NeReF: Neural Refractive Field for Fluid Surface Reconstruction and Rendering

Ziyu Wang, Wei Yang, Junming Cao, Qiang Hu, Lan Xu,
Junqing Yu, and Jingyi Yu, *Fellow, IEEE*

Abstract—We present a novel Neural Refractive Field (NeReF) to recover wavefront of transparent fluids by simultaneously estimating the surface position and normal of the fluid front. Unlike prior arts that treat the reconstruction target as a single layer of the surface, NeReF is specifically formulated to recover a volumetric normal field with its corresponding density field. A query ray will be refracted by NeReF according to its accumulated refractive point and normal, and we employ the correspondences and uniqueness of refracted ray for NeReF optimization. We show NeReF, as a global optimization scheme, can more robustly tackle refraction distortions detrimental to traditional methods for correspondence matching. Furthermore, the continuous NeReF representation of wavefront enables view synthesis as well as normal integration. We validate our approach on both synthetic and real data and show it is particularly suitable for sparse multi-view acquisition. We hence build a small light field array and experiment on various surface shapes to demonstrate high fidelity NeReF reconstruction.

Index Terms—Computational Photography, Fluid Reconstruction, Implicit Representation

1 Introduction

Modeling and reconstruction of refractive surfaces from photographs have great importance for applications ranging from fluid physics analysis, environmental monitoring to computer graphics. Refractive surfaces poses exceptional challenges as a light ray only diverts from its straight path when traversing the air-fluid interface which is invisible, hence it's difficult to directly recover the surface.

A common non-intrusive approach for estimating the shape of fluids is to analyze the distortions of a reference pattern placed under the fluid [1]. In particular, many approaches rely on imposing additional assumptions, such as pattern appearance, water height [2], [3], [4] and optics [5], [6], [7], while others create dedicated imaging/optics systems (e.g., camera array [8], Bokode [7] and light field probe [9]) for acquiring fluid structures. Morris et al. [2] introduce a novel refractive disparity for water surface recovery using "stereo matching". Qian et al. [3] use a camera array to estimate both water surface and the underwater scene through exploiting the surface normal consistence. Xiong and Heidrich [10] propose a novel differentiable framework to reconstruct the 3D shape of underwater environments from a single, stationary camera placed above the water. Notably, Thapa et al. [11] proposes learning-based single-image approach with recurrent layers modeling spatio-temporally consistence to recover dynamic fluid surfaces.

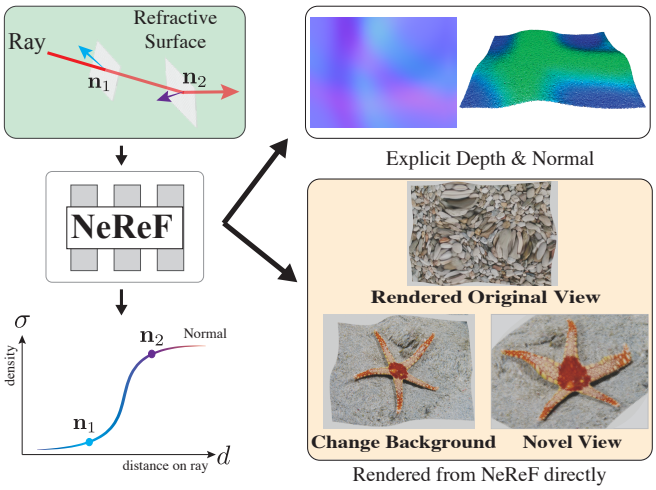


Fig. 1. Our Neural Refractive Field (NeReF) is formuated to recover a volumetric normal field with its corresponding density field. Despite the explicit depth and normal representation can be derived from NeReF, we can render refraction effect and synthesize novel views directly from NeReF.

Existing approaches unanimously model the air-fluid interface as one layer of geometry surface, which usually is explicitly represented by depth and normal maps. Re-rendering of the refraction effect from depth and normal requires ray tracing with hit point testing and two-bounce recursion. In this paper, we propose a Neural Refractive Field (NeReF) to implicitly represent a fluid surface, by taking the advantage of the recent success in neural implicit scene representation. Our work is inspired by the Neural Radiance Field (NeRF). More specifically, we use a fully-connected deep network to represent the fluid surface, whose input is a 3D coordinate and outputs the volume density and normal

- *Ziyu Wang is with the School of Information Science and Technology, ShanghaiTech University, Shanghai, 201210, China. E-mail: wangzy6@shanghaitech.edu.cn*
- *Wei Yang and Junqing Yu are with Huazhong University of Science and Technology, Wuhan, 430074, China. E-mail: {weiyangcs, yjqing}@hust.edu.cn*
- *Junming Cao is with Shanghai Advanced Research Institute, and also with ShanghaiTech University, Shanghai, 201210, China. E-mail: caojm@sari.ac.cn*
- *Qiang Hu, Lan Xu and Jingyi Yu are with the School of Information Science and Technology, ShanghaiTech University, and also with the Shanghai Engineering Research Center of Intelligent Vision and Imaging, Shanghai, 201210, China. E-mail: {huqiang,xulan1,yujingyi}@shanghaitech.edu.cn*

at the coordinate. We can perform re-rendering of refraction effects directly from the implicit representation by integrating normal along a target ray and deflecting it according to Snell's law. Explicit representations, such as depth and normal, can still be recovered effectively via volume integration (Fig. 1). In contrast to supervised approaches [7], NeReF optimization is performed in a per-scene manner and hence avoids the lack of real dataset problem. Moreover, NeReF models a continuous function and generates results at resolutions on demand. And NeReF model is relatively smaller (about 4MB), the advantage expands as the desired resolution increases. View synthesis from NeReF is easier compared to ray tracing. To evaluate our proposition, we construct a multi-camera system similar to [12] and capture a known pattern under dynamic fluid surface. We use the optical flow [1] technique to obtain the groundtruth point-ray correspondences for NeReF optimization. Experiment results show our training free approach can recover the fluid surface with high fidelity. Our system setup is shown in Fig. 4.

Compared to the original NeRF which models radiance along rays, we demonstrate that the geometric information (normal of refractive surface) of the ray can also be implicitly encoded in a neural field. To the best of our knowledge, we are the first to demonstrate this property.

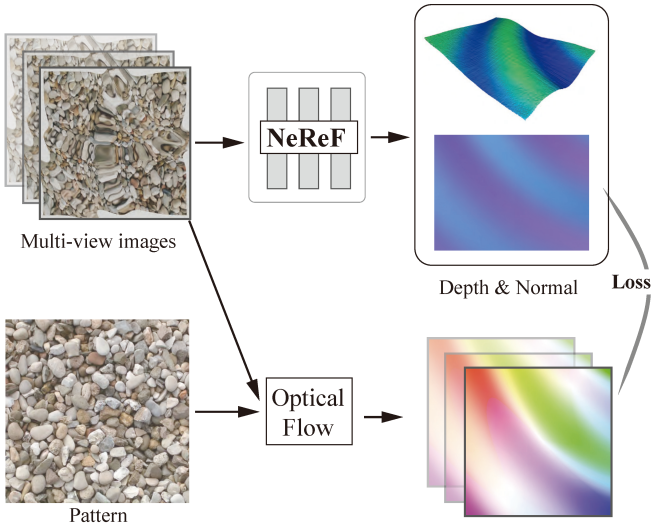


Fig. 2. The overview of our fluid surface reconstruction approach based on NeReF. Specifically, given multi-view inputs of fluid surface, we first calculate the optical flow w.r.t. the patter without water. Then we train NeReF with input images and it can produce depth and normal via volumetric rendering. Then the loss is computed between recovered depth normal and optical flow with refraction physics.

