpoint direction in the orthographic projection eliminates the need to estimate the absolute distance of the surface, since moving the surface away from the camera does not influence refraction. Unlike [19], our method estimates the absolute surface depth directly by optimizing reconstruction error on refracted light paths. Our energy optimizes over world coordinates instead of reprojection error in the image plane. This enables us to cope with the realistic setup of an arbitrary background shape, unlike the formulation of [19] which is limited solely to the flat background. Furthermore, optimization in world coordinates allows us to exploit the perspective projection of the refracted rays and therefore avoid the height-ambiguity present in [19, 21].

3 Refraction Through a Surface

Consider a camera whose reference frame is in the origin and pointing towards a positive Z coordinate. The matrix of the intrinsic parameters K of the camera is known. Furthermore,

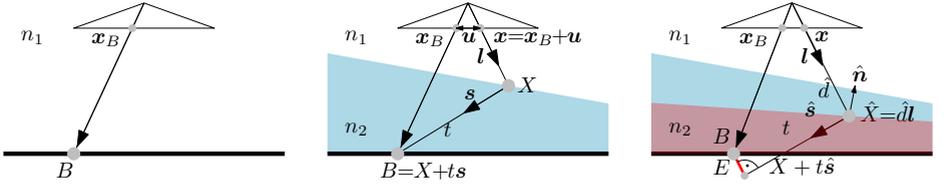


Figure 2: The geometry of monocular imaging through a surface. [Left] Imaging without refraction. A background point at B is projected on an image plane at \mathbf{x}_B to an image plane with color $I(\mathbf{x}_B)$. [Center] Imaging through a homogeneous refractive material between B and the camera. The light emanating from the background point B is refracted and projected to the coordinate \mathbf{x} . In this case, the light path consists of two line segments that go from \mathbf{x} through X and meet with the background point B . [Right] Light path when the estimated refractive surface (red) is different from the correct one (blue). We measure the energy of a proposed solution E as the shortest distance between the estimated light path ($\hat{X} + t\hat{\mathbf{s}}$, see Eq. (2)) and B which should (ideally) be intersected and then $E = 0$. The direction of the refracted ray $\hat{\mathbf{s}}$ is calculated from the normal $\hat{\mathbf{n}}$ and the line-of-sight \mathbf{l} and we know that the correct refracted surface point lies on \mathbf{l} .

consider a known background point B , see Fig. 2 [left]. When the medium between point B and the camera is homogeneous, the path of the ray from the point to the observer (camera) forms a straight line, as is usually assumed in computer vision. Then we denote the image coordinate of B as \mathbf{x}_B .

The situation changes when an object with a different refraction index is placed between the camera and the background. Light passing through the interface between the two media is refracted and the image of the background is distorted. We denote image coordinates of the refracted ray as \mathbf{x} , which is a projection of the 3D point surface point X located on the refractive surface, see Fig. 2 [center].

In our setting, we assume only two homogeneous media, one medium where the observer is located with an index of refraction (IOR) n_1 , the other where is the background with IOR n_2 . With known IORs, we can apply Snell's law [12] which defines how refraction happens between two media. Formally, let us denote in \mathbf{l} the direction of the ray coming out of the camera and hitting the refractive surface at a point that has a normal \mathbf{n} . Then, a ray exits the interface at direction \mathbf{s} , given by Snell's law [12]

$$\mathbf{s} = \text{snell}(\mathbf{l}, \mathbf{n}, \mu) := -\mu\mathbf{l} + \mathbf{n} \cdot \left(\mu\alpha - \sqrt{1 - \mu^2(1 - \alpha^2)} \right), \quad (1)$$

where $\mu = n_1/n_2$ is the ratio of the IORs and $\alpha = \mathbf{l}^T \mathbf{n}$. By knowing the point \mathbf{x} , we can calculate the line-of-sight (LOS) on which the point X must lie. The LOS is an unprojection of a point \mathbf{x} ,

$$\mathbf{l} \doteq K^+ \mathbf{x}. \quad (2)$$

The critical yet straightforward fact is that the point X must lie on its corresponding observed LOS \mathbf{l} . Given \mathbf{l} we know that for a distance d we can calculate a point X

$$X(d) = d\mathbf{l}, \quad (3)$$

i.e., X lies on the ray \mathbf{l} at a distance of d from the camera origin where $X = [x, y, z]^T$ is a surface point with a depth z . Combining Eq. (1) and Eq. (3) yields the expression for the

