Learning to Remove Refractive Distortions from Underwater Images

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

Abstract

The fluctuation of the water surface causes refractive distortions that severely downgrade the image of an underwater scene. Here, we present the distortion-guided network (DG-Net) for restoring distortion-free underwater images. The key idea is to use a distortion map to guide network training. The distortion map models the pixel displacement caused by water refraction. We first use a physically constrained convolutional network to estimate the distortion map from the refracted image. We then use a generative adversarial network guided by the distortion map to restore the sharp distortion-free image. Since the distortion map indicates correspondences between the distorted image and the distortion-free one, it guides the network to make better predictions. We evaluate our network on several real and synthetic underwater image datasets and show that it out-performs the state-of-the-art algorithms, especially in presence of large distortions. We also show results of complex scenarios, including outdoor swimming pool images captured by drone and indoor aquarium images taken by cellphone camera.

1. Introduction

Underwater scenes, when observed in air, suffer from strong distortion artifacts due to refraction caused by the wavy water surface. Restoring the true underwater images by removing the refractive distortions can benefit numerous tasks in underwater exploration and outer-space expedition (by extending to remove the atmospheric distortions).

However, it is non-trivial to remove the refractive distortions because 1) the geometric deformations are highly non-rigid and discontinuous due to the non-linear light transport through the wavy water surface, and 2) fast-evolving waves also cause blurriness in the image. Classical approaches usually take a long sequence of images (or video) of a static underwater scene, and rely on the mean/median images [31, 30] or the "lucky patch" [16, 14], which happens to be free from distortion in a certain frame, to restore the latent distortion-free image. As these methods require video input of a static scene, they cannot be used for images captured on a moving platform (for example, an underwater vehicle). The seminal work of [34] presents a model-based tracking method to undistort underwater images. But their parametric model cannot be easily tuned and applied to arbitrary waves. Most recently, Li *et al*. [28] propose a learning-based method to correct refractive distortions using a single image. This work demonstrates great potential of using deep neural networks to tackle the challenging problem of refractive distortion removal. But this network does not account for physical constraints and requires a large training set (over 300k images from the ImageNet [10]).

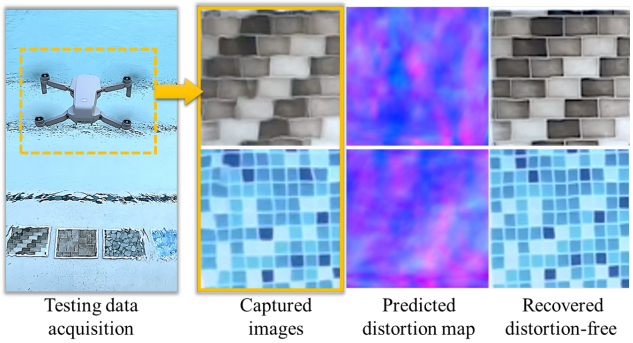

Figure 1. We design a physics-based distortion-guided network for underwater image correction. Our method predicts the distortion-free image, given three distorted underwater images.

In this paper, we present the distortion-guided network (DG-Net) for restoring distortion-free underwater images. The key idea is to use a distortion map to guide network training. The distortion map models the pixel displacement caused water refraction. As the distortion map reveals correspondences between the distorted and distortion-free images, we can use it to guide the network to make better predictions. We first use convolutional neural network (CNN) to estimate the distortion map from the refracted image. Specifically, we design training losses that follow the physical model of refractive distortions. We also exploit the temporal consistency of the distortion map by taking three sequential images as input. We use three parallel CNNs

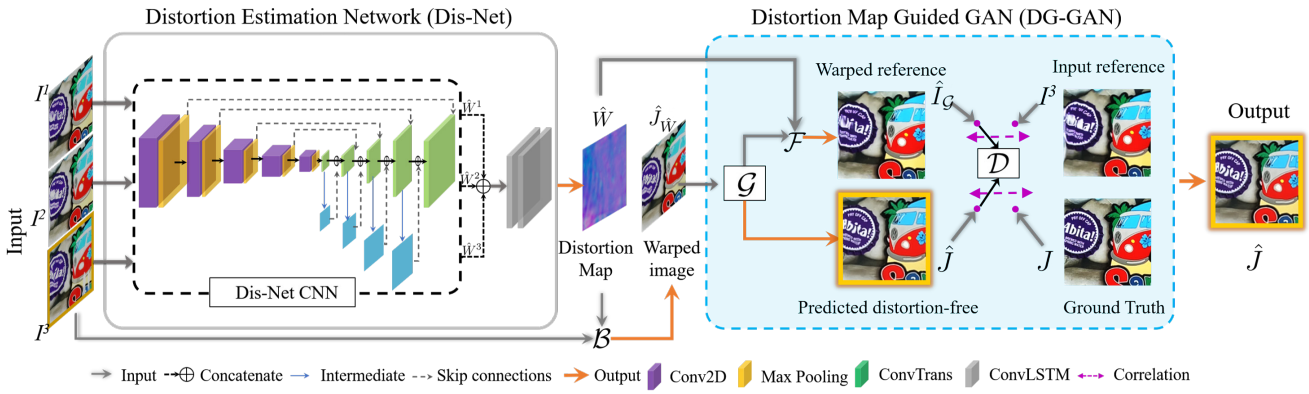

Figure 2. The overall architecture of DG-Net. It consists of two subnets: a convolutional network for estimating the refractive distortions (Dis-Net) and a distortion-guided generative adversarial network for restoring the distortion-free image (DG-GAN). Note that we have three CNN networks in Dis-Net which takes each of three inputs. The generator and discriminator of DG-GAN are represented by $\mathcal{G}$ and $\mathcal{D}$, respectively. $\mathcal{F}$ and $\mathcal{B}$ denotes forward and backward mapping of images.

to generalize features from each input, and then use recurrent layers to refine the CNN-predicted distortion maps by enforcing the temporal consistency among them. We can use the estimated distortion map to correct slight refractive distortions. Since large distortion are non-invertible (due to many-to-one mapping), we then use a distortion-guided generative adversarial network (GAN) to recover sharp distortion-free image. The distortion map is used to guide the training of both generator and discriminator of the GAN. Our network is trained on a synthetic refracted image dataset, with patterns that resemble the underwater scenes.