2 Related Work

Our work is closely related to researches in fluid surface reconstruction and neural scene representation.

Image based Fluid Surface Reconstruction Image-based reconstruction [13], [14], [15], [16], [17], [18], [19] and rendering (IBR) [20], [21], [22], [23], [24], [25], [26] is an active research area, which aims to synthesize novel views or recover the scene geometry from images captured at different viewpoints. Image-based fluid reconstruction is a sub-field of IBR, which commonly involves analyzing distortions in patterns placed underwater to reconstruct the water surface, as initially proposed by Murase [1]. The following shape-from-distortion methods can be categorized into single-view based methods or multi-view based methods. Approaches that adopt a single viewpoint setup usually assume additional surface constraints, such as planarity [27], [28], [29] and integrability [7], [9], to tackle the depth normal ambiguity. Notably, Tian and Narasimhan [30] develop a data-driven iterative algorithm to rectify the water distortion and recover water surface through spatial integration. Shan et al. [4] estimate the surface height map from refraction images with global optimization. Xiong and Heidrich [10] propose a novel differentiable framework to reconstruct the 3D shape of underwater environments from a single, stationary camera placed above the water in the wild environment. In contrast, multi-view based approaches rely on dedicately designed imaging/optic systems. Ye [7] exploit Bokode - a computational optical device that emulates a pinhole projector to capture ray-ray correspondences, which can be used to recover the surface normals directly. Morris et al. [2] extend the traditional multi-view triangulation to be appropriate for refractive scenes, and build up a stereo setup for water surface recovery. More recently, a learning-based single-image approach has recently been presented for recovering dynamic fluid surfaces [7]. Our capturing system setup is mainly similar to that of Qian et al. [3], which employs a 3x3 camera array and relies on the optical flow technique to recover both the underwater subject and fluid surface. However, like other approaches, it models the air-fluid interface as depth and normal maps. In contrast, We propose to implicitly encode the fluid surface into a fully connected neural network.

Another line of works aims to recover transparent objects, such as gas flow [31], which we recommend reading for extra information.

Neural Radiance Field The remarkable work of Neural Radiance Field is a milestone in novel view synthesis area. Before Mildenhall et at [32] raise the idea of NeRF, several methods are proposed to predict photo-realistic novel views of scenes base on dense sampling views. Theses methods can be divided into two classes. One class of methods use mesh-based representations of scenes with diffuse [33] or view-dependent [34], [35], [36] appearance, optimized by differentiable rasetizers [37], [38], [39], [40] or pathtracers [41], [42]. Another class of methods use volumetric representations to address this task. NeRF combines the implict representation with volumetric rendering to achieve compelling novel view synthesis with rich view-dependent effects. As the headstone of many following works, including ours, NeRF uses the weights of a multilayer perceptron(MLP) to represent a scene as a continuous volumetric field of particles that block and emit light. It takes single continuous 5D coordinates (spatial location (x, y, z) and view direction (θ, ϕ)) as input and outputs the volume density σ and view-dependent color c.

Based on the pipeline of NeRF, several extended works have been proposed. Ricardo et al [43] present an approach to enable the NeRF capable of modeling uncontrolled images from unstructured photo collections with learning a per-image latent embedding apperance variations and decomposing scenes into image-dependent components. In

Chen et al's [44] work, the MVSNeRF is a deep neural network that is able of utilizing three nearby input views via fast network inference to reconstruct radiance fields. Jonathan et al [45] combine the mip-map approach and NeRF together, simultaneously improve the original positional encoding into an integrated positional encoding, which represent the volume covered by each conical frustum, ending up with mip-NeRF which reduces aliasing and improves NeRF's ability to represent fine details. In Alex Yu et al [46]'s PlenOctreees for real-time rendering of Neural radiance fields, spherical harmonics are applied as the base of a representation of view-dependent colors. Besides, PlenOctree is applied to store density and SH coefficients modelling view-dependent appearance at each leaf. With these two measures, this method can render images at more than 150FPS, which is thousands times faster than the conventional method.

Neural Scene Representation Scene representation is a process that interprets the visual data into a feature representation. By providing the aimed pose and latent code [47] or alternatively transform views directly in the latent space [48], novel view images can be rendered. Generative Query Network(GQN) [49] provides framework within which machine learning scenes using their own sensors. However, this framework has the limitation of being oblivious to the 3D structure. Voxel grid representations and graph neural networks are the method that capture 3D structures. Vincent Sitzmann et al [50] propose a continuous 3D-structure-aware neural scene representation network, which trained from 2D images and their camera poses and encodes both geometry and appearance. Following their previous work, Sitzmann propose the light field network. This neural network represents the light field of a 3D scene implicitly, which can be used to extract depth maps from 360-degree light fields.

3 Neural Refractive Field

The core at our fluid surface reconstruction approach is as a **ne**ural **re**fraction **f**ield (NeReF), which retains the continuous scene representation ability of NeRF. Before proceeding, let's briefly review NeRF for easier explanation. NeRF represents a scene using a fully-connected deep network, whose input is a spatial location and viewing direction and output is the volume density and view-dependent emitted radiance. A novel image view is synthesized by sampling coordinates along camera rays and use volume rendering techniques to project the output colors and densities into an image. In this paper, we use the same scheme but adapt the NeRF for generating a normal value at each spatial location.

Specifically, we represent the continuous fluid surface as an implicit function $\mathcal{F}$, which is approximated by a Multi-Layer Perception (MLP). It takes a 3D location (x, y, z) as input, and outputs the surface normal $\mathbf{n}$ along with the volume density σ.

$$\mathcal{F} : (x, y, z) \rightarrow (\sigma, \mathbf{n}) \tag{1}$$

Note that although this modification is relatively small, the underlying hypothesis is very different. Our NeReF encodes the refractive surface normals which deflect the rays spatially, in contrast NeRF models rays' radiance which is in another dimension of spatial domain.

3.1 Depth and Normal from Volume Accumulation

From NeReF, depth and normal are accumulated along the camera ray. Specifically, given k sample points along a camera ray $\mathbf{r} = \mathbf{o} + \lambda\mathbf{d}$ where $\mathbf{o}$ is the origin of the ray and $\mathbf{d}$ is the ray direction, with each sample point determined by parameter $\lambda_i, i \in \{1, ..., k\}$. We query the sampling point's volume density σ_i and normal $\mathbf{n}_i$. Then, we calculate the surface normal $\mathbf{N}(r_k)$ and depth $D(r_k)$ of the last sampling point λ_k as:

$$\begin{aligned} \mathbf{N}(\mathbf{r}_k) &= \sum_{i=1}^{k} \tau_i [1 - \exp(-\sigma_{\delta i})] \mathbf{n_i} \\ D(\mathbf{r}_k) &= \sum_{i=1}^{k} \tau_i [1 - \exp(-\sigma_i \delta_i)] \lambda_i \end{aligned} \tag{2}$$

where $\tau_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right)$, and $\delta_i = \lambda_{i+1} - \lambda_i$ denotes the distance between two adjacent samples along the ray. Then, the coordinate of 3D point that $\mathbf{r}$ intersects the fluid surface is $\mathbf{p}_s = \mathbf{o} + D(\mathbf{r}_k)\mathbf{d}$

3.2 Refraction Execution

The implicit fluid representation should also obey the Snell's Law when a ray traverse the refractive surface. The refraction process follows Snell's law, i.e., $n_1 \sin\theta_1 = n_2 \sin\theta_2$, where n_1, n_2 and θ_1, θ_2 are the refraction indexes and incident and refracted ray angles respectively.