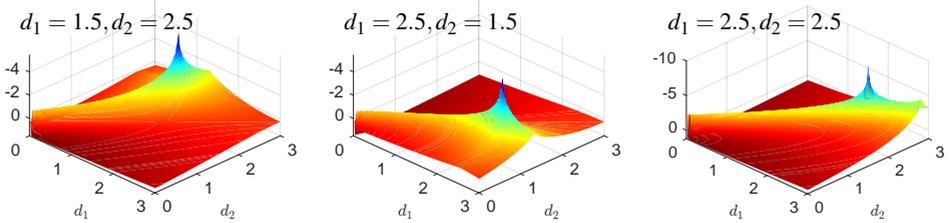


Figure 3: A minimal synthetic example of the proposed energy E for a two-point surface d_1, d_2 . The background plane is placed at $B^z = 3$ and the shown surfaces are energies of different combinations of d_1, d_2 with their ground truth location shown above. The values are the logarithm of E for better visibility. All energies show that the ground truth solution yields a global minimum. However, notice that the magnitude of the gradient gets very small around the location of the correct solution. Similarly, there is a ridge along a scalar multiple of the depth. The peak in the energy slightly deviated from the ground truth, because the normals are calculated from the numerical approximation of surface derivatives.

ray’s light path that connects X and B

$$B = X + t\mathbf{s} , \quad (4)$$

where t is the distance from X to the background point $B = [x^B, y^B, z^B]^T$, see Fig. 2 [center].

4 Optimization

The input to the method is I_B , an image of the background object acquired through a homogeneous medium (i.e. air) including its depth B for each pixel, and I_R , an image of the background object acquired through the refractive surface from the same viewpoint. Then, dense matches are found between these two images, which is done by estimating the flow field \mathbf{u} such that $I_B(\mathbf{x}) = I_R(\mathbf{x}_B + \mathbf{u})$.

Once the dense matches \mathbf{u} matches are found, the optimization solves for the refractive surface by finding the \mathbf{d} along the known LOS \mathbf{l} for each object point. The optimization aims to find the values \mathbf{d} for all points such that their light paths will pass through their respective background points B (see Fig. 2 [center]).

In each iteration of the optimization, we perform the following steps: 1) Calculating the surface point cloud (PC) using Eq. (3) with the current estimate of the distances \mathbf{d} . 2) Calculating the normals \mathbf{n} from the entire PC. 3) Calculating \mathbf{s} using Eq. (1) and reconstructing the estimated light path in Eq. (4).

The steps are performed in each update of \mathbf{d} until a stop criterion on the value of the energy is met.

4.1 The Energy

In this part, we describe the energy we optimize to obtain surface geometry. We first show how we measure the error of the depth map when it does not form a correct light path.

In Sec. 3 we said that B has to lie on the trajectory defined in Eq. (4) (see Fig. 2 [center]). During the estimation of the surface, we reconstruct light paths using Eq. (4), which should ideally intersect with B . For an incorrect \mathbf{d} , the path does not intersect with its B , and we

penalize the estimate by its smallest distance of the light path to its B . Note that we have to reconstruct light paths for the entire image since normals are needed to calculate \mathbf{s} in Eq. (1).

The situation is schematically shown in Fig. 2 [right], where the light path does not intersect with the background point B . In this case, the energy of a point is calculated as the shortest distance between a light path in Eq. (2) and its background point, B . This point-line distance has a following closed form,

$$E(X, \mathbf{s}, B) := \|B - [X + [\mathbf{s} \cdot (B - X)]\mathbf{s}]\|_2 \quad (5)$$

where \cdot is a dot product and X is given by Eq. (3). The total energy we optimize for all points jointly is

$$\mathbf{d}^* = \arg \min_{\mathbf{d}} \sum_{i \in \Omega} E(X_i, \mathbf{s}(I_i, \mathbf{n}_i, \mu), B_i) + \max\{0, z_i - z_i^B\} \quad (6)$$

where individual normals \mathbf{n}_i are calculated from all points in PC Ω of all surface points $X_i(\mathbf{d})$ in Eq. (3). The second term prevents estimated depth z_i to fall behind the background depth z_i^B . Note that the normals and PC are calculated in the iteration of the optimization step, but only \mathbf{d} is updated.

