Dynamic Fluid Surface Reconstruction Using Deep Neural Network

Simron Thapa Nianyi Li Jinwei Ye
Louisiana State University, Baton Rouge, LA 70803, USA
{sthapa5, nli5, jinweiye}@lsu.edu

Abstract

Recovering the dynamic fluid surface is a long-standing challenging problem in computer vision. Most existing image-based methods require multiple views or a dedicated imaging system. Here we present a learning-based single-image approach for 3D fluid surface reconstruction. Specifically, we design a deep neural network that estimates the depth and normal maps of a fluid surface by analyzing the refractive distortion of a reference background pattern. Due to the dynamic nature of fluid surfaces, our network uses recurrent layers that carry temporal information from previous frames to achieve spatio-temporally consistent reconstruction given a video input. Due to the lack of fluid data, we synthesize a large fluid dataset using physics-based fluid modeling and rendering techniques for network training and validation. Through experiments on simulated and real captured fluid images, we demonstrate that our proposed deep neural network trained on our fluid dataset can recover dynamic 3D fluid surfaces with high accuracy.

1. Introduction

Dynamic fluid phenomena are common in our environment. Accurate 3D reconstruction of the fluid surface helps advance our understanding of the presence and dynamics of the fluid phenomena and thus benefits many scientific and engineering fields ranging from hydraulics and hydrodynamics [5, 20] to 3D animation and visualization [13]. However, it is difficult to tackle this problem with non-intrusive image-based methods as the captured images are often severely distorted by the refraction of light that happens at the fluid-air interface. This is because to extract invariant and reliable image features under distortion is highly challenging. Further, the dynamic nature of fluid flow makes this problem even more challenging as we need to recover a sequence of 3D surfaces that are consistent both spatially and temporally to represent the fluid motion.

Classical image-based methods for recovering the 3D fluid surface typically place a known pattern at the bottom of the fluid body and use a single or multiple cameras to capture the reference pattern through the fluid flow. Pattern distortions over time or among multiple viewpoints are analyzed for 3D fluid surface reconstruction. Since a single viewpoint is under-constrained, single image-based methods often impose additional surface assumptions (*e.g.*, planarity [3, 6, 14], integrability[42, 43], and known average height [29, 32] *etc.*). Otherwise, dedicated imaging systems or special optics (*e.g.*, Bokode [43] and light field probe [21, 42]) need to be used. Multi-view approaches rely on the photo-consistency among different viewpoints to perform 3D reconstruction. The seminal work of Morris and Kutulakos [29] extends the traditional two-view geometry to refractive medium with single deflection assumption. Camera arrays [10] are further adopted for more robust and accurate reconstruction. As being heavily dependent on the acquisition system, these classical methods usually use costly equipment that requires much effort to build and calibrate. Applications of these methods are thus limited.

In this paper, we present a learning-based approach for reconstructing the 3D fluid surface from a single refraction image. Following the setting similar to the classical methods, we take refraction image of a reference pattern through the fluid from a top-down view. We design a deep neural network that takes the refraction image as input and gener-

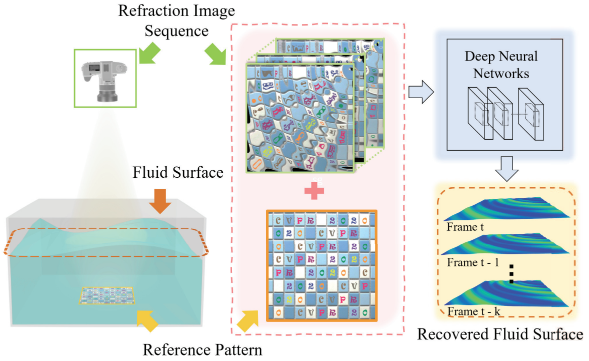

Figure 1. Our dynamic fluid surface reconstruction scheme. Given a sequence of refraction images captured through the dynamic fluid and the original reference pattern, we develop a deep neural network to recover spatio-temporally consistent 3D fluid surfaces.

alize distortion features for 3D fluid surface reconstruction. In recent years, deep learning techniques have achieved great success in solving computer vision problems, including depth estimation [11, 12, 23, 34, 27], 3D reconstruction [7, 19, 40], object detection and recognition [16, 22, 24], *etc*. Although most networks assume Lambertian scenes as limited by existing datasets, there is a rising trend to apply deep neural networks for interpreting more complex scenes with reflection, refraction, and scattering. Stets *et al*. [37] use a convolutional neural work to recover the shape of transparent refractive objects and show promising results. But both their network and dataset are not suitable for dynamic fluid surface reconstruction.

Specifically, our fluid surface reconstruction network (FSRN) consists two sub-nets: 1) an encoder-decoder based convolutional neural network (FSRN-CNN) for per-frame depth and normal estimation and 2) a recurrent neural network (FSRN-RNN) for enforcing the temporal consistency across multiple frames. Our FSRN-CNN compares the refracted pattern image with the original pattern to learn features from distortion for depth and normal estimation. We explicitly account for the physics of refraction in our loss function for training. Our FSRN-RNN uses the convolutional long-short term memory (conLSTM) layers to learn temporal dependencies from previous frames, and refines the depth and normal estimation for the current frame to enforce spatio-temporal consistency. We train the two subnets separately to reduce the number of network parameters. Both are trained with per-pixel depth and normal losses as well as a depth-normal consistency loss. Since no existing dataset can serve our purpose of fluid surface reconstruction, we synthesize a large fluid image dataset with over 40,000 fluid surfaces for network training and validation. We use a set of fluid equations [8, 36, 38] derived from the Navier-Stokes for realistic fluid surface modeling. We implement a physics-based renderer that considers complex light transport to simulate images through refraction. Our dataset also includes the ground truth depth and normal maps of the fluid surfaces. We perform experiments on our synthetic dataset as well as real captured fluid images. Both qualitative and quantitative results show that our approach is highly accurate in recovering dynamic fluid surfaces.

2. Related Work

In this section, we briefly review classical image-based methods for fluid surface reconstruction and the deep learning techniques that are relevant to our network design.