For each camera ray $\mathbf{r}$, we calculate the refracted ray $\mathbf{r}' = \mathbf{p}_s + \lambda\mathbf{d}'$ using the Snell's law in vector form, as:

$$\mathbf{d}' = \frac{s \cdot \mathbf{d} + (sa - b)\mathbf{N}(\mathbf{r})}{\|s \cdot \mathbf{d} + (sa - b)\mathbf{N}(\mathbf{r})\|_2} \tag{3}$$

where

$$\begin{cases} s = n_1/n_2, \\ a = -\mathbf{N}(\mathbf{r}) \cdot \mathbf{d_c}, \\ b = \sqrt{1 - s^2\,(1 - (\mathbf{N}(\mathbf{r}) \cdot \mathbf{d})^2)} \end{cases} \tag{4}$$

In our case, n_1 and n_2 are the refractive indexes of air and water, respectively.

4 Fluid Surface Reconstruction with NeReF

We represent the fluid surface implicitly with NeReF. One remaining problem is how to train the network. Previous section describes how to refract a ray using NeReF, in our implementation, we assume the camera ray refracts once when it hits the fluid surface, i.e., $\mathbf{N}(\mathbf{r}) = \mathbf{N}(\mathbf{r}_k)$ for $k \rightarrow \infty$, but remember this assumption is not required for our NeReF but only for the easy analysis of the fluid surface reconstruction problem.

We can use the consistency between the refracted rays to optimize the network, but need the ground truth refracted rays. Hence we follow the scheme of previous approaches and place a reference pattern under the water. We first capture images of the pattern without water and then pour the water in. Then we can use the positional differences of corresponding intersection points on pattern between the refracted rays and un-refracted rays and use it a loss for NeReF optimization. The process is visualized in Fig. 3.

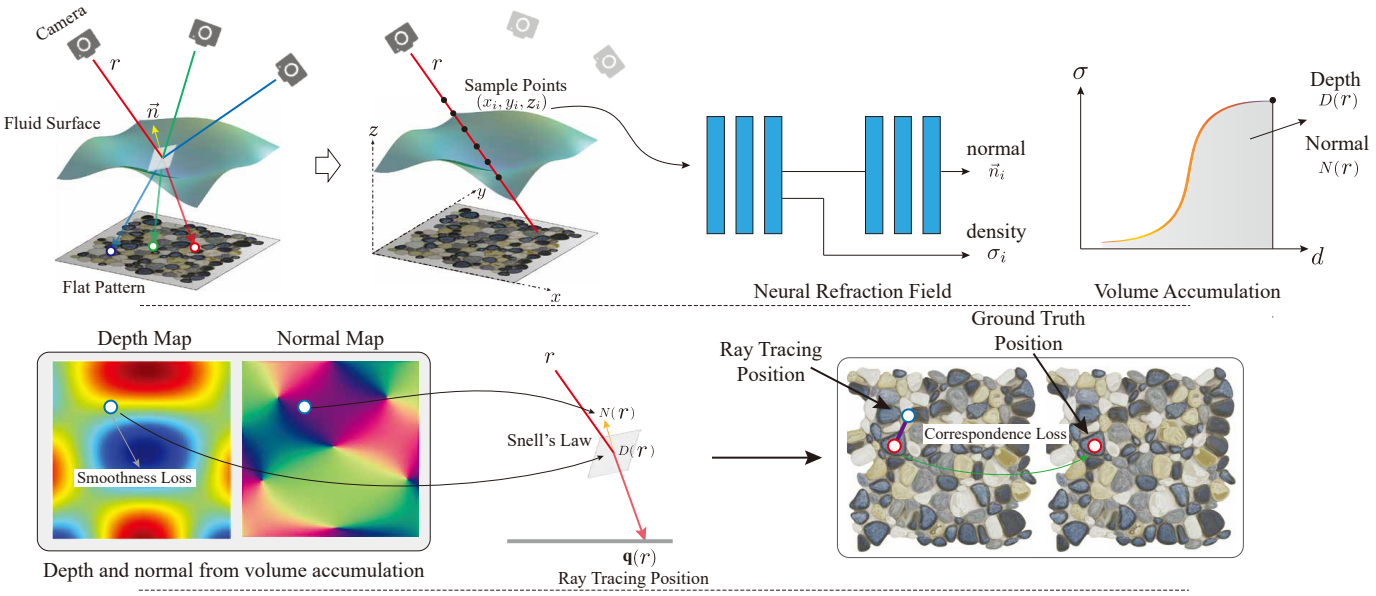


Fig. 3. The pipeline of our fluid surface recontruction method from NeReF. A sample point on input ray goes through a MLP network to predict the density and normal. The depth and normal can be explicitly rendered from NeReF and we use a correspondence and depth normal consistency loss to optimize NeReF.

4.1 Problem Formation

We set the $x-y$ plane of coordinate system to be aligned with the reference pattern, and z direction is up and perpendicular to the pattern plane. Then normal of the reference plane is $\mathbf{n}_\pi = [0, 0, 1]$. For a refracted ray $\mathbf{r}'(\lambda) = \mathbf{p}_s + \lambda \mathbf{d}'$, we can obtain the intersection point $\mathbf{q}'$ with the reference plane from ray-plane intersection:

$$\lambda_{\mathbf{q}'} = \frac{-\mathbf{p}_s \cdot \mathbf{n}_\pi}{\mathbf{d}' \cdot \mathbf{n}_\pi} \tag{5}$$

where $\cdot$ denotes the dot product. Similarly, we can obtain the intersection point $\mathbf{q}$ of the ray without refraction from $\lambda_{\mathbf{q}}$. Then our aim is to optimize the NeReF network from $\mathbf{q}$ and $\mathbf{q}'$.

However, there is still one remaining problem that we don't know how to correspond $\mathbf{q}$ and $\mathbf{q}'$. Notice $\mathbf{p}$ and $\mathbf{p}'$ are projection points on the image plane of rays $\mathbf{r}$ and $\mathbf{r}'$. So the ray $\mathbf{r}$ that before refraction can be directly obtained from camera parameters and pixel location $\mathbf{q}$. Then we use the optical flow [51] technique to obtain a dense warp field $\mathcal{W}$ from the distorted frame I' to reference frame I, presumably we obtain the shift of $\mathbf{q}'$ as:

$$I(\mathbf{q}_{gt}) = I'(\mathbf{q}' + d_q) \tag{6}$$

We set $\mathbf{q}_{gt}$ as our ground truth point for the refracted ray $\mathbf{r}'$ of $\mathbf{r}$. Then we refract $\mathbf{r}$ using NeReF network during training and produce a $\mathbf{q}_\mathcal{F}$. We define our correspondence loss as the L1 smoothed difference between $\mathbf{q}_{gt}$ and $\mathbf{q}_\mathcal{F}$ as:

$$\mathcal{L}_{corr} = \mathcal{S}^1(\|\mathbf{q}_\mathcal{F} - \mathbf{q}_{gt}\|_2) \tag{7}$$

where $\mathcal{S}^1$ denotes the L1 smooth operatoer.