Fig. 3 shows a simple example of our energy Eq. (5) with three different combinations of point distances of two adjacent points. The energies are calculated on a grid of distances. Each image shows that even when only one of the points is correct, the energy is negatively affected as the points are spatially linked by normals. This link imposed by normals affects neighbouring points, and the lower energy is always at the location which matches the true value for both points. The spatial linking through normals introduces the only regularization in our energy. It imposes physical correctness on the entire depth map because it is linked to other surface points through normals which are input for Snell's law. Moreover, the figure reveals that there is a significant drop in the gradient magnitude when a solution is either close to a correct solution or its scalar multiple of depth, which corresponds to the ridge in the energy surface shown in Fig. 3.

Initialization. The energy is non-convex and relies on a good initial guess, like coarse information about the surface distance. For this purpose, we developed an initialization scheme where we optimize a location of a plane perpendicular to the camera's principal axis. The plane location from the observer is defined by a single variable, a shift, which we can easily and quickly optimize with the energy, i.e., we optimize for c after plugging $\mathbf{d}_{\text{flat}} = c$ as the depth to the energy in Eq. (6). With the initial estimate, the location of the plane is usually roughly located around the mean position of the ground truth surface and serves as a good initial guess.

5 Experiments

To showcase our method, we compare it quantitatively to benchmarks with available ground truth [18, 21] and make a series of real-world experiments to estimate a surface of a liquid with a standard off-the-shelf camera. We have chosen a CPU Matlab implementation of L-BFGS [24] which does not require the calculation of the memory-prohibitive Jacobian or Hessian as the number of the variables equals the number of surface points. Run time of one energy evaluation is on average 2.06 ms for 64×64 px images, 7.26 ms for 128×128 px and 25.32 ms for 256×256 px images. For dense matches between background and refracted surface \mathbf{u} we used the optical flow [6]. We optimize the energy for the entire input surface

	Ours - indep. init		Ours - seq. init		Perspective Ours - $d_0 = 2$		9 Cameras [18]		4 Cameras [21]	
	RMSE [unit]	MAE [°]	RMSE [unit]	MAE [°]	RMSE [unit]	MAE [°]	RMSE [unit]	MAE [°]	RMSE [unit]	MAE [°]
	<i>wave1-z_{flat}</i>	0.15	5.89°	0.11	4.29°	0.05	5.06°	0.006	0.76°	0.014
<i>wave1-z_{func}</i>	0.14	6.06°	0.10	4.79°	0.05	4.80°				
<i>wave2-z_{flat}</i>	0.23	11.16°	0.45	13.28°	0.04	5.89°				
<i>wave2-z_{func}</i>	0.23	10.85°	0.43	13.18°	0.04	5.92°	0.002	0.37°	-	-

	Orthographic					
	Ours - indep. init		Full [21]		Single-input [21]	
	RMSE	MAE [°]	RMSE	MAE [°]	RMSE	MAE [°]
<i>ocean</i>	0.230	1.301°				
<i>ripple</i>	0.217	0.649°	0.126	-	0.262	-
<i>tian</i>	0.248	1.364°				

Table 1: Quantitative comparison of reconstructed surfaces with a multi-view [18] and a monocular method [21]. The upper part of the table shows results on two benchmark functions *wave1* and *wave2* in perspective setting. To demonstrate that our method can handle different background depths, we conducted experiments with two different background depths defined by z_{flat} and z_{func} . The table shows three different initialization schemes, the first column shows results when each frame is initialized separately, the second the sequential initialization when only a first frame is initialized and consecutive frames use the previous result and the third is where all frames use the same initial $d_0 = 2$. The lower part shows evaluation of three different surfaces (*ocean*, *ripple* and *tian*) compared to [21] (monocular setting under orthographic projection). Each of these surfaces consists of a ten-frame sequence generated with the code provided by [21].

simultaneously, but alternatively, the optimization step can be performed patch-wisely. The patch-wise computation can however hurt the quality in non-overlapping regions.

