Unsupervised Non-Rigid Image Distortion Removal via Grid Deformation

Nianyi Li[1,3], Simron Thapa[1], Cameron Whyte[2], Albert Reed[2], Suren Jayasuriya[2], and Jinwei Ye[1]

[1]Louisiana State University, Baton Rouge, LA 70803, USA
[2]Arizona State University, Tempe, AZ 85281, USA
[3]Clemson University, Clemson, SC 29634, USA

Abstract

*Many computer vision problems face difficulties when imaging through turbulent refractive media (*e.g.*, air and water) due to the refraction and scattering of light. These effects cause geometric distortion that requires either handcrafted physical priors or supervised learning methods to remove. In this paper, we present a novel unsupervised network to recover the latent distortion-free image. The key idea is to model non-rigid distortions as deformable grids. Our network consists of a grid deformer that estimates the distortion field and an image generator that outputs the distortion-free image. By leveraging the positional encoding operator, we can simplify the network structure while maintaining fine spatial details in the recovered images. Our method doesn't need to be trained on labeled data and has good transferability across various turbulent image datasets with different types of distortions. Extensive experiments on both simulated and real-captured turbulent images demonstrate that our method can remove both air and water distortions without much customization.*

1. Introduction

Imaging through turbulent refractive medium (*e.g.*, hot air, in-homogeneous gas, fluid flow) is challenging, since the non-linear light transport through the medium (*e.g.*, refraction and scattering) causes non-rigid distortions in perceived images. However, most computer vision algorithms rely on sharp and distortion-free images to achieve the expected performance. Removal of these non-rigid image distortions is therefore critical and beneficial for many vision applications, from segmentation to recognition.

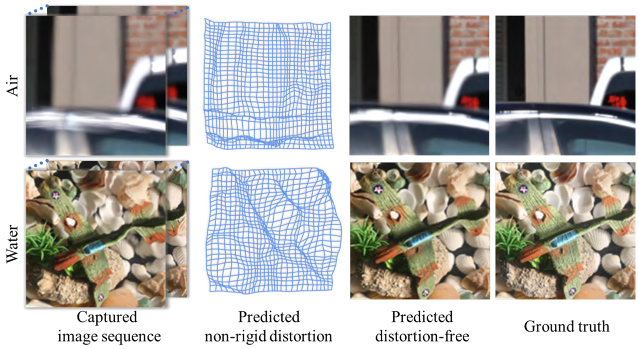


Figure 1. We present a novel unsupervised network to estimate the non-rigid distortion and latent distortion-free image when imaging through turbulent media. Our method works for both air (Row One) and water (Row Two) distortions.

Air turbulence distortion is caused by the constantly changing refractive index field of the air flow. It typically occurs when imaging through long-range atmospheric turbulence or short-range hot air turbulence (*e.g.*, fire flames, vapor streams). Water turbulence distortion, in contrast, is induced by the refraction of light at the water-air interface. Although these two types of distortions share certain visual similarities, they are fundamentally different as they are induced by different physical mechanisms. Air and water turbulent images are usually enhanced in different ways. For air turbulence, physics-based approaches use complex turbulence models (*e.g.*, the Kolmogorov model [22, 23]) to simulate the perturbation, and then restore clear images by inverting the models. For water turbulence, classical methods model the distortion as a function of the water surface height or normal by applying Snell's law [46, 57]. Recently, several learning-based methods are separately proposed to enhance either the air [13, 32] or the water [26] turbulent images. These methods typically require training on a large labeled dataset. Since it is difficult to obtain real turbulent images with ground truth sharp references, these methods use simulated images to augment their datasets and bootstrap the learning.