We evaluate the DG-Net on our own synthetic dataset and several real captured underwater image datasets [24, 33, 34]. The results show that our method out-performs the state-of-the-arts [23, 24, 28, 30, 34], especially in presence of large distortions. Compared with the model-based methods [24, 34, 30], we do not need long video sequence of a static underwater scene to achieve accurate reconstruction. Although we still take three images to exploit the temporal constraints, the images can be captured with the burst mode in a very short time interval. Our method can therefore be used for dynamic scenes such as videos from a flying drone and videos of aquatic scenes with moving objects. Compared with the learning-based methods [23, 28], our network requires fewer training data (around one tenth in size), but achieves better accuracy in presence of large distortions and generalizes well on real scenes.

2. Related Work

Recovering underwater images. The problem of recovering faithful underwater images is critical to underwater imaging. Early solutions [14, 27] take the mean/median of a distorted image sequence to approximate the latent distortion-free image. Although these methods work well on weak distortions, the mean image becomes blurry in presence of large distortions. Another popular class of methods rely on finding and stitching the "lucky patches" to recover the latent distortion-free image. Many solutions such as clustering [11, 12], manifold embedding [14], and Fourier-based averaging [39] are proposed to locate the "lucky patch" in the input sequence. The seminal work of [34, 35] presents a model-based tracking method to restore underwater images. Oreifej *et al*. [30] propose a two-step algorithm that first iteratively aligns the distorted images to the mean image and then denoises the estimation with low-rank constraint. More recently, James *et al*. [24] propose a compressed sensing (CS) solver for underwater image restoration by tracking a few salient feature points across the frames of a video sequence of the submerged scene. All these methods require a long sequence of distorted images (∼60 to 100 frames) as input and cannot work for single or few images. Li *et al*. [28] propose a generative adversarial network to correct refractive distortions using a single image. In this work, our proposed network consider the physical model of refractive distortions, and use the distortion maps as training guidance. Our method can recover high quality distortion-free image with three input images.

Estimating pixel displacement between images. The problem of estimating pixel displacement has been extensive studied in motion/flow estimation. Most methods [19, 2] in this category consider rigid motion and estimate the displacement vectors through matched corresponding features. Recent trend is to use deep neural networks to tackle this problem. The FlowNet [5, 13, 29] is proposed to estimate the shift between two consecutive images. Kanazawa *et al*. [25] propose the WarpNet to match invariant features between cross-category images. However, the refractive distortions caused by wavy water is highly non-rigid and it is difficult to find invariant features from the distorted images. Xue *et al*. [41] adapt the classical opti-

cal flow to estimate small refractive distortions caused by hot air or gas. In this work, we propose a physically constrained convolutional network with recurrent layers to estimate large refractive distortions caused by wavy water.

Image-to-image generation. The generative adversarial networks (GANs) [17] have shown great success in solving image-to-image generation problems, such as image super-resolution [40, 4, 45, 38], denoising [47, 43, 6], deblurring [26, 46], inpainting [9, 37], *etc*. The key idea is to use an adversarial discriminator network (discriminator) to pit against the generative network (generator) and force the generator to produce realistic images. Most existing GANs are trained with images of natural scenes [28, 3, 42, 8] or human faces [44, 36, 20], and usually is trained on a large dataset (with millions of images). In contrast, our GAN is trained with patterns that resemble the underwater environment. In addition, we use the refractive distortions to guide the training of both generator and discriminator. As result, our network requires fewer training data ($\sim$ 50k images), but can achieve better accuracy.

3. Proposed Method

We consider the setting that a camera is looking at the underwater scene through the wavy water surface. The captured images therefore suffer from refractive distortions. Assume J is the true image of the underwater scene unaffected by the water waves, our goal is to estimate a distortion-free image $\hat{J}$ that appears close to J from the captured distorted images I.

We propose a distortion-guided network (DG-Net) to tackle this problem. Specifically, ground truth distortion maps are used to guide the training of our networks. The overall structure of our network is shown in Fig. 2. Our DG-Net has two subnets: a convolutional network for estimating the refractive distortions (Section 3.1) and a distortion-guided generative adversarial network for restoring the distortion-free image (Section 3.2). The two subnets are trained separately. Notice that although our network takes three sequential images $\{I^t\}_{t=1}^3$ as input, we only output one distortion-free image for the last frame (I^3). The first two frames are used for enforcing the temporal consistency of our distortion estimation. Unlike classical methods that require a video of static scene, our method can be used for moving scenes as the three sequential images can be captured with the burst mode in a very short time interval.

3.1. Distortion Estimation

We first use a distortion estimation network (Dis-Net) to predict the distortion map between the input distorted image and the latent distortion-free image. Our Dis-Net considers the physical model of refractive distortions and use temporal constraints to improve the estimation accuracy.

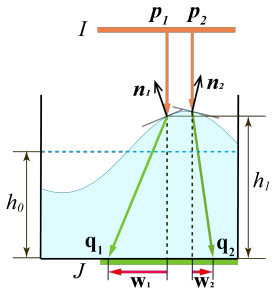

Figure 3. Illustration of refractive distortion. I and J are distorted and distortion-free images respectively; $\mathbf{p}_1$, $\mathbf{q}_1$ and $\mathbf{p}_2$, $\mathbf{q}_2$ are two pairs of corresponding pixels; h_0 is the average water surface height, and h_1 is the height at the pixel $\mathbf{p}_1$; $\mathbf{n}_1$ and $\mathbf{n}_2$ are normal vectors; $\mathbf{w}_1$ and $\mathbf{w}_2$ are distortion vectors.

Refractive distortion model. Given a distorted image I and the true distortion-free image J, we define a distortion map $W = \{\mathbf{w}_i\}_{i=1}^M$ (where $\mathbf{w}_i \in R^2$ is per-pixel distortion vector and M is the total number of pixels) to represent the the pixel displacement between I and J caused by the refraction of water-air interface. $\mathbf{w}_i$ can then be written as:

$$\mathbf{w}_i = \mathbf{q}_i - \mathbf{p}_i \tag{1}$$

where $\mathbf{p}_i \in R^2$ is a pixel in I, and $\mathbf{q}_i \in R^2$ is a pixel in J. $\mathbf{q}_i$ maps to $\mathbf{p}_i$ through refraction.