Synthetic Experiments. We compare our method to two methods, a multiview method [18] and a monocular method [21].

In the first experiment, we use the same benchmark functions presented in [18] with a perspective projection. The benchmarks are denoted *wave1* and *wave2*. The surface *wave1* is defined as $z_1(x, y, t) = 2 + 0.1 \cos[\pi(t + 50) \sqrt{(x-1)^2 + (y-0.5)^2} / 80]$. It can be intuitively seen as a diagonally rolling wave. The second benchmark *wave2*, $z_2(x, y, t) = 2 - 0.1 \cos[\pi(t + 60) \sqrt{(x+0.05)^2 + (y+0.05)^2} / 75]$ can be seen as a wave growing in the center. Since surfaces are smooth, a resolution of 64×64 px is sufficient to represent the surface. Scenes are rendered with Blender [5] with Cycles Engine. The rendering scripts and input images will be made available with the paper. Both functions are evaluated on 100 surfaces in $t \in (0, 99)$. We tested both benchmark functions with two background shapes to show that background geometry can be arbitrary. The first, a plane at $z_{\text{flat}} = 2.5$ and the second, a more complex shape, $z_{\text{func}}(x, y) = 2.5 + 0.05[\sin(2\pi x) + \cos(2\pi y)]$.

The results of the first experiment are shown upper part of Tab. 1 and Fig. 4. We show results for three initialization schemes, the independent initialization is done for each frame t independently and sequential initialization uses the previous estimate as the initial value, which is similar to the technique used in [18]. Lastly, to demonstrate that our method works well when an initial surface location is provided, we initialized the method with the surface location at $d_0 = 2$, which roughly corresponds to a mean location of the sought surface. Error is reported in both Root Mean Square Error (RMSE) of the reconstructed depth map and Mean Angular Error (MAE) of the estimated normals. We used the same calibration as [18] therefore the RMSE is measured in the same (arbitrary) units.

The errors vary with time as the surfaces become more complex, but the shape of the background surface has little influence on the results. We see that our method is sensitive to

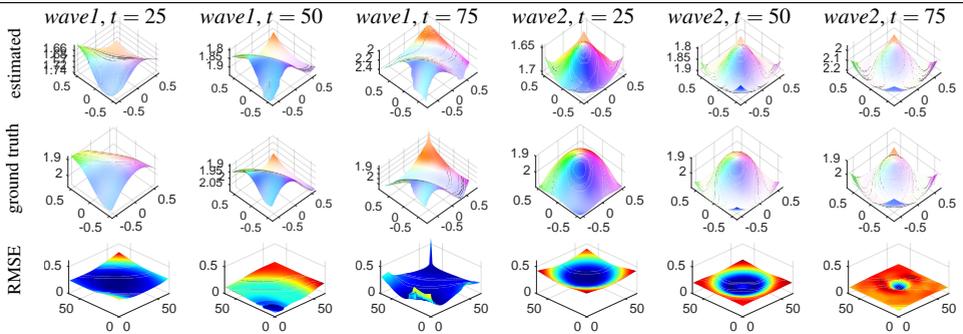


Figure 4: Results of the *wave1* and *wave2* with z_{func} background at times $t \in \{25, 50, 75\}$. Initialization was done for each frame independently. Estimated [top] and ground truth surfaces [middle] with colour-coded [B] optical flow [B]. The bottom row shows a colour coded error of the estimated depth map.