Classical image-based methods usually measure the refractive distortions of a known background pattern to recover fluid surfaces. We refer the readers to [18] for a comprehensive survey on refractive and reflective object reconstruction. Notably, Murase's pioneering work [30] analyzes the optical flow between the distorted image and the original one to recover per-pixel normals for water surface. Tian and Narasimhan [39] develop a data-driven iterative algorithm to rectify the water distortion and recover water surface through spatial integration. Shan *et al*. [33] estimate the surface height map from refraction images using global optimization. As surface reconstruction from a single viewpoint suffers from the intractable depth-normal ambiguity [18, 29], most single image-based methods assume additional surface constraints such as planarity [3, 6, 14] and integrability [42, 43]. Morris and Kutulakos [29] first extend the classical multi-view geometry to refractive medium and recover the fluid surface using a stereo setup. Ding *et al*. [10] further adopt a 3×3 camera array for more robust feature tracking under distortion. Qian *et al*. [32] develop a global optimization framework to improve the accuracy of refractive stereo. Another class of computational imaging approaches directly acquire ray-ray correspondences using special optics [17, 21, 43] and then triangulate the light rays for surface reconstruction. Being heavily dependent on the acquisition system, these classical methods usually use costly equipment that requires much effort to build and calibrate. In contrast, our approach allows for more flexible imaging setup and uses a learning-based algorithm for fluid surface reconstruction.

Deep learning techniques have achieved unprecedented success in numerous computer vision tasks including depth/normal estimation [11, 12, 23, 34, 27], 3D reconstruction[7, 19, 40] and object detection and recognition [16, 22, 24]. The encoder-decoder convolutional network architecture has proven effective in feature generalization for various applications. Most relevant networks are the ones for monocular depth/normal estimation. Eigen *et al*. [12] and Liu *et al*. [26] develop end-to-end trained convolutional networks for depth estimation from a single image. Wang *et al*. [41] and Bansal *et al*. [4] use fully connected convolutional networks with semantic labeling for single-image surface normal estimation. Qi *et al*. [31] present a network for joint depth and normal estimation that incorporates geometric constraints between depth and normal. However, all these networks assume Lambertian scenes because they are trained on datasets that are mostly composed of diffuse objects (*e.g*., NYU Depth [9] and KITTI [15]). They are, therefore, not applicable to recover fluid surfaces with reflective and refractive reflectance. Most recently, Li *et al*. [25] present a network to un-distort the refractive image of an underwater scene. Stets *et al*. [37] use convolutional network to recover the shape of transparent refractive objects. But these networks are not suitable for fluid surface reconstruction due to limitations of their datasets. In this work, we create a large physics-based fluid surface dataset with ground truth depth and normal. In addition, our network use recurrent layers [28] to capture the temporal dynamics of fluid flows.

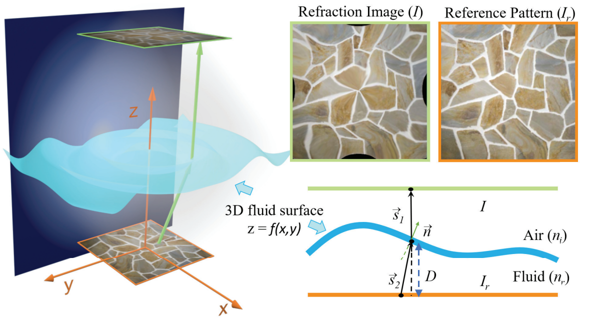

Figure 2. The setting of our fluid surface reconstruction problem. Given a refraction image (I) viewed from the top through the fluid flow and the original background pattern (I_r), we aim at recovering the fluid surface in form of depth and normal maps. Our network explicitly accounts for the physics of refraction in the training loss function.

3. Fluid Surface Reconstruction Network

In this section, we present our *fluid surface reconstruction network (FSRN)*. We first introduce the setting of our fluid surface reconstruction problem, and then describe our network structure and the generation of our physics-based fluid dataset.

3.1. Problem Definition

We represent the dynamic 3D fluid surfaces as a temporal sequence of the surface depths $\{z^t | t = 1, 2, ...\}$, where t is the time instant and $z^t = f^t(x, y)$ is the height field of the fluid surface at t. As is shown in Fig. 2, given a reference pattern I_r placed underneath the fluid surface at the $z = 0$, we can map I_r to a refraction image I^t as being distorted by the refraction that occurs at the fluid surface z^t:

$$I^t = \Phi(I_r, z^t). \tag{1}$$

where Φ is the mapping function that follows the physics of refraction (*i.e.*, the Snell's law).

Given a sequence of the refraction images $\{I^t | t = 1, 2, ...\}$ and the reference pattern I_r, we aim to estimate the dynamic fluid surfaces $\{z^t | t = 1, 2, ...\}$. Practically, I^t can be captured by an orthographic camera that looks at the fluid surface from the top and I_r is assumed known in advance. In our network, we estimate both the depth map and the normal map of a fluid surface as they can be independently inferred from the refractive distortions. Since they are also geometrically correlated, the depth and normal estimations can be further refined with a consistency loss. Finally, we can generate 3D fluid surface meshes from our estimated depths and normals through Delaunay triangulation.

3.2. Network Architecture

Our *fluid surface reconstruction network (FSRN)* consists of two sub-nets: 1) an encoder-decoder based convolutional neural network (FSRN-CNN) for per-frame depth and normal estimation and 2) a recurrent neural network (FSRN-RNN) for enforcing the temporal consistency across multiple frames. Fig. 3 shows the workflow of our FSRN and Fig. 4 shows its architecture.