Since the refractive distortion is caused by the fluctuation of water surface, the amount of distortion (or pixel placement) is naturally related to the water surface height. By applying the first-order approximation of the Snell's law, Tian and Narasimhan derive that the distortion vector $\mathbf{w}_i$ has linear relationship with the gradient of surface height [34]. The mapping from the surface height map $H = \{h_i\}_{i=1}^M$ (where $h_i \in R$ is a height value) to the distortion map W can be written as:

$$W = f(H) = \alpha \nabla H \tag{2}$$

where $\nabla . = [\frac{\partial .}{\partial x}, \frac{\partial .}{\partial y}]$ is the gradient operator, and $\alpha = h_0(1 - \frac{1}{n})$ is a constant scalar determined by the average surface height h_0 and the refractive index n. The inverse mapping from W to H can then be found by integrating the distortion vectors:

$$\begin{aligned} H = f^{-1}(W) &= h_0 + \iint_{x,y} \nabla H d_x d_y \\ &= \frac{\alpha \cdot n}{n-1} + \iint_{x,y} \frac{W}{\alpha} d_x d_y \end{aligned} \tag{3}$$

As surface normals are related to the 2D height gradients, we can also derive the normal map $N = \{\mathbf{n}_i\}_{i=1}^M$ (where $\mathbf{n}_i \in R^3$ is a normal vector) from the distortion map W as:

$$\mathbf{n}_i = \gamma_i [-\frac{\mathbf{w}_i(x)}{\alpha}, -\frac{\mathbf{w}_i(y)}{\alpha}, 1] \tag{4}$$

where $\gamma_i = 1/\|\mathbf{n}_i\|$ is a normalization factor.

Given the surface height map H and the distortion-free image J, the ground truth distortion map W can be found by backward tracing rays from the image plane through the water surface to the underwater image J, as shown in Fig. 3. We use the ground truth distortion maps as well as physics-based losses derived from our refraction model to guide the training of the Dis-Net.

Network structure. The Dis-Net takes three distorted images $\{I^t\}_{t=1}^3$ as input, and output one distortion map prediction $\hat{W}$ for the last frame (I^3). The network consists of three concatenated convolutional neural networks (CNNs) followed by two recurrent layers (see Fig. 2).

Note that our network can be easily modified to take arbitrary number of images (by adding or reducing the CNN branches). We find that three images are sufficient to achieve decent performance even in presence of large distortions. Adding more input images results in more network parameters, but the performance gain is marginal.

The structure of a CNN branch is shown in Fig. 2. Each CNN estimate a distortion map from one distorted image. The encoder of our CNN is made up of standard stacked convolutional layers with max-pooling. The decoder uses variational refinement [13] to preserve fine details in the distortion map. Specifically, at each layer, we concatenate the transpose-convolved feature map, the corresponding feature map from the encoder, and an intermediate distortion map output by the current feature map. The intermediate distortion maps are compared with downsampled ground truth maps using our training losses.

The three distortion maps $\{\hat{W}^t\}_{t=1}^3$ output from the CNNs are concatenated as a temporal sequence and fed into two stacked convolutional LSTM layers with batch normalization. The convolutional LSTM layers transmit hidden states from previous time frames to learn the temporal dependencies [32]. By exploiting the temporal consistency among the distortion maps, the prediction accuracy is further improved, which is shown in the ablation study (see Section 4.3).

Loss functions. The Dis-Net takes ground truth distortion free image (J), distortion map (W) and surface height map (H) for training. We design loss functions following the refractive distortion model. Our loss functions consists of three terms: the distortion map loss, the refraction loss, and the consistency loss.

The *distortion map loss* has three components. For each component, we use the scale-invariant error function [15] to measure the difference between two distortion maps:

$$\varepsilon(W, W^*) = \frac{1}{M}\sum_{i=1}^{M}(\mathbf{w}_i - \mathbf{w}_i^*)^2 - \frac{1}{2M^2}(\sum_{i=1}^{M}(\mathbf{w}_i - \mathbf{w}_i^*))^2 \tag{5}$$

where $\mathbf{w}_i$ is a distortion vector in W; $\mathbf{w}_i^*$ is a distortion vector in W^*; M is the total number of pixels.

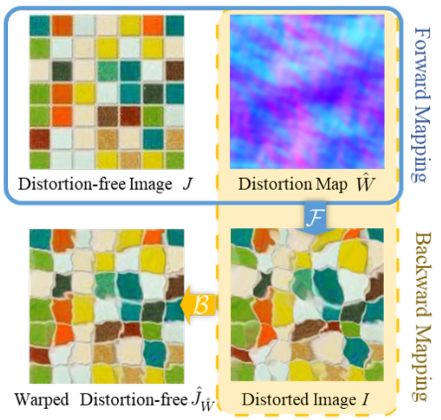

Figure 4. Illustration of forward $\mathcal{F}$ and backward mapping $\mathcal{B}$. In presence of large distortions, backward mapping alone with the correct distortion map are insufficient to obtain an accurate distortion-free image.

Intuitively, we compare the predicted distortion map $\hat{W}$ with the ground truth one and calculate the error $\varepsilon(W, \hat{W})$. Since the distortion map is directly related to water surface depth and normal, we consider two additional errors that are constrained by the physical models. Specifically, by applying Eq. 2 to H, we can obtain another distortion map W_H converted from the ground truth height. We compare $\hat{W}$ with W_H to enforce their consistency. By applying Eq. 3 and Eq. 4 to $\hat{W}$, we can map our predicted distortion map to its corresponding height map $\hat{H}$ and normal map $\hat{N}$. We can then apply backward ray tracing and obtain a new distortion map $W_{\hat{H}}$. We compare $W_{\hat{H}}$ with the ground truth map W. Since our converted height $\hat{H}$ is accurate, the two map should be consistent. In sum, our distortion map loss can be written as $\mathcal{L}_W = \alpha_1\varepsilon(W, \hat{W}) + \alpha_2\varepsilon(W_H, \hat{W}) + \alpha_3\varepsilon(W, W_{\hat{H}})$, where $\alpha_{1,2,3}$ are weighting factors.