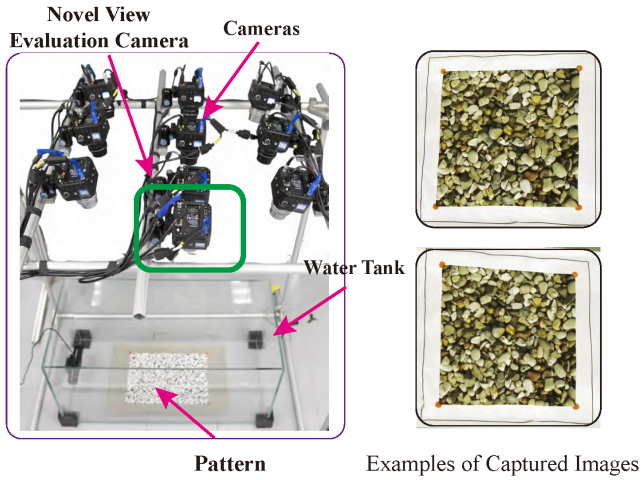


Fig. 4. The real fluid surface capture system which consists of 10 Z Cam E2, a water tank and a pattern under the water. We reserve one camera specifically for the evaluation purpose. On the right shows example real images captured.

4.2 Depth Smoothness

In practice, we notice depth tends to contain more noise compared to normal. While according to fluid dynamics, depth should be very smooth due to physic constraints. Hence we use a depth smoothness loss:

$$\mathcal{L}_{ds}(D) = \frac{1}{M} \sum_{\mathbf{r}} \mathcal{S}^1(|\frac{\partial D(\mathbf{r})}{\partial x}| + |\frac{\partial D(\mathbf{r})}{\partial y}|) \tag{8}$$

Finally, our total loss then is the weighted combination of previous losses as:

$$\mathcal{L}_{tol} = \mathcal{L}_{corr} + \lambda_{ds}\mathcal{L}_{ds} \tag{9}$$

Specifically, we set $\lambda_{ds} = 0.15$ in our implementation.

5 Implementation Details

In our implementation, the NeReF consists of 8 fully connected layers and each layer has 256 channels. The network takes location as input and generates density σ. Then we use another 3 layers to regress normal from σ. Hence the total size of our NeReF is about 4MB.

Training Details. We train our models using Adam optimizer with a learning rate 4E-4 which decays 5E-5 per 1000 iterations during training. Besides, we sample 2048 camera rays for each mini-batch and sample 96 and 192 points from near to far following the hierarchical sampling strategy. We optimize all our networks on a PC with a single Nvidia GeForce RTX3090 GPU. For each video sequence, we train the first frame for 10 epochs. which takes about 1 hours. The following frames are initialized using the previous one, requiring 15 minutes of training.

System Setup. To validate the proposition of our NeReF, we construct a fluid capture system consists of a water tank, a pattern underwater and a 10 camera array. The water tank is with size 12 inches for wave simulation. The industrial cameras we use is Z Cam E2 and we place all cameras on on top to record videos of the water. We calibrate the intrinsics and extrinsics of the cameras, and remove lens distortion using OpenCV [52]. We use 9 cameras for NeReF training and the remaining one for evaluation and testing.

6 Experiment

We first evaluate the fluid surface reconstruction approach using our NeReF, for synthetic fluid data generated by Blender [53]. We exploit the fluid physics simulation function of Blender and set up a scene that contains 25 cameras and water with size 2 x 1 x 4. We place a binary planar pattern beneath the water, and then use the wave modifier to simulate the shallow water equation, Grestner's equation, and Gaussian equation effects. We generate 4 sequences, of which each sequence contains 90 images. We place 25 pinhole cameras on top of the fluid to capture the pattern distortions under various wave functions.

Our approach relies on the accurate optical flow for corresponding rays w/o fluid refractions. We obtain a dense warp field using RAFT [51], and train our NeReF using 9 camera views as described above. We then render the depth, normal from NeReF using volumetric integration approach. We use the ratio between the L2 error of the rendered depth and normal and groundtruth as our metric. As shown in Fig. 5, the recovered normals on synthetic data is smooth and the result point clouds are very accurate. Errors at most parts of the normal and depth maps are indistinguishable, only normals at the peaks of ripple waves show relatively larger errors. This is reasonable as the normal changes dramatically in the ripple peak regions. Notice in Fig. 5, the scale of the error map is very small, i.e., 0.1, and normal error in row 3 is just relatively larger. The synthetic experiments demonstrate that our proposed NeReF works accurately for fluid surface reconstruction.

6.1 Evaluation metrics

Depth Accurancy: Root Mean Square Error (RMSE) for the recovered depth is

$$\mathrm{RMSE}(D, \hat{D}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(D_i - \hat{D}_i)^2}$$

where D represents the ground truth and $\hat{D}$ is the estimated depth map.

Normal Error: Average Angle Difference for the recovered normal is

$$\mathrm{AngDiff}(N, \hat{N}) = \frac{1}{n}\sum_{i=1}^{n}\arccos < N_i, \hat{N}_i >$$

where N represents the ground truth and the $\hat{N}$ is the estimated normal map.

6.2 Quantitative Evaluation.

To evaluate our method quantitatively, we compute the PSNR, LPIPS and SSIM metrics on real captured fluid data. Recall that we mounted 10 cameras on top of a water tank for capturing the fluid dynamics. However, only 9 of them are used for NeReF training and we use the remaining one for testing. Specifically, with the optimized NeReF, we synthesize a testing view image by setting the sampling camera to be in the same location as the testing camera. We compare the synthesized image with the captured testing image in terms of PSNR, LPIPS and SSIM metrics and the result is shown in Tab. 1. We also include the metrics reported by the original NeRF for reference. We have tried our best to find an alternative fluid surface reconstruction method to compare ours with. We can only identify that Ding's [8] has the same multi-camera setup with available code. Hence we use Ding's approach to recover the depth and normal, and then generate the view and testing camera using ray tracking and report the metrics in Tab. 1 too. As we can see the NeReF shows good metrics and it means that the synthesized data matches well with the testing camera.

	PSNR	**LPIPS**	**SSIM**
Ding's	25.110	0.062	0.878
NeRF	21.560	0.217	0.684
Ours	28.926	0.033	0.942

TABLE 1
Comparison of our NeReF with the original NeRF and Ding's [8] in term of recover accuracy. PSNR, LPIPS and SSIM values of our method and the original NeRF on the real fluid sequence.

We further visualize the reconstructed point-clouds and normal maps extracted from NeReF in Fig. 8 and Fig. 9. We render the depth maps from NeReF and then back-project the depth into point-clouds. Fig. 8 exhibits four recovered sequences of point-clouds which seems plausible. In Fig. 9, we compare our recovered point-clouds and normals with Ding's [8]. Ding's results contain lots of noises and shape edges, while our point-clouds and normals are smoother and consistent with fluid dynamics.

To evaluate the accuracy of recovered depth and normal, we compute the RMSE and Angle Difference metrics on the synthetic fluid sequence. We compare our methods, without depth smoothness, with Ding's [8]. The result is shown in Tab. 2. Our full model demonstrates the best results on both of these metrics.