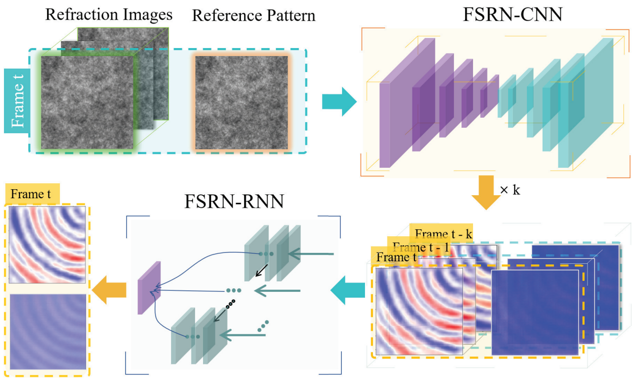

Figure 3. The workflow of our FSRN. The FSRN-CNN estimates depth and normal maps given a refraction image and the reference pattern. Its output is then structured into a temporal sequence and fed into the FSRN-RNN for refinement by enforcing the temporal consistency.

FSRN-CNN. Our CNN subnet takes in the refraction image I^t and the reference pattern I_r to estimate the the depth map D^t and normal map N^t of the fluid surface at time t (superscript t indicates the time instance). It uses the encoder-decoder structure to generalize features from refractive distortion. The encoder is consisted of stacked convolutional layers with max-pooling layers. The decoder is made up of transpose convolutional layers with skip connections (see Fig. 4). Specifically, our decoder has two branches: one predicts normalized depth and normal maps (D^t and N^t), and the other predicts the absolute ranges of depth and normal maps (R^t_D and R^t_N). In order to generalize scale-invariant features, we normalize our depth and normal maps to the range of $\left[0, 1\right]$. The absolute ranges are therefore critical to restore the actual scale of the fluid surface. To better exploit the geometric consistency between depth and normal, we use a common set of decoding layers for both depth and normal estimation. This subnet is end-to-end trained with loss functions described in Sec. 3.3.

FSRN-RNN. Our RNN subnet refines the depth and normal estimation by enforcing the temporal consistency. We concatenate multiple scaled depth and normal maps estimated by the FSRN-CNN as temporal sequences: $\{D^t | t = t, t-1, t-2, ...\}$ and $\{N^t | t = t, t-1, t-2, ...\}$. The temporal sequences of depth and normal maps are then used as input to feed into the FSRN-RNN. The output is refined depth and normal maps at the current time t. We use con-

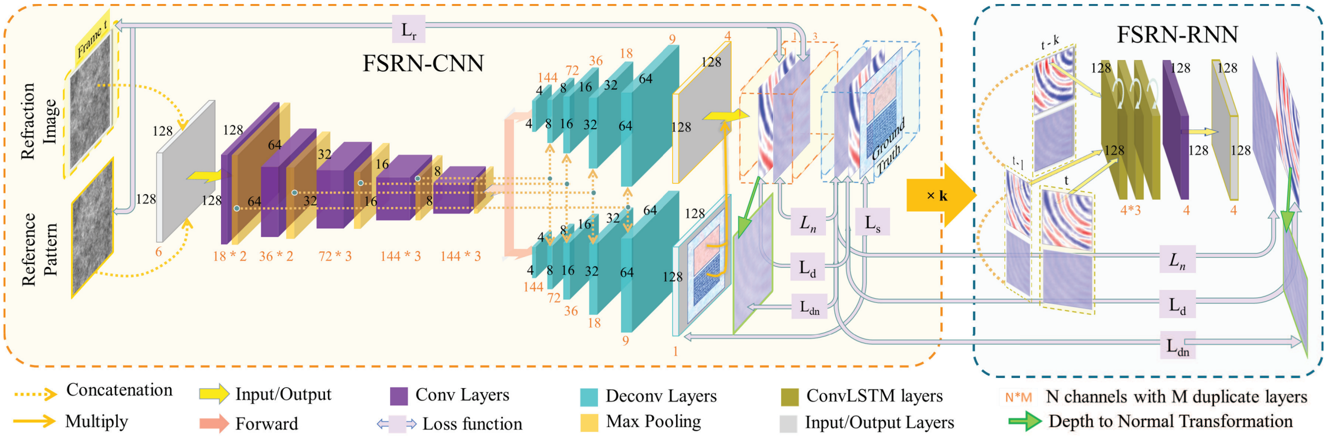

Figure 4. The overall architecture of our FSRN. Please refer to the supplementary material for more detailed parameters of our network.

volutional long-short term memory (conLSTM) layers [35] to construct our recurrent network. The conLSTM layers transmit hidden states from previous time frames to learn the temporal dependencies. This subnet therefore enforces temporal consistency in our reconstruction as well as enhances the estimation accuracy. The ablation study and real experiment results in Sec. 4 confirm the effectiveness of using the recurrent layers. This subnet is separately trained from the FSRN-CNN to reduce the number of network parameters. The loss functions are described in Sec. 3.3.

3.3. Loss Functions

Depth Loss. We use a per-pixel depth loss to compare our predicted depth map (D) with the ground truth one ($\hat{D}$). Similar to [11], we consider the L2-norm difference and scale-invariant difference (the first and second term in Eq. 2). The scale-invariant difference term panelize differences of opposite directions. It therefore preserves the shape of the surface regardless of the scale. In addition, we also consider a gradient term (the third term in Eq. 2) that takes the four-directional differences to favor smoother prediction. Let $d(p) = D(p) - \hat{D}(p)$ be the per-pixel depth difference (where $p \in [1, M]$ is the pixel index with M as the total number of pixels), our depth loss L_d is defined as

$$L_d(D, \hat{D}) = \frac{1}{M}\sum_p d(p)^2 - \frac{1}{2M^2}(\sum_p d(p))^2 + \frac{1}{M}\sum_p\sum_i \delta_i(p)^2 \quad (2)$$

where i indicate the indices of four neighboring pixels of p and $\delta_i(p) = d(i) - d(p)$ represents the four-directional difference of $d(p)$.

Normal Loss. As we predict our depth and normal maps in the same decoder branch, the x, y, and z components of the normal map are estimated in three separate passes. Our normal loss function is similar to the depth loss except that the computation is extended to three channels. We also exclude the third smooth term because the normals tend to have more drastic changes than depth. Given the predicted normal map N and the ground truth one $\hat{N}$, our normal loss L_n is defined as

$$L_n(N, \hat{N}) = \frac{1}{M}\sum_p n(p)^2 - \frac{1}{2M^2}(\sum_p n(p))^2 \quad (3)$$

where $n(p) = N(p) - \hat{N}(p)$ is the per-pixel difference.