The *refraction loss* minimizes the difference between the input image I and the distorted image $I_{\hat{W}}$ traced with the height map $\hat{H}$ mapped from $\hat{W}$. We use the $l2$ norm as error metric: $\varepsilon_{l2}(I, I^*) = \frac{1}{M}\sum(I - I^*)^2$. Our refraction loss is therefore written as $\mathcal{L}_R = \varepsilon_{l2}(I, I_{\hat{H}})$.

The *consistency loss* enforces consistent estimations from the three parallel CNNs. As three inputs $\{I^t\}_{t=1}^3$ are captured in a short time interval, we assume their latent distortion-free images are the same. Specifically, we use the predicted distortion maps $\{\hat{W}^t\}_{t=1}^3$ to undistort their corresponding inputs by applying Eq. 1 and obtain $\{\hat{J}^t_{\hat{W}^t}\}_{t=1}^3$. We use the $l2$ error to compare pairwise difference among $\{\hat{J}^t_{\hat{W}^t}\}_{t=1}^3$. The consistency loss is therefore written as $\mathcal{L}_C = \frac{1}{3}\sum_{t,s=1}^3 \varepsilon_{l2}(\hat{J}^t_{\hat{W}^t}, \hat{J}^s_{\hat{W}^s})$.

We combine $\mathcal{L}_W$, $\mathcal{L}_R$, and $\mathcal{L}_C$ to train the Dis-Net. The training is performed end-to-end. The CNNs and the recurrent layers use different sets of weights for the losses.

3.2. Image Restoration

Given the estimated distortion map $\hat{W}$, we propose a distortion-guided adversarial network (DG-GAN) to estimate the distortion-free image $\hat{J}$. By directly applying $\hat{W}$ to undistort the input distorted image I, we can obtain an intermediate image $\hat{J}_{\hat{W}} = \mathcal{B}(I, \hat{W})$, where $\mathcal{B}$ refers to backward mapping:

$$\mathcal{B}(I, W) = I(\mathbf{p} - \mathbf{w}) \tag{6}$$

We use this warped image $\hat{J}_{\hat{W}}$ as input to the DG-GAN. Although $\hat{J}_{\hat{W}}$ appear less distorted than I, some large distortions cannot be inverted as several pixels in J may map to one pixel in I through refraction as shown in Fig. 4.

Our DG-GAN has similar structure to the conditional GAN [23], but adopts distortion-guided training losses. The generator $\mathcal{G}$ uses the "U-Net" as base architecture. It has 6 convolutional layers in the encoder and 6 deconvolutional layers with skip connections in the decoder. $\mathcal{G}$'s goal is to produce distortion-free images $\hat{J}$ that cannot be distinguished from "real" by the discriminator. $\mathcal{G}$ is trained with both the $l1$ and $l2$ losses that force its output to appear similar to the ground truth distortion-free image J. The $l1$ loss encourages less blurring and help to generate sharper image. In addition, we can apply the ground truth distortion map W on $\mathcal{G}$'s output to obtain a distorted image $\hat{I}_{\mathcal{G}} = \mathcal{F}(\mathcal{G}(\hat{J}_{\hat{W}}), W)$, where $\mathcal{F}$ refers to forward mapping:

$$\mathcal{F}(I, W) = I(\mathbf{p} + \mathbf{w}) \tag{7}$$

If $\mathcal{G}$'s output appears similar to J, then $\hat{I}_{\mathcal{G}}$ should be consistent with the input distorted image I. The loss function for training $\mathcal{G}$ is therefore written as:

$$\begin{aligned}\mathcal{L}_{\mathcal{G}} =& \frac{1}{M}\Big(\sum |\mathcal{G}(\hat{J}_{\hat{W}}) - J| \\ &+ \sum (\mathcal{G}(\hat{J}_{\hat{W}}) - J)^2 + \sum (\hat{I}_{\mathcal{G}} - I)^2\Big)\end{aligned} \tag{8}$$

The discriminator $\mathcal{D}$ is adversarially trained to identify the "fake" hallucinated images from generator. Our $\mathcal{D}$ is formed with 6 modules of the form convolution-BatchNorm-ReLu modules [22]. Besides learning a mapping from the input $\hat{J}_{\hat{W}}$ to the distortion-free image J, the network also learns to predict whether the distortion constraint is satisfied. Specifically, we apply the ground truth distortion map W to $\hat{J}_{\hat{W}}$, the discriminator $\mathcal{D}$ then favors predictions that appear closer to input distorted image I, instead of the forward mapping result $\hat{I}_{\mathcal{G}}$ of $\mathcal{G}$'s output. The objective function of our DG-GAN can be written as:

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(\mathcal{G}, \mathcal{D}) =& E[log\mathcal{D}(\hat{J}_{\hat{W}}, J)] + E[log\mathcal{D}(\mathcal{F}(\hat{J}_{\hat{W}}, W), I)] \\ &+ E[log(1 - \mathcal{D}(\hat{J}_{\hat{W}}, \mathcal{G}(\hat{J}_{\hat{W}})))] \\ &+ E[log(1 - \mathcal{D}(\mathcal{F}(\hat{J}_{\hat{W}}, W), \hat{I}_{\mathcal{G}}))]\end{aligned} \tag{9}$$

The corrected distortion-free image is then optimized as $\hat{J} = \arg\min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{L}_{\text{GAN}}$.

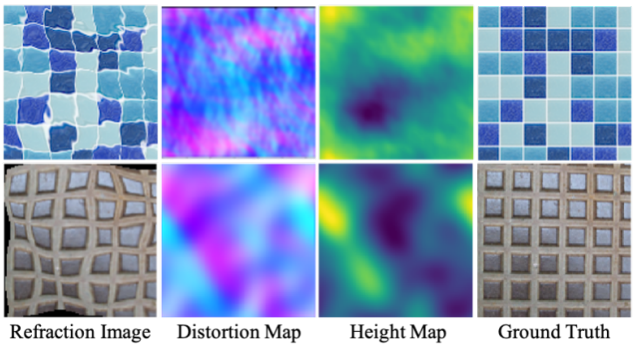

Figure 5. Sample images of our synthetic underwater image dataset. From left to right, we show the ground truth (GT) distortion-free image, GT height map, GT distortion map, and the refraction image.