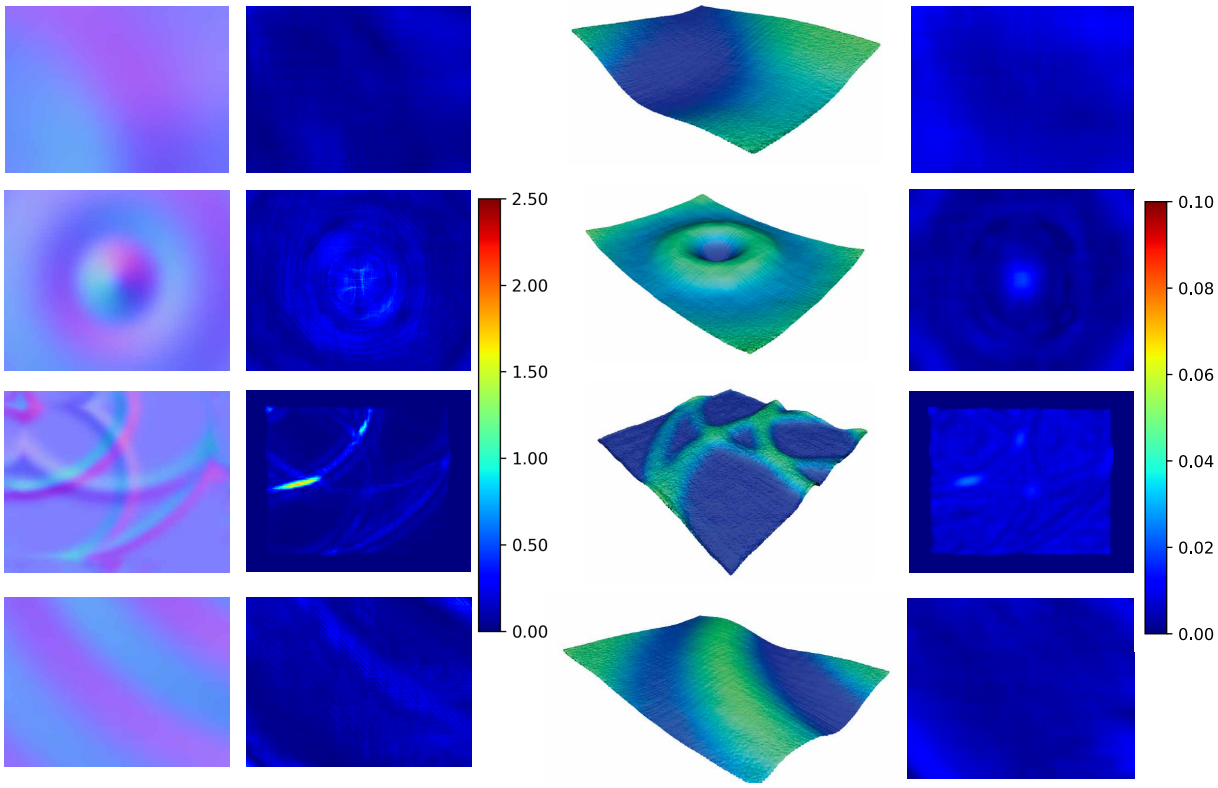


Fig. 5. Example fluid surfaces from synthetic data generated by Blender reconstructed by NeReF. From left to right are recovered normal map, normal error (represented as angles in degrees), depth visualized as point cloud, and depth error map (in meters). The results exhibit very small error and prove the effectiveness of our method.

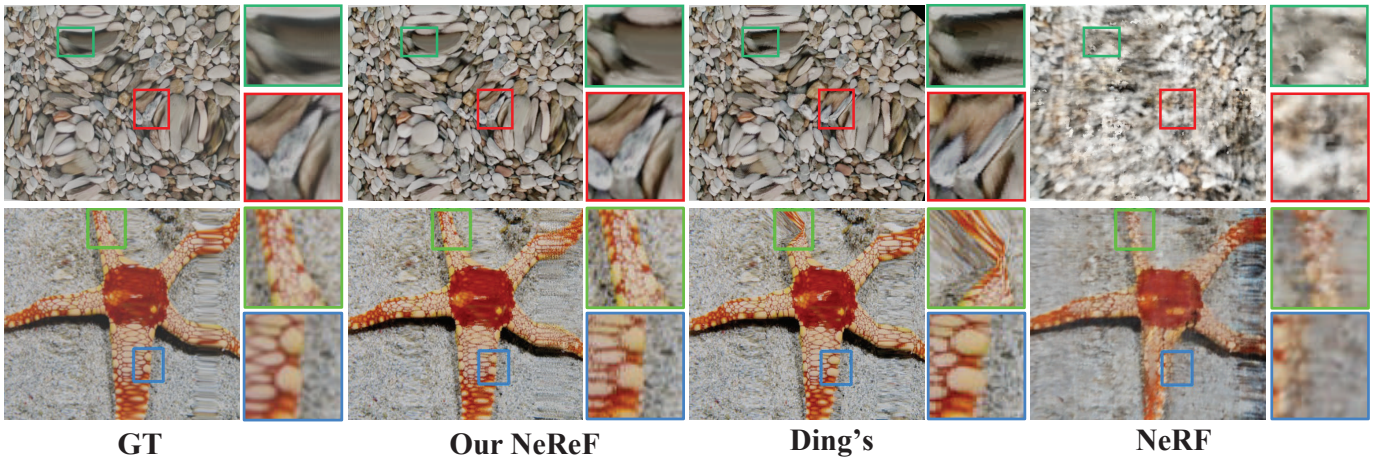


Fig. 6. The re-rendering results directly from NeReF (not through ray tracing using depth and normal) on synthetic data compared with other methods.

6.3 Ablation Study

Here we conduct ablation studies to analyze how several key factors would affect the performance of NeReF:

Tolerance of flow error Since optical flow provides the ray correspondences for our NeReF training, we first evaluate how errors in optical flow affect the final performance. From the synthetic data as described in the above section, we calculate the ground truth optical flow. Then we add random noise at different amplitude levels, train and test the NeReF model and then use the trained NeReF model for depth and normal recovery tasks.

Figure 7 shows the depth accuracy w.r.t. noise level. Here we also use the error ratio w.r.t. the groundtruth depth as the depth accuracy metric, and mean error for normal. As we can see in Fig. 7, the depth and normal errors increase while adding larger noise to the optical flow. We also observe that the error increases linearly w.r.t. the optical flow noises, which demonstrates that our NeReF is relatively robust.

Camera Number Evaluation The number of cameras is another important factor that may affect NeReF performance. We gradually decease the camera views for training from 9, 7, 5 to 3. After training, we synthesize images at the testing viewpoint with the trained model and compare with the ground truth depth and normal. We use the average

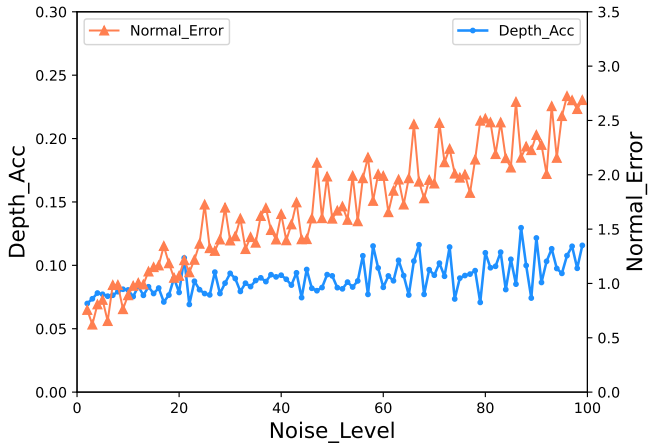


Fig. 7. We study how noises in optical flow affect the performance of NeReF. We gradually increase the noise level and train the NeReF for depth and normal recovery, and calculate the error with ground truth.

Methods	Recovered Depth RMSE (meters)	Recovered Normal Angle Diff. (deg.)
Ding's	0.07362	0.67489
Ours (w/o smooth.)	0.03991	0.45411
Ours (full model)	0.02252	0.28477

TABLE 2
Comparison of the accuracies of the recovered depth and normal between our NeReF and Ding's [8]: the RMSE and Average Angle Difference values on the synthetic fluid sequence.

depth and normal error of all testing views. The result is shown in Tab. 3, as the number of cameras decreases the reconstruction error becomes larger.

Depth Smoothing In Fig. 10, we conduct ablation studies on post depth smoothing process. With the recovered depth from NeReF, we conduct additional smoothing on depth maps can use the smoothed depth maps for rendering. The rendered testing view contains much less error compared with the result without depth smoothing.