Depth-Normal Loss. Since depth and normal are geometrically correlated, we use a depth-normal loss to enforce consistency between our depth and normal estimations. Specifically, given the predicted depth map D, we convert it to its corresponding normal map (N_d) by taking the partial derivatives: $N_d(p) = [\partial D(p)/\partial x, \partial D(p)/\partial y, -1]^\top$. We then normalize the normal vectors to unit lengths and convert the ranges of their x, y, and z components to $\big[0, 1\big]$. We then use the normal loss (Eq. 4) to compare the depth-converted normal map (N_d) with the ground truth normal map $\hat{N}$. Our depth-normal loss L_{dn} is then defined as

$$L_{dn}(N_d, \hat{N}) = \frac{1}{M}\sum_p n'(p)^2 - \frac{1}{2M^2}(\sum_p n'(p))^2 \quad (4)$$

where $n'(p) = N_d(p) - \hat{N}(p)$ is the per-pixel difference.

Refraction Loss. We use a refraction loss to directly account for the physics of refraction that occurs at the fluid-air interface. We trace a refraction image using the predicted depth and normal maps and the original reference pattern. We then compare the traced image with the input refraction image to minimize their difference. Specifically, we assume all incident rays $\vec{s}_1$ to the fluid surface are $[0,0,1]^\top$ as we assume an orthographic camera with top-down view. Given a predicted fluid surface normal $\vec{n}$, we can compute the refracted ray $\vec{s}_2$ by

$$\vec{s}_2 = \frac{n_r}{n_i}\left[\vec{n} \times (-\vec{n} \times \vec{s}_1)\right] - \vec{n}\sqrt{1 - (\frac{n_r}{n_i})^2(\vec{n} \times \vec{s}_1)^2} \quad (5)$$

where n_i and n_r are the refractive indices of air and water.

We then use the predicted depth values to propagate $\vec{s}$ and intersect with the reference pattern. The colors of the intersection points are returned to form our predicted refraction image (I). We then use the L2-norm difference to compare I with the ground truth refraction $\hat{I}$, which is the input to our network. Our refraction loss L_r is defined as

$$L_r(I, \hat{I}) = \frac{1}{M}\sum_p (\hat{I}(p) - I(p))^2 \quad (6)$$

Scale Loss. As our CNN subnet also predicts the absolute ranges of depth and normal maps in order to restore them to the actual scale, we simply use the L2-norm difference to compare our predicted ranges (R_D and R_N)with the ground truth ones ($\hat{R}_D$ and $\hat{R}_N$). Our ground truth ranges are obtained by taking the minimum and maximum values of the depth and normal maps [1] (*e.g.*, $\hat{R}_D = [\min(D), \max(D)]$). Our scale loss L_s is defined as

$$L_s(R_{D,N}, \hat{R}_{D,N}) = \frac{1}{M}\sum_p (R_{D,N}(p) - \hat{R}_{D,N}(p))^2 \quad (7)$$

Total Losses. Our two sub-nets are trained separately to reduce the number of network parameters. The total losses for FSRN-CNN (L_{CNN}) and FSRN-RNN (L_{RNN}) are combinations of the above described losses

$$L_{\text{CNN}} = \alpha_1 L_d + \alpha_2 L_n + \alpha_3 L_{dn} + \alpha_4 L_r + \alpha_5 L_s \quad (8)$$

$$L_{\text{RNN}} = \beta_1 L_d + \beta_2 L_n + \beta_3 L_{dn} \quad (9)$$

$\alpha_{1,\ldots,5}$ and $\beta_{1,2,3}$ are weighted factors and they are separately tuned for each subnet. Notice that we only use the refraction loss in FSRN-CNN as this computation is expensive and it's more efficient to apply it on a single frame rather than a temporal sequence. We also exclude the scale loss in FSRN-RNN because the inputs to this subnet have already been scaled to their actual ranges.

3.4. Physics-based Fluid Dataset

It is challenging to acquire fluid dataset with ground truth surface depths and normals using physical devices. We resort to physics-based modeling and rendering to synthesize a large fluid dataset for our network training. We use fluid equations derived from the Navier-Stokes to model realistic fluid surfaces and implement a physics-based renderer to simulate refraction images.

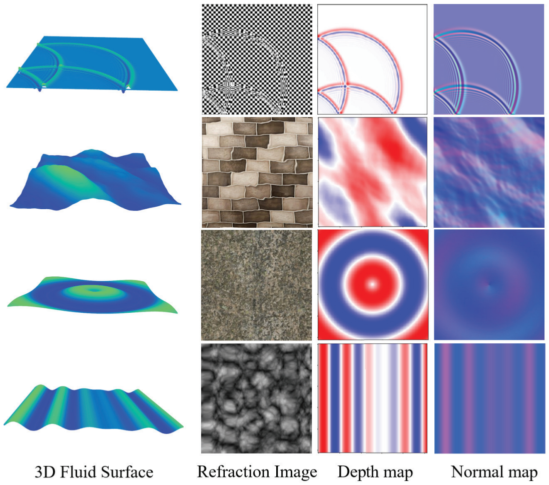

Figure 5. Sample images from our fluid dataset. From top to bottom, we show waves simulated by the shallow water equation, Grestner's equation, Gaussian equation, and sinusoidal equation. The patterns used are checkboard, tiles, concrete, and perlin noise.