4. Experiments

In this section, we evaluate our DG-Net on both synthetic and real underwater image datasets. Specifically, we compare our method with competitive state-of-the-art methods, and perform ablation studies on our network.

4.1. Network Training

Data preparation. Our DG-Net is trained on a synthetic underwater dataset. We generate the dataset using physics-based modeling and rendering. Specifically, we use partial derivative equations derived from the Navier-Stokes to model the water waves. We consider waves with various heights and fluctuations to create distortions of different scales. To show that our method is robust to different types of water wavefronts, we simulate three types of waves: ripple waves, ocean waves, and Gaussian waves. More details of these wave equations can be found in the supplementary material. Fig. 5 shows exemplary water distortion images and their corresponding distortion maps and height maps.

The distortion-free underwater images are chosen from the Describable Textures Dataset (DTD) [7]. DTD contains a broad range of realistic texture images. We select a subset from the DTD, whose appearance resembles the underwater scenes (for example, pool tiles, sea plants, pebbles, *etc.*). In addition, we add $\sim$ 500 various text images to our set as underwater patterns. Altogether, our dataset contains $\sim$63k distorted refraction images, generated from 6354 unique distortion-free images (or reference pattern). We keep 10 consecutive frames per wave. For each refraction image, we provide the ground truth distortion-free image, distortion map, and height map of the water surface. We divide our dataset as 70% for training (43,600), 15% for validation (9980), and 15% for testing (9960). Note that all the waves and reference patterns are non-overlapping among the training set, validation set, and testing set.

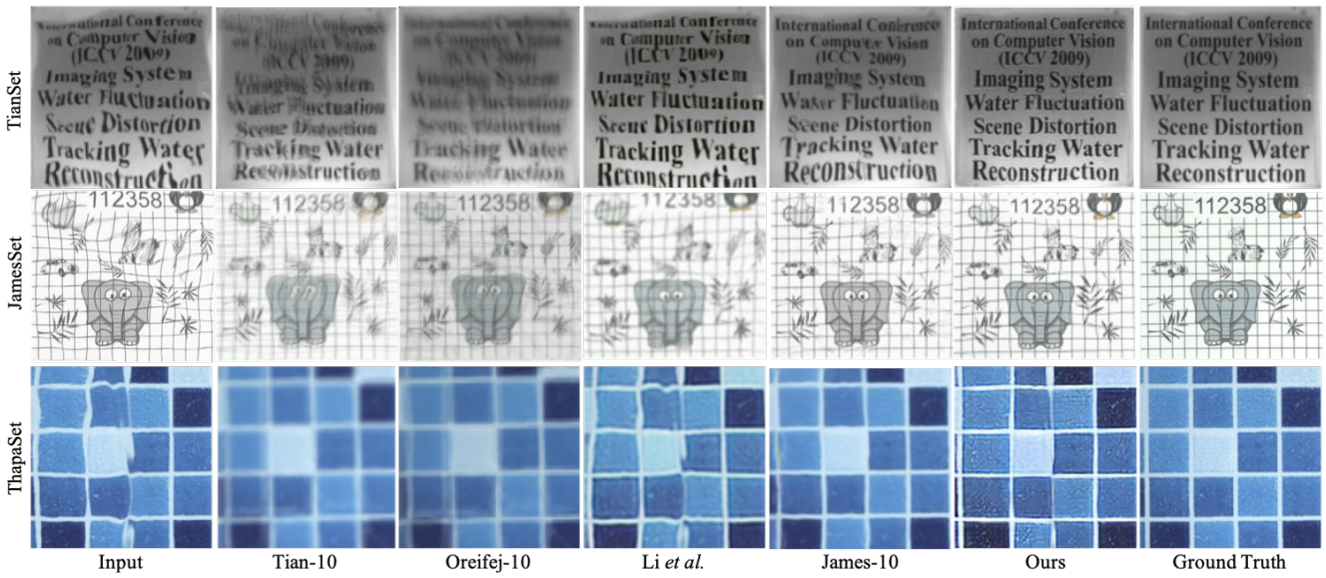

Figure 6. Visual comparison with the state-of-the-arts on real captured datasets: the TianSet [34] (top), the JamesSet [24] (middle), and the ThapaSet [33] (bottom). Here Tian-10, Oreifej-10, and James-10 refer to using a 10-frame sequence as input to the methods [34], [30], and [24], respectively. Please see our supplementary material for more visual comparison results.

Implementation details. We implement our network with TensorFlow [1]. The overall network (DG-Net) has around 50 million trainable parameters, which includes 3.1 million for the Dis-Net, 41 million for the generator of DG-GAN, and 6.9 million for the discriminator of DG-GAN. All computations are performed with a desktop computer with Xeon E5-2620 CPU and two NVIDIA GTX 1080 Ti GPUs.

The DG-Net is trained in two steps. We first train the distortion estimation network (Dis-Net) on the synthetic training set. We set the weights ~ 0.55, ~ 0.25, and ~ 0.15 for the distortion map loss $\mathcal{L}_W$, refraction loss $\mathcal{L}_R$, and consistency loss $\mathcal{L}_C$, respectively. We use the Adam optimizer to train the network. We use batch size 64 for both training and validation with the learning rate of 10^{-4}. We train the network with the loss functions described in Section 3.1 for 60 epochs until converge.

We then train the distortion-guided generative adversarial network (DG-GAN) for restoring the distortion-free image. We first use the estimated distortion map to backward map the distorted image to an intermediate undistorted image, and then use it as input to the DG-GAN. We use the Adam optimizer to train DG-GAN with a fixed learning rate of 2×10^{-4}. We train the network with the loss functions described in Section 3.2 for around 400 epochs that suffices to produce good predictions.

4.2. Comparison with the State-of-The-Arts

We compare our methods with the state-of-the-art underwater image restoration methods [34, 30, 23, 28, 24]. Specifically, Tian and Narasimhan [34], and Oreifej *et al*. [30] are two classical model-based approaches. Tian and Narasimhan[34] use parametric models of distortion to restore the images. Oreifej *et al*. [30] perform per-frame registration with the mean image. James *et al*. [24] is a recently proposed compressed sensing (CS) solver for underwater image restoration. All these methods require a long input sequence to achieve good performance.