6.4 Refraction and View Synthesis with NeReF

One main advantage of NeRF is its ability to synthesize photo-realistic novel views. This ability comes from the density-based representation, and novel views can be synthesized via volumetric integration of density. Similarly, we can integrate the volume represented as NeReF to obtain depth and normal. With the result depth and normal, we can directly apply Snell's law to find the refracted ray and hence find where the ray hits after refraction. As such, we synthesize novel views and simulate the fluid dynamics with a new background from the NeReF.

We first compare our NeReF with the original NeRF and ray tacing through depth and normal generated by [8]. The comparison is shown in Fig. 6. We can see original NeRF performs very poorly as the ray geometry are wrong. Moreover, results from ray tracing [8] contain artifacts as ray tracing can only be performed at a certain resolution.

num. of cameras	Recovered Depth RMSE (meters)	Recovered Normal Angle Diff. (deg.)
9	0.02252	0.28477
7	0.03193	0.34128
5	0.04540	0.43536
3	0.05683	0.84187

TABLE 3
Ablation study on number of cameras. RMSE of depth and Average Angle Difference of normal increases as the number of cameras descreases.

	PSNR	LPIPS	SSIM
No Depth Smooth.	27.931	0.036	0.929
Full Model	28.926	0.033	0.942

TABLE 4
Ablation study on depth smoothness regularization in term of recover accuracy. PSNR, LPIPS and SSIM values on the real fluid sequence.

While the synthesis result from our NeReF is the most close to ground truth.

Then we compare the novel view synthesis result on real fluid data. Fig. 11 exhibits that results of ray tracing from Ding's [8] depth and normal usually contain minor artifacts, the result from vanilla NeRF is unusable, while results of our approach are very close to the ground truth. The rendered binary pattern is much clearer and deforms in the same way as in the real image.

In Fig. 12, we show the recovered depth and normal for the frame captured in Fig. 11. And we re-render using trained NeReF at various viewpoints with a new pattern as background. Notice how distortions are presented differently in each image due to viewing point change.

7 Discussion

In this paper, we propose a neural scene representation for refractive fluid surfaces, called the Neural Refractive Field or NeReF. Specifically, we represent the fluid surface as a fully-connected deep network, which takes 3D coordinates as input and output the volume density and normal. We can render the refraction effect directly from the implicit representation. We are first to demonstrate that normal can be encoded into a neural field.

We only train and test the NeReF with fluid surface in a water tank. The surface is then rather flat and we assume one time of refraction. This assumption may not hold for water in flow or more complicate fluid surface. Hence, in the future, we plan to employ our method for complicated refractive shape recovery. Moreover, our adaption approach has removed the view dependent radiance from NeRF. Hence the render results from NeReF is a constant color from pattern no matter what direction the target view is. We plan to add the view dependent radiance to NeReF and make it be able to synthesize specular highlights.

Acknowledgments

This work was supported by NSFC (grant nos. 61976138 and 61977047), and STCSM (2015F0203-000-06).

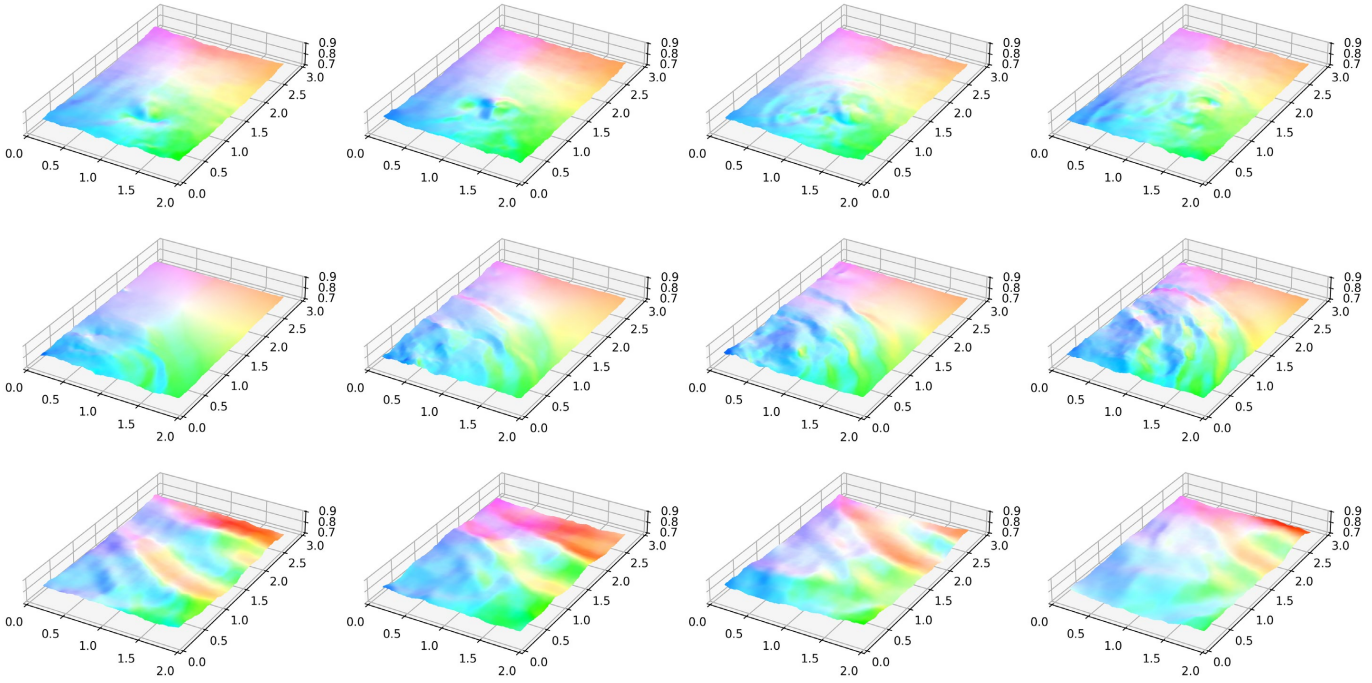


Fig. 8. Visualization of point-cloud sequences extracted from NeReFs trained on real captured fluid data. We obtain the point-clouds by back-projecting the rendered depth maps from NeRF. We show four frames of a sequence in each row. Please see the supplementary material for videos.

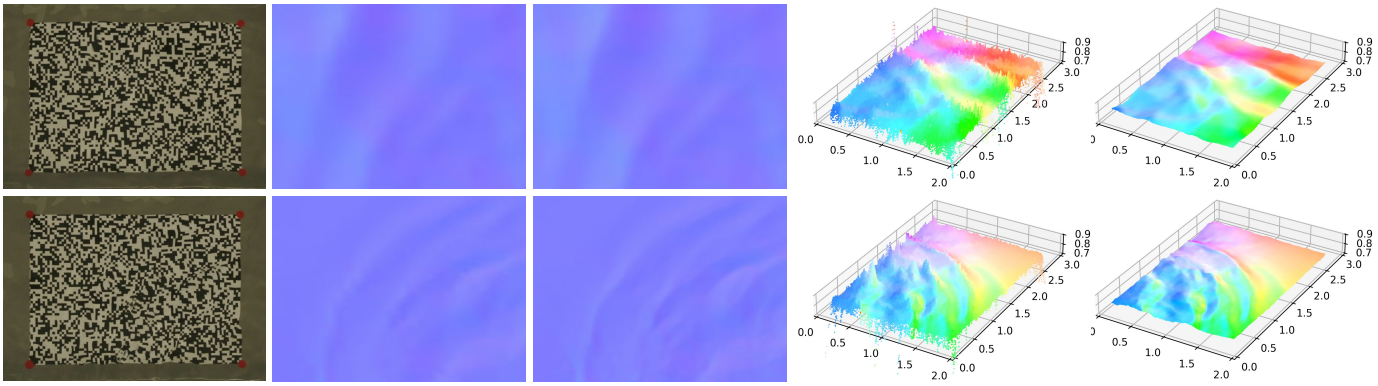


Fig. 9. We compare the recovered normal maps and point-clouds with Ding's [8]. Our point-clouds and normals are more smooth and consistent the observations.