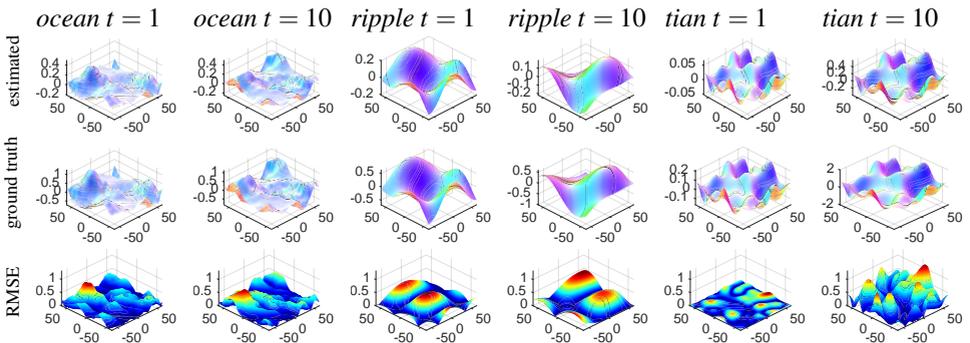


Figure 5: Estimated surfaces of *ocean*, *ripple*, *tian* of surfaces generated from [21] under an orthographic projection. Estimated [top] and ground truth surfaces [middle] with colour-coded [B] given optical flow [B]. The bottom row shows a colour coded error of the estimated depth map. Due to the height-ambiguity imposed by the problem, the comparison is performed with surfaces normalized to have a zero mean.

initialization. There is a noticeable difference in RMSE when a good initial value is provided ($d_0 = 2$, roughly the location of the correct surface). Results differ particularly on *wave2*, where RMSE is approximately six times better when a good initialization is used. A less noticeable gap in results is apparent on *wave1* where MAE is even better when sequential initialization is used than in the case of the good initial guess ($d_0 = 2$) but still worse in RMSE. Note that we rely on a monocular setting with just a predefined background. In comparison with the nine-camera setting of [18], our method performs worse both in RMSE and MAE. Since the benchmark surfaces are largely non-flat, our method sometimes estimated the mean location of the surface slightly off but recovered the normals comparatively accurately to the four camera settings in [18] on *wave1* benchmark. However, in the reduced setting with four cameras [18], our method is on par in MAE, but is still worse in RMSE.

The second experiment compares our method to [21] in a monocular setting with orthographic projection. We used the code provided by [21] to generate the surfaces and their optical flow. We rendered ten frames of the three available scenarios *ocean*, *ripple* and *tian* with 128×128 px under orthographic projection. Note that there is a height-ambiguity im-



Figure 6: [Left] The input background texture we used in our real-world experiment. [Center] One of the input images of a refracted surface (denoted as Frame 6 in Fig. 7). [Right] Optical flow between these frames.

posed by the orthographic projection, as a shift of the entire surface does not change the refracted ray \mathbf{s} since the angle subtended between \mathbf{l} and \mathbf{n} does not change with by moving the surface in Z coordinate, therefore the estimated surfaces are normalized to have zero mean before their RMSE is calculated. For the same reason, all frames were initialized identically with an arbitrary flat surface at $\mathbf{d}_0 = 1$.

Our method quantitatively performs better in RMSE than [21] in a single frame setting. For the full multi-frame FSRN-RNN [21] the RMSE error is half compared to ours but this method takes into account fluid dynamics between subsequent frames, unlike our method which works independently on each frame. It is important to stress that in [21] a flat background at $z = 0$ is required, while in our case the background depth can be arbitrary. Furthermore, [19, 21] are limited only to orthographic projection and thus can only estimate height-field, while our approach is more general and can estimate a 3D surface under the perspective setting.

In both experimental setups, our method is efficient in finding the correct surface shape as is evident in the low MAE. Most of the RMSE error stems from errors in the shift and scale of the surface, as discussed in Sec. 4.

Real-World Experiments. We further evaluated our approach on real-world scenes using a water tank and a flat surface with a printed complex texture as the background (shown in Fig. 6 [left]). We used a Nikon D810 camera with a Nikon AF-S Nikkor 105 mm lens. Camera calibration was performed with the MATLAB Camera Calibration Toolbox. We also used the results of the depth of the intrinsic calibration marker to measure the depth of the flat background surface. The optical flow was estimated with [6]. We fixed the camera to a steel custom-made structure approximately 630 mm away from the background (an averaged distance of the flat calibration pattern) where the background is perpendicularly aligned to the principal axis of the camera. The textures contain many fine details, which are particularly good for finding the dense matches between the background image and the images distorted by refraction. We show the background and one refracted input image in Fig. 6 [left] and the estimated surfaces in Fig. 7.