Isola *et al*. [23] and Li *et al*. [28] are learning-based methods for image generation/restoration. Isola *et al*. [23] is a general-purpose pixel-to-pixel image generation network. It has good performance on style transfer, image coloring and inpainting. Li *et al*. [28] is an adversarial network specifically for restoring refracted images, trained with ∼300k images from the ImageNet.

Testing datasets. We perform experiments on four dataset (one synthetic and three real): 1) **SynSet**: our own synthetic dataset with 9960 testing images (generated with 996 different reference patterns); 2) **TianSet**: a real captured dataset by Tian and Narasimhan [34]; 3) **JamesSet**: a real captured dataset by James *et al*. [24], in which we test on three videos: Cartoon, Elephant, and Eye; and 4) **ThapaSet**: a real captured dataset by Thapa *et al*. [33]. The TianSet contains four real captured videos with refractive distortions. The four sequences use different reference patterns and each sequence has 61 frames. In our experiments, we also test Tian *et al*. [34], Oreifej *et al*. [30] and James *et al*. [24] on shorter sequences with 10 consecutive frames. We take three real underwater scenes from the ThapaSet. We also test [34], [30], and [28] on the ThapaSet for further comparisons. We perform both qualitative and quantitative evaluations on the image restoration results.

	Methods	PNSR↑	SSIM↑	SSD↓	SSDG↓
SynSet	Isola *et al.*	18.680	0.300	0.0136	0.0068
	Li *et al.*	19.250	0.425	0.0118	0.0055
	DG-Net (Ours)	**24.069**	**0.800**	**0.0039**	**0.0019**
TianSet	Tian-61	16.778	0.810	0.0210	0.0107
	Tian-10	16.402	0.740	0.0229	0.0112
	Oreifej-61	20.457	0.820	0.0090	0.0047
	Oreifej-10	15.884	0.550	0.0258	0.0125
	Isola *et al.*	10.008	0.400	0.0998	0.0490
	Li *et al.*	10.087	0.510	0.0985	0.0486
	James-61	**20.223**	0.753	**0.0095**	**0.0049**
	James-10	16.556	0.721	0.0221	0.0109
	DG-Net (Ours)	19.586	**0.840**	0.0110	0.0056
JamesSet	Tian-61	17.099	0.787	0.0195	0.0095
	Tian-10	15.086	0.574	0.0310	0.0153
	Oreifej-61	15.272	0.765	0.0297	0.0141
	Oreifej-10	14.948	0.559	0.0315	0.0150
	Li *et al.*	12.226	0.662	0.0599	0.028
	James-61	**20.779**	**0.927**	**0.0084**	**0.0041**
	James-10	16.785	0.512	0.0209	0.0098
	DG-Net (Ours)	20.227	0.902	0.0095	0.0052
ThapaSet	Tian-61	23.187	0.827	0.0048	0.0029
	Tian-10	22.076	0.909	0.0062	0.0031
	Oreifej-61	23.372	0.875	0.0046	0.0025
	Oreifej-10	20.506	0.903	0.0089	0.0034
	Li *et al.*	21.426	0.950	0.0072	0.0020
	James-61	**26.778**	0.948	**0.0210**	**0.0013**
	James-10	23.979	0.935	0.0401	0.0019
	DG-Net (Ours)	26.021	**0.951**	0.0251	**0.0013**

Table 1. Quantitative comparison with the state-of-the-arts.

Evaluation metrics. We use four standard image quality/ similarity metrics for quantitative evaluation: 1) Peak Signal-to-Noise Ratio (PSNR) [21], 2) Structural Similarity Index (SSIM) [18], 3) Sum Squared Difference (SSD) [30], and 4) SSD in Gradient (SSDG) [30]. See our supplementary material for equations of these metrics.

Comparison results. Fig. 6 shows the qualitative comparisons between our method and the state-of-the-arts. We show comparison results on three real captured datasets. Please see our supplementary material for more visual comparison results on both synthetic and real datasets. Note that here we show results for Tian and Narasimhan [34], Oreifej *et al.* [30], and James *et al.* [24] with 10 input frames (referred to as Tian-10, Oreifej-10 and James-10 in Fig. 6). We can see that their recovered images are severely downgraded because the input sequence is too short. We can also see that our method, which uses only three input frames, outperforms all the other methods in terms of distortion correction capacity and image sharpness, and produces results with the best visual quality. For qualitative comparisons with [34, 30, 24] using 61 input frames, please refer to our supplementary material.

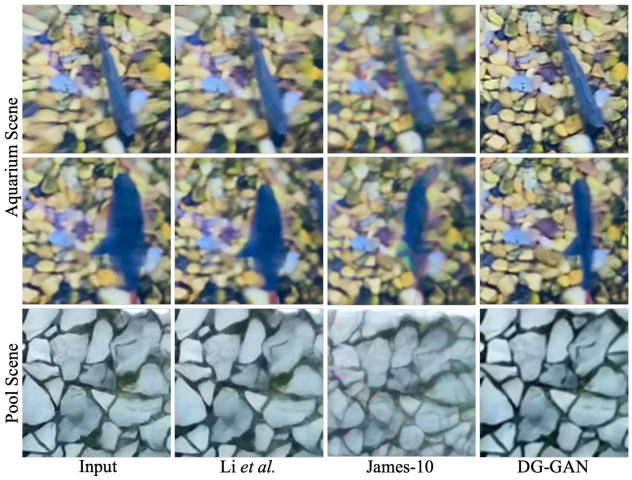

Figure 7. Comparison on the "in-the-wild" data. The top two rows are from the aquarium scene captured by a cellphone camera. The bottom row is from the pool scene captured by a drone camera.

Table 1 shows quantitative comparisons of all methods on both the synthetic and real captured datasets (SynSet, TianSet, JamesSet and ThapaSet). For fair comparison, Isola *et al.* [23] is trained on our dataset. We can see that this general purpose GAN [23] does not work well on refractive distortion correction. Our method achieves the best performance on most of the metrics. It's worth noting that in TianSet, the PSNR of our results is lower than some model-based methods [30, 24] with 61 input frames on several datasets because the PSNR favors blurry images. From the qualitative comparison, we can see that our results are much sharper than those methods and in fact have better visual appearance. Also, some of our results have higher SSIM but lower PSNR. This is because PSNR and SSIM have different degrees of sensitivity to different forms of image degradation. For the task of distortion correction, the estimated distortion-free might have slight misalignment. As our results are usually sharp, such misalignment may lead to drastic decrease in PSNR. On the other hand, SSIM is window-based and less sensitive to the misalignment.