Specifically, we use an Eularian mesh-based fluid simulation to model fluid surfaces. We use a variety of fluid equations derived from the Navier-Stoke to account for the versatility of natural fluid flows. The fluid equations we use include The shallow water equations [8], Grestner's equations [38], Gaussian equations, and sinusoidal equations. We choose these wave equations as they model different behaviors of fluid waves. The shallow water equations are a set of partial differential equations derived from the Navier-Stokes. They describe in-compressible property of fluid where the mass and linear momentum is conserved. The Grestner's equations are widely used in computer graphics to simulate ocean waves. We use them to model fluid with relatively large volumes. The Gaussian equations are used for creating water ripples with damping effects. The sinusoidal equations are used to model linearly propagating waves. More details of these waves equations can be found in the supplementary material. We use weighted linear combination of these equations to simulate the 3D fluid surfaces that are used in our dataset.

To render refraction images, we implement a ray tracer that considers the refraction of light. We setup our scene following the configuration shown in Fig. 2, where the camera, 3D fluid surface and the reference pattern are center-aligned. We trace rays from an orthographic camera and use Eq. 5 to compute the refracted rays (where we assume the indices of refraction for air and fluid are 1 and 1.33). The refracted rays are then traced to the reference pattern to form the refraction image.

Our dataset contains over 45,000 refraction images (75 fluid sequences) with the ground truth depth and normal maps. We also use a variety of reference patterns to en-

[1] For the normal map, we treat the three channels separately but in the same manner.

rich our dataset, which include noise patterns (*e.g.*, Perlin, Simplex, and Worley), checkerboards with different sizes, and miscellaneous textures (*e.g.*, bricks, tiles *etc.*). Sample images from our dataset are shown in Fig. 5.

4. Experiments

In this section, we evaluate our approach through both synthetic and real expriments.

4.1. Network Training

We implement our FSRN in TensorFLow [1] with around 1.7 million trainable parameters. All computations are performed in a computer with Xeon E5-2620 CPU and two NVIDIA GTX 1080 Ti GPUs. We segment our fluid dataset into 40,000 training images, 5,000 validation images and 1000 testing images. We set the parameters $\alpha_{1,\ldots,5} = 0.2$, $\beta_{1,2} = 0.4$, and $\beta_3 = 0.2$ for our total loss functions. It takes around 6 hours to train our network.

Our FSRN is trained in two steps. First, we train the FSRN-CNN with our fluid dataset. We process the training data by normalizing the input (*i.e.*, refraction image, depth and normal maps) to the range $[0, 1]$ slice-by-slice and save their true scale ranges. We use the Adam optimizer to train the network. We use batch size 32 for both training and validation. We initialize the learning rate as 10^{-3} and decrease it by half after 15 epochs. We train the network for 35 epochs till convergence. Second, we train the FSRN-RNN with a temporal sequence of re-scaled predictions from FSRN-CNN as input. Here we consider three consecutive frames. We use the Adam optimizer to train this network with a fixed learning rate of 10^{-3}. The batch size is 32 for both training and validation. We train the network for 15 epoch till converge.

4.2. Experiments on Synthetic Data

We first evaluate our approach on our synthetic fluid dataset. Our validation set contains 5,000 refraction images (20 unique dynamic fluid videos) that doesn't overlap with the training set. These data is rendered with various types of reference patterns. Our fluid surface reconstruction results are shown in Fig. 6. More dynamic fluid video results can be found in our supplementary material. We can see that our recovered fluid surfaces are highly accurate and well preserve the wave structure of the ground truths.

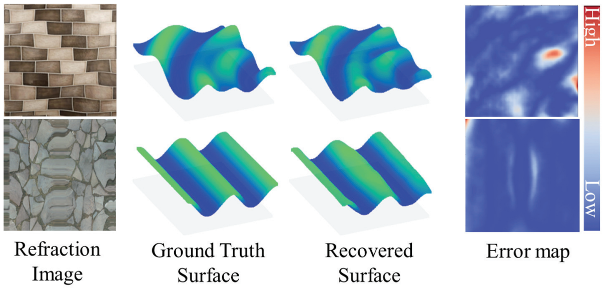

Figure 6. Fluid surface reconstruction on synthetic data.

Methods	Error metric		Accuracy metric		
	RMSE	Abs Rel	ρ<1.25	ρ<1.25^2	ρ<1.25^3
DenseDepth [2]	0.851	0.408	0.016	0.033	0.051
RefractNet [37]	0.303	0.274	0.226	0.422	0.584
FSRN-S	0.262	0.247	0.338	0.572	0.710
FSRN-CNN	0.128	0.105	0.557	0.803	0.896
FSRN (Ours)	**0.126**	**0.098**	**0.562**	**0.812**	**0.901**

Table 1. Quantitative comparison with existing methods on depth estimation. We highlight the best performance in **bold**.

We also perform quantitative evaluation in comparison with existing methods. As there are not many networks designed for fluid surface reconstruction, we choose two networks to compare with: 1) the RefractNet by Stets *et al.* [37], which is a CNN designed to reconstruct transparent refractive objects and 2) the DenseDepth by Alhashim and Wonka [2], which is a latest state-of-the-arts network for single-image depth estimation but designed for Lambertian scenes. As the DenseDepth doesn't perform well in our task, we didn't pick other Lambertian scene-based depth estimation network for comparison. We use five error metrics following [12] to evaluate our prediction: the root mean square error (RMSE), the absolute relative error (Abs Rel), and three threshold accuracies ($\rho < 1.25, 1.25^2, 1.25^3$). Formulas for computing these error metrics can be found in our supplementary material. As both the RefractNet and DenseDepth take a single image as input, for fair comparison, we also implement a single-input variation of our network (FSRN-S) that only take the refraction image (without the reference pattern) by not considering the refraction loss. We also compare with the depth prediction directly obtained from FSRN-CNN (without using the FSRN-RNN). All networks are trained on our fluid dataset. The quantitative comparison results are shown in Table 1. We can see that our FSRN out-performs the existing methods in all error metrics. We also show the visual comparison of predicted depth map in Fig. 7. We can see that the Lambertian scene-based method (DenseDepth) is unable to produce meaningful prediction. Although the RefractNet can recover some ripple waves, their overall estimation is highly noisy and inaccurate. In contrast, our FSRN can estimate highly accurate depth for fluid surface. And even without using the reference pattern and recurrent layers, our prediction still out-performs the existing methods.