We perturbed the surface by hand to make different types of waves with varying smoothness. Frames 1 and 2 in Fig. 7 show reconstructions of smooth surfaces with rather larger waves. Frame 4 showcases a situation with a slightly larger depth range where we shook the surface. Lastly, frames 5 and 6, in contrast, show waves with rather sharper edges of smaller depths. Since the real-world surface is mostly flat, the initial estimate is often accurate enough and produces convincing depths.

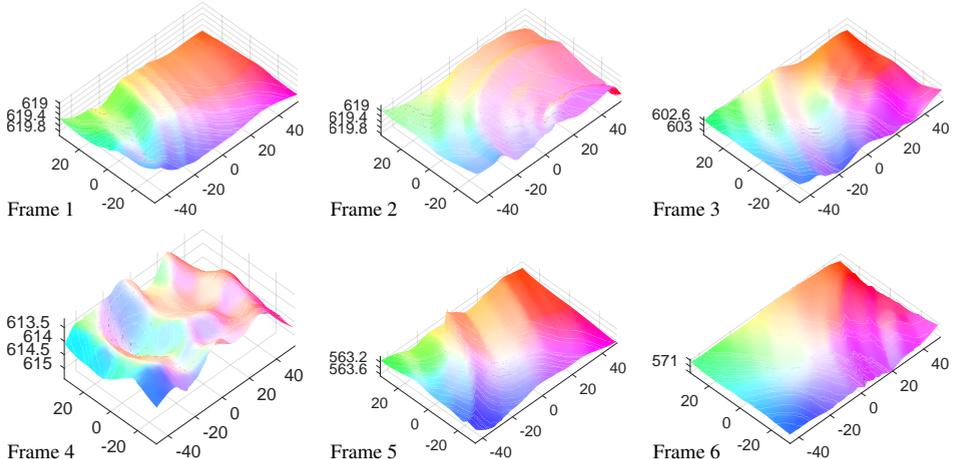


Figure 7: Estimated surfaces of our real-world water experiment. Fig. 6 shows input images. Surfaces are shown with their corresponding colour-coded [9] optical flow [8]. The experiment was made with a water tank with a fixed camera acquiring the scene from above. Apart from making artificial waves on the surface, the amount of liquid was also changed to vary the overall depth. Because the magnitude of depth is significantly smaller than surface width and height, the shown surface depths are re-scaled 20 times for better visibility.

6 Conclusions

A refractive object can hinder many conventional computer vision methods. Nevertheless, refraction can teach us about the physics of the scene and pose interesting challenges. Here we looked at the task of reconstructing a refractive surface over a known background using just a single view. This setup was never tackled before in the general setup of a perspective camera and an arbitrary background shape.

Unlike previous papers that used multiple views, frames, or additional constraints, we showed that a simple energy function based on Snell’s law can estimate a surface under a monocular setup when background depth is given. Working directly at world coordinates, we express our energy as a distance between the original background point and the expected ray’s trajectory. The energy looks for the surface that best explains the refracted image altogether. By optimizing for the entire surface, implicit smoothness is imposed, and the surface shape can be reconstructed.

We demonstrate our method on numerous synthetic and real-world scenes with varying complexity. We achieve errors that are on par in MAE with a previous method that used four views and similarly accurate in RMSE with a deep learning single-frame monocular setup under orthographic projection.

Since our method is indirectly optimizing the depth through normal fields, it is highly sensitive to the initial depth and performs better in reconstructing surface normals than reconstructing absolute depth. For initialization, we developed a simple initialization scheme, which works robustly on mostly flat surfaces. Our method is based on calculating the optical flow between the refracted image to the original one. Therefore, it is limited to scenarios where the optical flow can be reliably calculated, i.e., enough texture and moderate distortions from the refraction.

Acknowledgement

This work was supported by the grant “Internationalisierung Exzellenzstrategie” from University of Konstanz, “SFB Transregio 161 - Quantitative Methods for Visual Computing (project B5)”, Leona M. and Harry B. Helmsley Charitable Trust, The Maurice Hatter Foundation, Israel Science Foundation grant #680/18 and the Technion Ollendorff Minerva Center for Vision and Image Sciences.