"In-the-wild" experiments. The existing methods refractive distortion correction [24, 34, 30] are mostly tested on static and planar image patterns because they require long sequence to restore the underwater image. However, in real scenarios, either the viewing camera or the underwater objects might be moving. To evaluate the robustness of our method on these challenging scenarios, we perform experiments under two settings 1) a moving camera looking at a static underwater scene and 2) a stationary camera looking at a dynamic underwater scene. For the first setting, we use the DJI Mavic Mini drone camera to capture videos (at 60 fps) of an outdoor swimming pool with static underneath patterns (see Fig. 1). For the second setting, we

Methods	Image Restoration			
	PSNR↑	SSIM↑	SSD↓	SSDG↓
Dis-Net$_W$	15.010	0.358	0.0315	0.0162
Dis-Net$_R$	17.853	0.407	0.0164	0.0089
Dis-Net$_C$	18.090	0.422	0.0155	0.0077
Dis-Net	20.015	0.610	0.0099	0.0052
DG-Net	**24.069**	**0.800**	**0.0039**	**0.0019**

Table 2. Quantitative ablation on physics-based loss terms.

use a cellphone camera to capture videos (at 120 fps) of an indoor aquarium with swimming fish. We compare our results with Li *et al*. [28] and James *et al*. [24] (with 10 input frames). The results are shown in Fig. 7 (more results are included in the supplementary material). We can see that our method works well on these challenging scenarios. In particular, our restoration results have consistent fish shape in the aquarium scenes. While Li *et al*. [28] fails to correct some distortions, and results of James *et al*. [24] are blurry.

4.3. Ablation Studies

Effect of the physical constrains. We first perform ablative experiments on the physics-based loss terms described in Section 3.1 to show their effectiveness. We compare our full network (DG-Net) with the Dis-Net (without the distortion-guided GAN), and three variants of the Dis-Net: 1) Dis-Net$_W$, which removes the last two terms of $\mathcal{L}_W$ (notice that these terms are constrained by our physical model); 2) Dis-Net$_R$, which removes the refraction loss $\mathcal{L}_R$ in Dis-Net; and 3) Dis-Net$_C$, which removes the consistency loss $\mathcal{L}_C$ in Dis-Net. The quantitative comparisons are shown in Table 2. Qualitative comparisons are shown in our supplementary material. We can see that all loss terms contribute to improve our network performance.

Effect of the temporal constraint. To evaluate the effect of the temporal constraint, we create a single input version (DG-Net-S) of our full network by removing the recurrent layers and keeping only one CNN branch. Here we compare with Li *et al*. [28], as it takes a single image as input. We perform experiments on the SynSet. In addition to the recovered images, we also compare the estimated distortion map. We use the root mean square error (RMSE) and the absolute relative error (Abs Rel) to evaluate the distortion map estimation. The quantitative comparison is shown in Table 3. We can see that our full network achieves the best performance. Even our single input version achieves better result than Li *et al*. [28]. We therefore conclude that using the recurrent layer to exploit temporal consistency helps improve the performance.

Effect of refraction distortion constraint. To evaluate the effect of using distortion map as guidance, we create two variants of our network: 1) our network without distortion guidance (noted as ours w/o DG) and 2) our network without the DG-GAN (that leaves the Dis-Net alone). We compare our full network with the two variants, as well as Li *et al*. [28] and Isola *et al*. [23]. Experiments are performed on refracted images with different distortion levels. We categorize the SynSet into seven distortion levels (where level 0 indicates distortion-free, and level 7 indicates the strongest distortions). We quantify the distortion level using the average magnitude of the distortion map of the input image. Fig. 8 compares the PSNR of recovered images from all methods at different distortion levels. See our supplementary material for details on how we compute the distortion levels and the visual comparison results. We can see that, in contrast to the other methods without distortion guidance, our method stays relatively robust for all distortion levels. Although certain distortions still persist when the input images have high level of distortions, our method still largely improve the image quality and make the underwater scene discernible. This is especially important for text scenes.

Methods	Distortion Map		Image Restoration			
	RMSE↓	AbsRel↓	PSNR↑	SSIM↑	SSD↓	SSDG↓
Li *et al*.	0.1089	0.082	19.251	0.425	0.0118	0.0055
DG-Net-S	0.0872	0.070	21.198	0.500	0.0076	0.0041
DG-Net	**0.0624**	**0.038**	**24.069**	**0.800**	**0.0039**	**0.0019**

Table 3. Quantitative ablation on the temporal consistency.

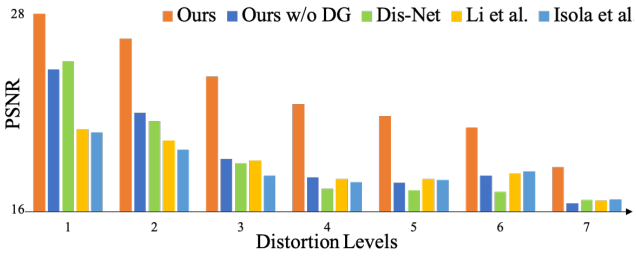

Figure 8. Comparisons with respect to the distortion levels.

5. Conclusions

We presented a physically constrained distortion-guided network (DG-Net) for correcting refractive distortions. We first use a convolutional network that exploits the physical model of refractive distortions for estimating the distortion map. We then use a GAN to restore sharp distortion-free image by using the estimated distortion map as guidance. Experimental results demonstrated that our method generalizes well on real scene and has the capacity of handling challenging scenarios.

Acknowledgements

This project is supported by NSF award CRII-1948524, Louisiana Board of Regents grant LEQSF(2018-21)-RD-A-10, and a gift from DGene.