Ablation Study. We perform ablation study to demonstrate the effectiveness of loss functions. In particular, we create three variations of our network: 1) FSRN-CNN$_1$ that only uses the basic depth and normal losses; 2)

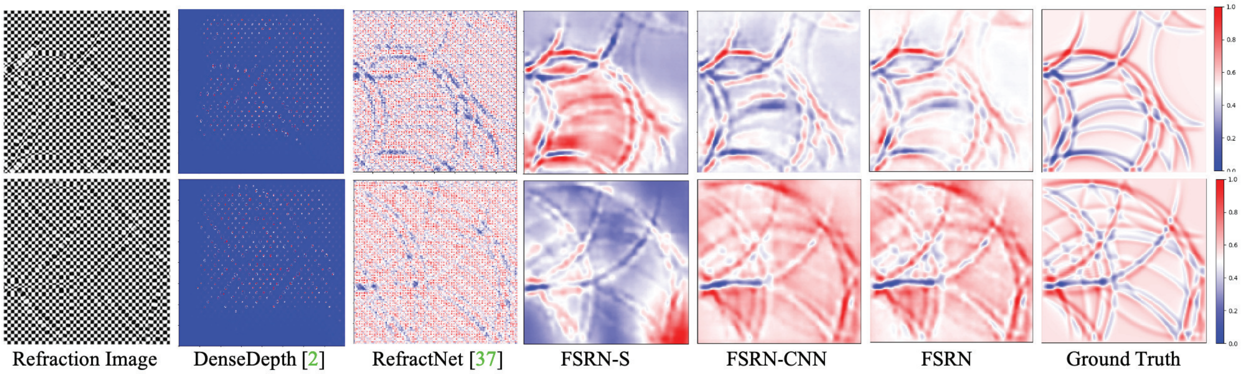

Figure 7. Visual comparison with existing methods on depth estimation. All depth maps are normalized to $[0, 1]$ for fair comparison.

Methods	Error metric		Accuracy metric		
Depth Estimation					
	RMSE	Abs Rel	$\rho<1.25$	$\rho<1.25^2$	$\rho<1.25^3$
FSRN-CNN$_1$	0.198	0.184	0.398	0.684	0.833
FSRN-CNN$_2$	0.183	0.175	0.469	0.707	0.848
FSRN-CNN$_3$	0.137	0.112	0.551	0.790	0.881
FSRN-CNN	0.128	0.156	0.557	0.803	0.896
FSRN (Ours)	**0.126**	**0.098**	**0.562**	**0.812**	**0.901**
Normal Estimation					
Methods	Error metric		Accuracy metric		
	RMSE	Abs Rel	$\rho<1.25$	$\rho<1.25^2$	$\rho<1.25^3$
FSRN-CNN$_1$	0.118	0.095	0.577	0.707	0.794
FSRN-CNN$_2$	0.112	0.094	0.578	0.715	0.799
FSRN-CNN$_3$	0.110	0.088	0.580	0.721	0.813
FSRN-CNN	0.098	0.108	0.051	0.759	0.864
FSRN (Ours)	**0.079**	**0.051**	**0.693**	**0.829**	**0.912**

Table 2. Depth and surface normal estimation measurements for ablation study. We highlight the best performance in **bold**.

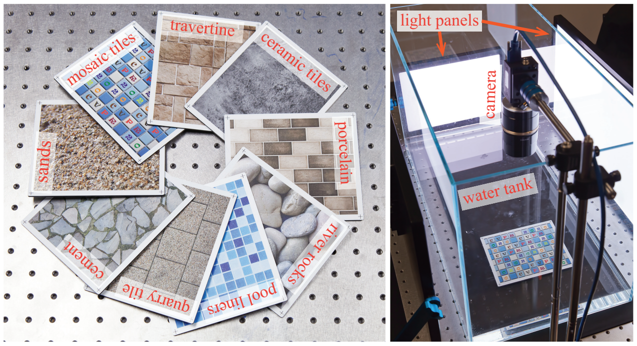

Figure 8. Our experimental setup for real data acquisition. Left: Sample reference patterns that we use for the real experiments; Right: We setup a camera on top of the fluid tank to capture refraction images of the reference pattern.

FSRN-CNN$_2$ that adds the depth-normal loss; and 3) FSRN-CNN$_3$ that also adds the refraction loss. We also compare with the FSRN-CNN subnet without using the recurrent layers. FSRN is our full proposed network that uses both sub-nets with the complete set of loss functions. The quantitative comparison results for both depth and normal estimations are shown in Table 2. We can see that performance of our network gradually improves as we incorporate more loss functions. This indicates that our depth-normal loss, refraction loss, and the recurrent subnet are effective and help improve the accuracy of prediction. We refer the readers to our supplementary material for visual comparisons of our ablation study.

4.3. Experiments on Real Data

We also perform real experiment to evaluate our network. Our experimental setup is shown in Fig. 8. We use a water tank with size 12 × 24 × 18 inches for wave simulation. Our reference pattern is placed at the bottom of the tank. We use a variety of patterns (*e.g.*, Perlin noise, pool liners, river rocks, and sands *etc.*) to test the robustness of our approach. We mount a machine vision camera (FLIR GS3-U3-32S4C-C) to the top to record videos of the water wave. As we assume orthographic camera model, we mount the camera high (around 50cm to the tank bottom) and use a long focal length lens (50mm, Horizontal FoV 8°) to minimize the perspective effect. We also use a small aperture size (f/8) to extend the depth-of-field. We further calibrate the camera [44] . We use the camera intrinsic parameters to remove lens-related distortions and the extrinsic parameters to compensate camera rotations such that the image plane is frontal parallel to the fluid surface. We capture the dynamic fluid video with the reference pattern as background at a frame rate of 121fps and use fast shutter speed 1ms to reduce motion blur. We therefore place four LED light panels to surround the water tank in order to have sufficient light. We finally crop the regions with background pattern from our raw images and use them as input to our network.