References

- [1] Marina Alterman, Yoav Y Schechner, and Yohay Swirski. Triangulation in random refractive distortions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 39(3):603–616, 2016.
- [2] Yuta Asano, Yinqiang Zheng, Ko Nishino, and Imari Sato. Shape from water: Bispectral light absorption for depth recovery. In *Proc. European Conf. on Computer Vision*, pages 635–649. Springer, 2016.
- [3] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *Int. J. of Computer Vision*, 92(1), 2011.
- [4] Moshe Ben-Ezra and Shree K Nayar. What does motion reveal about transparency? In *Proc. Int. Conf. on Computer Vision*, 2003.
- [5] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam, 2020. URL <http://www.blender.org>.
- [6] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(3):500–513, 2010.
- [7] Yao-Jen Chang and Tsuhan Chen. Multi-view 3D reconstruction for scenes under the refractive plane with known vertical direction. In *Proc. Int. Conf. on Computer Vision*, 2011.
- [8] Vishes Chari and Peter Sturm. A theory of refractive photo-light-path triangulation. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 1438–1445, 2013.
- [9] Kai Han, Kwan-Yee K Wong, and Miaomiao Liu. A fixed viewpoint approach for dense reconstruction of transparent objects. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2015.
- [10] Ivo Ihrke, B Golduecke, and Marcus Magnor. Reconstructing the geometry of flowing water. In *Proc. Int. Conf. on Computer Vision*, volume 2, pages 1055–1060, 2005.
- [11] Ivo Ihrke, Kiriakos N Kutulakos, Hendrik PA Lensch, Marcus Magnor, and Wolfgang Heidrich. Transparent and specular object reconstruction. In *Computer Graphics Forum*, volume 29. Wiley Online Library, 2010.

- [12] Michael J Kidger. *Fundamental optical design*. SPIE Bellingham, 2001.
- [13] Kiriakos N Kutulakos and Eron Steger. A theory of refractive and specular 3D shape by light-path triangulation. *Int. J. of Computer Vision*, 76(1), 2008.
- [14] Nigel JW Morris and Kiriakos N Kutulakos. Dynamic refraction stereo. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(8):1518–1531, 2011.
- [15] Hiroshi Murase et al. Surface shape reconstruction of a nonrigid transparent object using refraction and motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(10):1045–1052, 1992.
- [16] Yiming Qian, Minglun Gong, and Yee Hong Yang. 3D reconstruction of transparent objects with position-normal consistency. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 4369–4377, 2016.
- [17] Yiming Qian, Minglun Gong, and Yee-Hong Yang. Stereo-based 3d reconstruction of dynamic fluid surfaces by global optimization. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2017.
- [18] Yiming Qian, Yinqiang Zheng, Minglun Gong, and Yee-Hong Yang. Simultaneous 3D reconstruction for water surface and underwater scene. In *Proc. European Conf. on Computer Vision*, 2018.
- [19] Qi Shan, Sameer Agarwal, and Brian Curless. Refractive height fields from single and multiple images. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2012.
- [20] Jonathan Stets, Zhengqin Li, Jeppe Revall Frisvad, and Manmohan Chandraker. Single-shot analysis of refractive shape using convolutional neural networks. In *Winter Conference on Applications of Computer Vision*, 2019.
- [21] Simron Thapa, Nianyi Li, and Jinwei Ye. Dynamic fluid surface reconstruction using deep neural network. In *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2020.
- [22] Gordon Wetzstein, David Roodnick, Wolfgang Heidrich, and Ramesh Raskar. Refractive shape from light field distortion. In *Proc. Int. Conf. on Computer Vision*, 2011.
- [23] Bojian Wu, Yang Zhou, Yiming Qian, Minglun Gong, and Hui Huang. Full 3D reconstruction of transparent objects. *arXiv preprint arXiv:1805.03482*, 2018.
- [24] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. on Mathematical Software (TOMS)*, 23(4):550–560, 1997.