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 *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2016. 6

[2] John L. Barron, David J. Fleet, and Steven S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision (IJCV)*, 12(1):43–77, 1994. 2

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 3

[4] Adrian Bulat, Jing Yang, and Georgios Tzimiropoulos. To learn image super-resolution, use a GAN to learn how to do image degradation first. In *Proceedings of the European conference on computer vision (ECCV)*, 2018. 3

[5] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012. 2

[6] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[7] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 5

[8] Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. Towards diverse and natural image descriptions via a conditional GAN. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 3

[9] Ugur Demir and Gozde Unal. Patch-based image inpainting with generative adversarial networks. *arXiv preprint arXiv:1803.07422*, 2018. 3

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 1

[11] Arturo Donate, Gary Dahme, and Eraldo Ribeiro. Classification of textures distorted by waterwaves. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2006. 2

[12] Arturo Donate and Eraldo Ribeiro. Improved reconstruction of images distorted by water waves. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP)*, 2006. 2

[13] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015. 2, 4

[14] Alexei Efros, Volkan Isler, Jianbo Shi, and Mirkó Visontai. Seeing through water. In *Advances in Neural Information Processing Systems*, 2005. 1, 2

[15] 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*, 2014. 4

[16] David L. Fried. Probability of getting a lucky short-exposure image through turbulence. *JOSA*, 68(12):1651–1658, 1978. 1

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014. 3

[18] Alain Hore and Djemel Ziou. Image quality metrics: PSNR vs. SSIM. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, 2010. 7

[19] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. In *Techniques and Applications of Image Understanding*. International Society for Optics and Photonics, 1981. 2

[20] Rui Huang, Shu Zhang, Tianyu Li, and Ran He. Beyond face rotation: Global and local perception GAN for photorealistic and identity preserving frontal view synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 3

[21] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of PSNR in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008. 7

[22] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, pages 448–456, 2015. 5

[23] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 5, 6, 7, 8

[24] Jerin Geo James, Pranay Agrawal, and Ajit Rajwade. Restoration of non-rigidly distorted underwater images using a combination of compressive sensing and local polynomial image representations. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 6, 7, 8

[25] Angjoo Kanazawa, David W Jacobs, and Manmohan Chandraker. WarpNet: Weakly supervised matching for single-view reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[26] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. DeblurGAN: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3

[27] Iosif M. Levin, Victor V. Savchenko, and Vladimir Ju. Osadchy. Correction of an image distorted by a wavy water surface: laboratory experiment. *Applied Optics*, 47(35):6650–6655, 2008. 2

[28] Zhengqin Li, Zak Murez, David Kriegman, Ravi Ramamoorthi, and Manmohan Chandraker. Learning to see through tur-

bulent water. In *Proceedings of IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018. 1, 2, 3, 6, 8

[29] Peter Muller and Andreas Savakis. Flowdometry: An optical flow and deep learning based approach to visual odometry. In *Proceedings of IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017. 2

[30] Omar Oreifej, Guang Shu, Teresa Pace, and Mubarak Shah. A two-stage reconstruction approach for seeing through water. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011. 1, 2, 6, 7

[31] R Shefer, M Malhi, and A Shenhar. Waves distortion correction using cross correlation. *Tech. Report, Israel Institute of Technology*, 2001. 1

[32] 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. *Advances in Neural Information Processing Systems*, pages 802–810, 2015. 4

[33] Simron Thapa, Nianyi Li, and Jinwei Ye. Dynamic fluid surface reconstruction unsing deep neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 6

[34] Yuandong Tian and Srinivasa G. Narasimhan. Seeing through water: Image restoration using model-based tracking. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009. 1, 2, 3, 6, 7

[35] Yuandong Tian and Srinivasa G. Narasimhan. A globally optimal data-driven approach for image distortion estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 2

[36] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning GAN for pose-invariant face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

[37] Weiyue Wang, Qiangui Huang, Suya You, Chao Yang, and Ulrich Neumann. Shape inpainting using 3D generative adversarial network and recurrent convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 3

[38] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. EsrGAN: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 3

[39] Zhiying Wen, Donald Fraser, Andrew Lambert, and Hongdong Li. Reconstruction of underwater image by bispectrum. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 2007. 2

[40] You Xie, Erik Franz, Mengyu Chu, and Nils Thuerey. TempoGAN: A temporally coherent, volumetric GAN for super-resolution fluid flow. *ACM Transactions on Graphics (TOG)*, 37(4):1–15, 2018. 3

[41] Tianfan Xue, Michael Rubinstein, Neal Wadhwa, Anat Levin, Fredo Durand, and William T. Freeman. Refraction wiggles for measuring fluid depth and velocity from video. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. 2

[42] Jianwei Yang, Anitha Kannan, Dhruv Batra, and Devi Parikh. Lr-GAN: Layered recursive generative adversarial networks for image generation. *arXiv preprint arXiv:1703.01560*, 2017. 3

[43] Qingsong Yang, Pingkun Yan, Yanbo Zhang, Hengyong Yu, Yongyi Shi, Xuanqin Mou, Mannudeep K Kalra, Yi Zhang, Ling Sun, and Ge Wang. Low-dose ct image denoising using a generative adversarial network with wasserstein distance and perceptual loss. *IEEE Transactions on Medical Imaging*, 37(6):1348–1357, 2018. 3

[44] Weidong Yin, Yanwei Fu, Leonid Sigal, and Xiangyang Xue. Semi-latent GAN: Learning to generate and modify facial images from attributes. *arXiv preprint arXiv:1704.02166*, 2017. 3

[45] Chenyu You, Guang Li, Yi Zhang, Xiaoliu Zhang, Hongming Shan, Mengzhou Li, Shenghong Ju, Zhen Zhao, Zhuiyang Zhang, Wenxiang Cong, et al. Ct super-resolution GAN constrained by the identical, residual, and cycle learning ensemble (GAN-circle). *IEEE Transactions on Medical Imaging*, 39(1):188–203, 2019. 3

[46] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Wei Liu, and Hongdong Li. Adversarial spatio-temporal learning for video deblurring. *IEEE Transactions on Image Processing*, 28(1):291–301, 2018. 3

[47] Lei Zhang, Paul Bao, and Quan Pan. Threshold analysis in wavelet-based denoising. *Electronics Letters*, 37(24):1485–1486, 2001. 3