REFERENCES

[1] H. Murase, "Surface shape reconstruction of an undulating transparent object," 1992.

[2] N. J. Morris and K. N. Kutulakos, "Dynamic refraction stereo," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 8, pp. 1518–1531, 2011.

[3] Y. Qian, M. Gong, and Y.-H. Yang, "Stereo-based 3d reconstruction of dynamic fluid surfaces by global optimization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1269–1278.

[4] Q. Shan, S. Agarwal, and B. Curless, "Refractive height fields from single and multiple images," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 286–293.

[5] K. Han, K.-Y. K. Wong, and M. Liu, "Dense reconstruction of transparent objects by altering incident light paths through refraction," *International Journal of Computer Vision*, vol. 126, no. 5, pp. 460–475, 2018.

[6] Y. Ji, J. Ye, and J. Yu, "Reconstructing gas flows using light-path approximation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2507–2514.

[7] J. Ye, Y. Ji, F. Li, and J. Yu, "Angular domain reconstruction of dynamic 3d fluid surfaces," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 310–317.

[8] Y. Ding, F. Li, Y. Ji, and J. Yu, "Dynamic fluid surface acquisition using a camera array," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 2478–2485.

[9] G. Wetzstein, R. Raskar, and W. Heidrich, "Hand-held schlieren photography with light field probes," in *2011 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2011, pp. 1–8.

[10] J. Xiong and W. Heidrich, "In-the-wild single camera 3d reconstruction through moving water surfaces," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 558–12 567.

[11] S. Thapa, N. Li, and J. Ye, "Dynamic fluid surface reconstruction

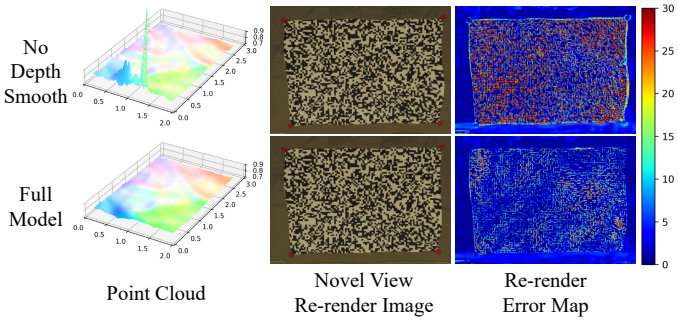


Fig. 10. We render the testing view from an optimized NeReF and compare it with the real captured testing image. The figures on the right illustrate error maps, where most errors are around the texture boundaries, and we can significantly reduce the error via depth smoothing.

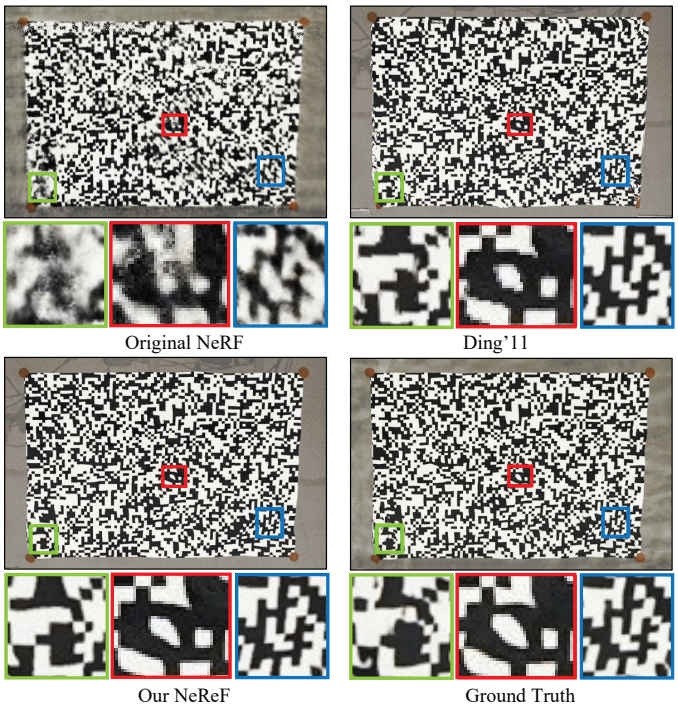


Fig. 11. The re-rendering results of real data directly from NeReF (no depth or normal explicitly involved) compared with the original NeRF, ray tracing from Ding's [8] depth and normal, the ground truth at the testing view.

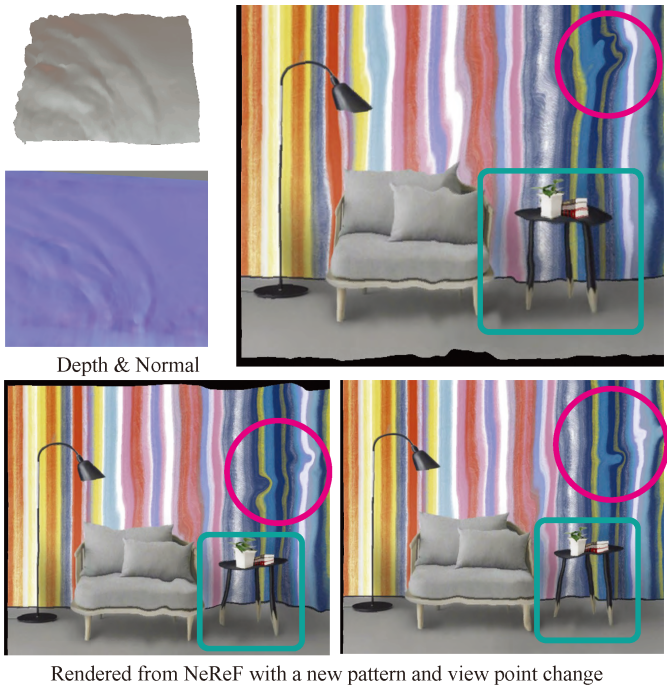


Fig. 12. Depth and normal from NeReF optimized on our real captured data. We also show 3 re-rendering image from NeReF with a changed pattern at different viewpoints, notice how distortions are presented at different views.

using deep neural network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 21–30.

[12] Y. Qian, Y. Zheng, M. Gong, and Y.-H. Yang, "Simultaneous 3d reconstruction for water surface and underwater scene," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 754–770.

[13] K. Kutulakos and S. Seitz, "A theory of shape by space carving," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 1, 1999, pp. 307–314 vol.1.

[14] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, "Building rome in a day," in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 72–79.

[15] S. Galliani, K. Lasinger, and K. Schindler, "Massively parallel multiview stereopsis by surface normal diffusion," June 2015.

[16] Y. Jiang, D. Ji, Z. Han, and M. Zwicker, "Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[17] L. Yariv, Y. Kasten, D. Moran, M. Galun, M. Atzmon, B. Ronen, and Y. Lipman, "Multiview neural surface reconstruction by disentangling geometry and appearance," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[18] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," in *Conference on Neural Information Processing Systems*, 2021.

[19] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," in *Conference on Neural Information Processing Systems*, 2021.

[20] P. Debevec, C. Bregler, M. Cohen, and L. Mcmillan, "Image-based modeling and rendering," 1998, p. 299.