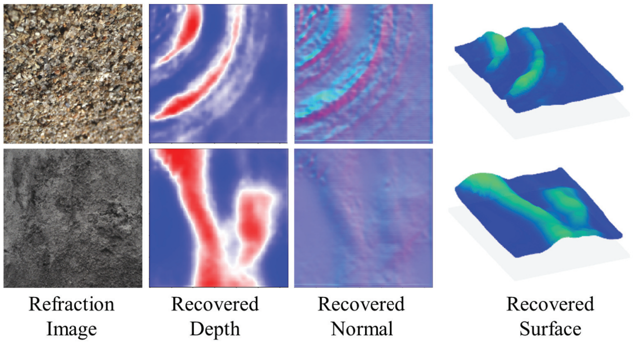

Figure 9. Reconstruction results on real data. Complete video results can be found in the supplementary material.

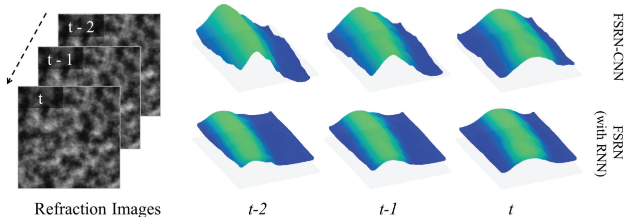

Figure 10. Comparison between FSRN-CNN and our full network FSRN (with RNN).

Our real fluid surface reconstruction results are shown in Fig. 9. We can see that our reconstructions are consistent with the refractive distortions. Please see the supplementary material for videos of recovered dynamic fluid surfaces. We also compare the 3D reconstruction results using FSRN-CNN and our full network FSRN (with the RNN subnet). The reconstruction results for three consecutive frames are shown in Fig. 10. We can see that the FSRN-CNN results obviously change more abruptly while our full network produces a smoother propagation. This indicates that our FSRN-RNN can effectively enforce the temporal consistency in our reconstruction. We also perform re-rendering experiments to demonstrate the accuracy of our approach. We use our recovered fluid surface to re-render the distortion image as seen by the camera. We compare our re-rendered image with the actually captured refraction image (see Fig. 11). We can see that the pattern distortions are highly similar.

4.4. Discussions

Our network is able to achieve good performance on both synthetic and real fluid data although it is trained on a synthetic dataset. This could be due to two reasons: 1) our physics-based fluid dataset preserve characteristics of natural fluid flows thanks to the diversity of fluid equations we use to model the fluid surface and 2) we consider the physics of refraction in our loss function for more accurate reconstruction. However, the refraction loss requires to take the original reference pattern as input. This limits the application of network in outdoor fluid scenes. We can overcome this problem by incorporating a network similar to [25] that first estimates the undistorted pattern and then use it for computing the refraction loss.

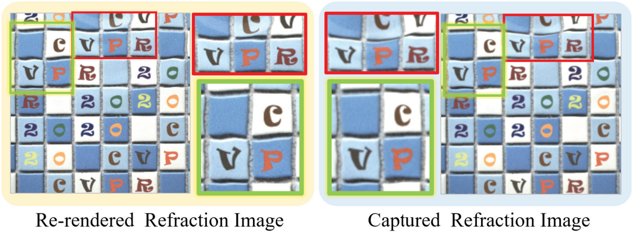

Figure 11. Re-rendered refraction image using our reconstructed fluid surface in comparison with the real captured image with zoom-in views.

In addition, we observe that our network produce more accurate prediction on noisy pattern (*e.g.*, sand and cement textures) than on regular patterns (*e.g.*, checkboard and pool liners). This is because these noisy patterns contain more high-frequency components that better preserve the refractive distortion features.

5. Conclusions

We have presented a deep neural network (FSRN) for dynamic fluid reconstruction from refraction images. We use a convolutional network for depth and normal estimation, and a recurrent network for enforcing the temporal consistency of the dynamic fluid. We consider the depth-normal consistency and the physics of refraction in our loss functions for training. We have also created a large fluid dataset using physics-based fluid modeling and rendering. Through both synthetic and real experiments, we have shown that our network can recover fluid surfaces with high accuracy. One future direction is to generalize our network to arbitrary background to eliminate the use of a reference pattern. We plan to further extend our network to handle more challenging fluid scenes with reflection and scattering. As there's very few work on applying deep learning to non-Lambertian scene, we expect our network and dataset can serve as baseline for studying fluid scenes.

Acknowledgements

This work was partially supported by the Louisiana Board of Regents under Grant LEQSF(2018-21)-RD-A-10. The authors would like to thank Dr. Yu Ji for his help in setting up the real experiments.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016. 6

[2] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *arXiv preprint arXiv:1812.11941*, 2018. 6

[3] Yuta Asano, Yinqiang Zheng, Ko Nishino, and Imari Sato. Shape from water: Bispectral light absorption for depth recovery. In *European Conference on Computer Vision*, pages 635–649, 2016. 1, 2

[4] Aayush Bansal, Bryan Russell, and Abhinav Gupta. Marr revisited: 2D-3D alignment via surface normal prediction. In *Conference on Computer Vision and Pattern Recognition*, pages 5965–5974, 2016. 2

[5] William JD Bateman, Chris Swan, and Paul H Taylor. On the efficient numerical simulation of directionally spread surface water waves. *Journal of Computational Physics*, 174(1):277–305, 2001. 1

[6] Yao-Jen Chang and Tsuhan Chen. Multi-view 3D reconstruction for scenes under the refractive plane with known vertical direction. In *International Conference on Computer Vision*, pages 351–358, 2011. 1, 2

[7] Christopher B Choy, Danfei Xu, Junyoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In *European conference on computer vision*, pages 628–644, 2016. 2

[8] Adrian Constantin and Joachim Escher. Wave breaking for nonlinear nonlocal shallow water equations. *Acta Mathematica*, 181(2):229–243, 1998. 2, 5

[9] Camille Couprie, Clément Farabet, Laurent Najman, and Yann LeCun. Indoor semantic segmentation using depth information. *arXiv preprint arXiv:1301.3572*, 2013. 2