[21] M. Levoy, "Light field rendering," in *Conference on Computer Graphics and Interactive Techniques*, 1996, pp. 31–42.

[22] A. Chen, M. Wu, Y. Zhang, N. Li, J. Lu, S. Gao, and J. Yu, "Deep surface light fields," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 1, no. 1, pp. 1–17, 2018.

[23] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: Image synthesis using neural textures," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–12, 2019.

[24] S.-Q. Li, Y. Gao, and Q.-H. Dai, "Image de-occlusion via event-enhanced multi-modal fusion hybrid network," *Machine Intelligence Research*, vol. 19, no. 4, pp. 307–318, 2022.

[25] S. Wizadwongsa, P. Phongthawee, J. Yenphraphai, and S. Suwajanakorn, "NeX: Real-time view synthesis with neural basis expansion," in *2021*.

[26] B. Attal, J.-B. Huang, M. Zollhöfer, J. Kopf, and C. Kim, "Learning neural light fields with ray-space embedding networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[27] Y. Asano, Y. Zheng, K. Nishino, and I. Sato, "Shape from water: Bispectral light absorption for depth recovery," in *European Conference on Computer Vision*. Springer, 2016, pp. 635–649.

[28] Y.-J. Chang and T. Chen, "Multi-view 3d reconstruction for scenes under the refractive plane with known vertical direction," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 351–358.

[29] R. Ferreira, J. P. Costeira, and J. A. Santos, "Stereo reconstruction of a submerged scene," in *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2005, pp. 102–109.

[30] Y. Tian and S. G. Narasimhan, "A globally optimal data-driven approach for image distortion estimation," in *2010 IEEE Computer*

Society Conference on Computer Vision and Pattern Recognition. IEEE, 2010, pp. 1277–1284.
[31] B. Atcheson, I. Ihrke, W. Heidrich, A. Tevs, D. Bradley, M. Magnor, and H.-P. Seidel, "Time-resolved 3d capture of non-stationary gas flows," *ACM transactions on graphics (TOG)*, vol. 27, no. 5, pp. 1–9, 2008.
[32] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *European conference on computer vision*. Springer, 2020, pp. 405–421.
[33] M. Waechter, N. Moehrle, and M. Goesele, "Let there be color! large-scale texturing of 3d reconstructions," in *European conference on computer vision*. Springer, 2014, pp. 836–850.
[34] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle, "Surface light fields for 3d photography," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 287–296.
[35] P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 11–20.
[36] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, "Unstructured lumigraph rendering," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001, pp. 425–432.
[37] W. Chen, H. Ling, J. Gao, E. Smith, J. Lehtinen, A. Jacobson, and S. Fidler, "Learning to predict 3d objects with an interpolation-based differentiable renderer," *Advances in Neural Information Processing Systems*, vol. 32, pp. 9609–9619, 2019.
[38] K. Genova, F. Cole, A. Maschinot, A. Sarna, D. Vlasic, and W. T. Freeman, "Unsupervised training for 3d morphable model regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8377–8386.
[39] S. Liu, T. Li, W. Chen, and H. Li, "Soft rasterizer: A differentiable renderer for image-based 3d reasoning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7708–7717.
[40] M. M. Loper and M. J. Black, "Opendr: An approximate differentiable renderer," in *European Conference on Computer Vision*. Springer, 2014, pp. 154–169.
[41] M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob, "Mitsuba 2: A retargetable forward and inverse renderer," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–17, 2019.
[42] T.-M. Li, M. Aittala, F. Durand, and J. Lehtinen, "Differentiable monte carlo ray tracing through edge sampling," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1–11, 2018.
[43] R. Martin-Brualla, N. Radwan, M. S. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "Nerf in the wild: Neural radiance fields for unconstrained photo collections," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7210–7219.
[44] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su, "Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo," *arXiv preprint arXiv:2103.15595*, 2021.
[45] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields supplemental material."
[46] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "Plenoctrees for real-time rendering of neural radiance fields," *arXiv preprint arXiv:2103.14024*, 2021.
[47] V. Sitzmann, J. Thies, F. Heide, M. Nießner, G. Wetzstein, and M. Zollhofer, "Deepvoxels: Learning persistent 3d feature embeddings," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2437–2446.
[48] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Interpretable transformations with encoder-decoder networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5726–5735.
[49] S. A. Eslami, D. J. Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor *et al.*, "Neural scene representation and rendering," *Science*, vol. 360, no. 6394, pp. 1204–1210, 2018.
[50] V. Sitzmann, M. Zollhöfer, and G. Wetzstein, "Scene representation networks: Continuous 3d-structure-aware neural scene representations," *arXiv preprint arXiv:1906.01618*, 2019.
[51] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *European conference on computer vision*. Springer, 2020, pp. 402–419.
[52] *The OpenCV Reference Manual*, 2nd ed., Itseez, April 2014.
[53] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org

Ziyu Wang received the BS degree from the School of Information Science and Technology, ShanghaiTech University, Shanghai, China, in 2020. He is currently working toward the Ph.D degree at ShanghaiTech University, Shanghai, China. His research interests include computer vision, deep learning, and computational photography.

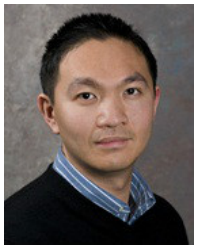
Wei Yang received the B.E. degree from the Huazhong University of Science and Technology, Wuhan, China, the MS degree from the Harbin Institute of Technology, Harbin, China, and the PhD degree from the University of Delaware, Newark, Delaware, in 2017. He is currently an Associate Professor with the Huazhong University of Science and Technology. His research focuses on imaging, graphics, computer vision, and artificial intelligence.

Junming Cao received the BS degree from Fudan University, Shanghai, China in 2018. He received the MS degree from Columbia University in the City of New York, in 2020. He is currently working toward the Ph.D degree at Shanghai Advenced Research Institute, Chinese Academy of Science, Shanghai, China. He is also with ShanghaiTech University, Shanghai, China. His research interests include computer vision, deep learning, and computational photography.

Qiang Hu received a B.E. degree from University of Electronic Science and Technology of China in 2013 and a Ph.D. degree in information and communication engineering from Shanghai Jiao Tong University in 2019. He is currently an Assistant Researcher with ShanghaiTech University. His research interests focus on video compression, neural rendering, and free-viewpoint video streaming.

Lan Xu received the B.E. degree from Zhejiang University in 2015 and the Ph.D. degree from the Department of Electronic and Computer Engineering (ECE), The Hong Kong University of Science and Technology (HKUST), in 2020. He is currently an Assistant Professor with ShanghaiTech University. His research interests include computer vision, computer graphics, and machine learning.

Junqing Yu received the BS degree from Wuhan University, in 1997, and the PhD degree from Wuhan University, in 2002. He is currently a Professor with the Huazhong University of Science and Technology. His research focuses on intelligent media computation.



Jingyi Yu received BS from Caltech in 2000 and PhD from MIT in 2005. He is currently the Vice Provost at the ShanghaiTech University. Before joining ShanghaiTech, he was a full professor in the Department of Computer and Information Sciences at University of Delaware. His research interests span a range of topics in computer vision and computer graphics, especially on computational photography and non-conventional optics and camera designs. He is a recipient of the NSF CAREER Award and the AFOSR YIP Award, and has served as an area chair of many international conferences including CVPR, ICCV, ECCV, IJCAI and NeurIPS. He is currently a program chair of CVPR 2021 and will be a program chair of ICCV 2025. He has been an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence, the IEEE Transactions on Image Processing, and the Elsevier Computer Vision and Image Understanding. He is a fellow of IEEE.