[10] Yuanyuan Ding, Feng Li, Yu Ji, and Jingyi Yu. Dynamic fluid surface acquisition using a camera array. In *International Conference on Computer Vision*, pages 2478–2485, 2011. 1, 2

[11] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *International Conference on Computer Vision*, pages 2650–2658, 2015. 2, 4

[12] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, pages 2366–2374, 2014. 2, 6

[13] Douglas Enright, Stephen Marschner, and Ronald Fedkiw. Animation and rendering of complex water surfaces. In *ACM Transactions on Graphics*, volume 21, pages 736–744, 2002. 1

[14] Ricardo Ferreira, Joao P Costeira, and Joao A Santos. Stereo reconstruction of a submerged scene. In *Iberian Conference on Pattern Recognition and Image Analysis*, pages 102–109, 2005. 1, 2

[15] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2

[16] Georgia Gkioxari, Ross Girshick, and Jitendra Malik. Contextual action recognition with R* CNN. In *International Conference on Computer Vision*, pages 1080–1088, 2015. 2

[17] Kai Han, Kwan-Yee K Wong, and Miaomiao Liu. Dense reconstruction of transparent objects by altering incident light paths through refraction. *International Journal of Computer Vision*, 126(5):460–475, 2018. 2

[18] Ivo Ihrke, Kiriakos N Kutulakos, Hendrik PA Lensch, Marcus Magnor, and Wolfgang Heidrich. Transparent and specular object reconstruction. In *Computer Graphics Forum*, volume 29, pages 2400–2426, 2010. 2

[19] Aaron S Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. Large pose 3D face reconstruction from a single image via direct volumetric cnn regression. In *International Conference on Computer Vision*, pages 1031–1039, 2017. 2

[20] Bernd Jähne, Jochen Klinke, and Stefan Waas. Imaging of short ocean wind waves: a critical theoretical review. *JOSA A*, 11(8):2197–2209, 1994. 1

[21] Yu Ji, Jinwei Ye, and Jingyi Yu. Reconstructing gas flows using light-path approximation. In *Conference on Computer Vision and Pattern Recognition*, pages 2507–2514, 2013. 1, 2

[22] Huaizu Jiang and Erik Learned-Miller. Face detection with the faster R-CNN. In *International Conference on Automatic Face & Gesture Recognition*, pages 650–657, 2017. 2

[23] Bo Li, Chunhua Shen, Yuchao Dai, Anton Van Den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFS. In *Conference on Computer Vision and Pattern Recognition*, pages 1119–1127, 2015. 2

[24] Xiu Li, Min Shang, Hongwei Qin, and Liansheng Chen. Fast accurate fish detection and recognition of underwater images with fast r-cnn. In *OCEANS MTS/IEEE Washington*, pages 1–5, 2015. 2

[25] Zhengqin Li, Zak Murez, David Kriegman, Ravi Ramamoorthi, and Manmohan Chandraker. Learning to see through turbulent water. In *Winter Conference on Applications of Computer Vision*, pages 512–520, 2018. 2, 8

[26] Beyang Liu, Stephen Gould, and Daphne Koller. Single image depth estimation from predicted semantic labels. In *Conference on Computer Vision and Pattern Recognition*, pages 1253–1260, 2010. 2

[27] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2024–2039, 2015. 2

[28] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *International Speech Communication Association*, 2010. 2

[29] Nigel JW Morris and Kiriakos N Kutulakos. Dynamic refraction stereo. *Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1518–1531, 2011. 1, 2

[30] Hiroshi Murase. Surface shape reconstruction of an undulating transparent object. In *International Conference on Computer Vision*, pages 313–317, 1990. 2

[31] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Conference on Computer Vision and Pattern Recognition*, pages 283–291, 2018. 2

[32] Yiming Qian, Minglun Gong, and Yee-Hong Yang. Stereo-based 3D reconstruction of dynamic fluid surfaces by global optimization. In *Conference on Computer Vision and Pattern Recognition*, pages 1269–1278, 2017. 1, 2

[33] Qi Shan, Sameer Agarwal, and Brian Curless. Refractive height fields from single and multiple images. In *Conference on Computer Vision and Pattern Recognition*, pages 286–293, 2012. 2

[34] Evan Shelhamer, Jonathan T Barron, and Trevor Darrell. Scene intrinsics and depth from a single image. In *International Conference on Computer Vision Workshops*, pages 37–44, 2015. 2

[35] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, pages 802–810, 2015. 4

[36] Jos Stam. Stable fluids. In *Siggraph*, volume 99, pages 121–128, 1999. 2

[37] 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*, pages 995–1003, 2019. 2, 6

[38] Jerry Tessendorf. Simulating ocean water. *Simulating Nature: Realistic and Interactive Techniques. SIGGRAPH*, 1(2):5, 2001. 2, 5

[39] Yuandong Tian and Srinivasa G Narasimhan. A globally optimal data-driven approach for image distortion estimation. In *Conference on Computer Vision and Pattern Recognition*, pages 1277–1284, 2010. 2

[40] Peng-Shuai Wang, Chun-Yu Sun, Yang Liu, and Xin Tong. Adaptive O-CNN: a patch-based deep representation of 3D shapes. In *SIGGRAPH Asia 2018 Technical Papers*, page 217, 2018. 2

[41] Xiaolong Wang, David Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *Conference on Computer Vision and Pattern Recognition*, pages 539–547, 2015. 2

[42] Gordon Wetzstein, Ramesh Raskar, and Wolfgang Heidrich. Hand-held schlieren photography with light field probes. In *International Conference on Computational Photography*, pages 1–8, 2011. 1, 2

[43] Jinwei Ye, Yu Ji, Feng Li, and Jingyi Yu. Angular domain reconstruction of dynamic 3D fluid surfaces. In *Conference on Computer Vision and Pattern Recognition*, pages 310–317, 2012. 1, 2

[44] Zhengyou Zhang. A flexible new technique for camera calibration. *Transactions on Pattern Analysis and Machine Intelligence*, 22, 2000. 7