Motivated by the aforementioned issues, we design an *unsupervised* network that is able to remove non-rigid dis-

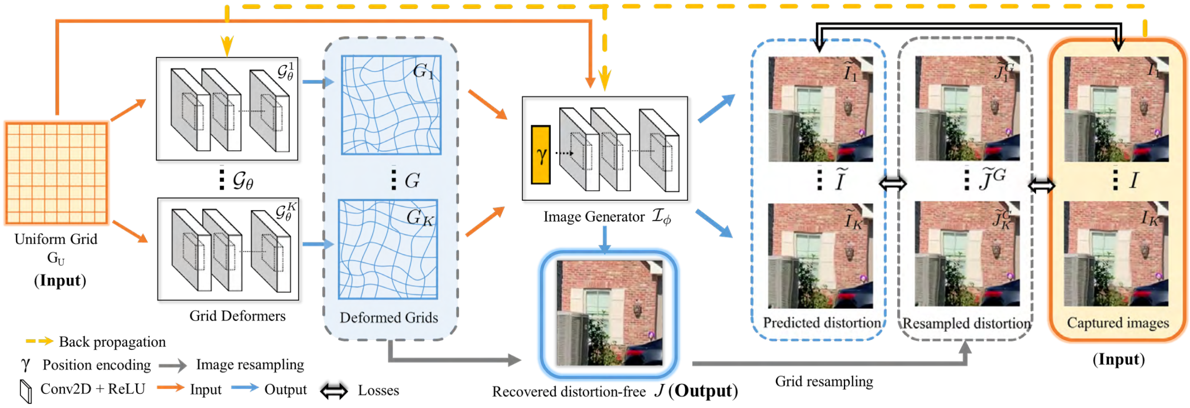


Figure 2. The overall architecture of our unsupervised non-rigid image distortion removal network. The network predicts the distortion-free image J, given a sequence of distorted turbulent image $\{I_k | k = 1, 2, ...K\}$ and uniform grid G_U. $\widetilde{I}$ and $\widetilde{J}^G$ are two intermediate results to constrain the optimization procedure. We use the pair-wise differences among I, $\widetilde{I}$, and $\widetilde{J}^G$ as the optimization losses.

tortions from both air and water turbulent images, as shown in Fig. 1. The key idea is to model the non-rigid distortions as a deformable grids. For example, we model the distortion-free image as a straight and uniform grid, and turbulent images with distorted grids. Inspired by recent works on the Neural Radiance Field (NeRF) [30, 44], we generate the distortion-free image using a grid-based rendering network. Our method, therefore, bypasses sophisticated and heterogeneous physical turbulence models and is able to restore images with different types of distortions.

The overall structure of our network is illustrated in Fig. 2. Our network consists of two main components: a grid deformer $\mathcal{G}$ that estimates the grid deformation and an image generator $\mathcal{I}$ that renders a color image that matches the distortion of an input grid. One critical component in our network is the position encoding operator commonly used in NeRF networks [6, 10, 40, 44, 58]. By incorporating this operator into the image generator, we can simplify our network structure while maintaining fine spatial details in the reconstructed output images.

Our network works as an optimizer for generating the distortion-free image by minimizing pairwise differences between the captured input images, the network's predicted distorted images, and resampled distorted images from the distortion-free image. Our network is fully unsupervised and optimized from scratch on each new example. It does not require any diverse training set to learn from, but running gradient descent on the input data. Specifically, our network is optimized in two steps: we first initialize our network parameters by exploiting the locally-centered property [35] of pixel displacement caused by turbulent media; we then iteratively update the estimated distortion-free image J by minimizing our objective function. Empirically, this two-step optimization converges within 2000 iterations (Adam steps), which takes around 65-499s, depending on the number of input frames. Our initialization provides a reasonable estimation that largely reduces the search space.

We perform extensive experiments on both simulated and real-captured air and water turbulent images. We compare our method with the state-of-the-art methods that are specific to either the air or the water turbulence. We show that our method has better performance in correcting the geometric distortions for both types of turbulence. We summarize our contributions as follows:

- Our network jointly estimates the non-rigid distortions and recovers the latent distortion-free image. It works for both air and water distortions without much customization. It is fully *unsupervised* and does not need to be trained on a labeled dataset.
- Our network leverages the position encoding operator, such that even with fewer numbers of convolutional layers and trainable parameters, it can still generate high-quality images that preserve fine details.
- We propose a two-step optimization framework to guide the training of the unconstrained non-rigid distortion restoration model.
- Extensive experiments demonstrate that state-of-the-art performance can be achieved when applying the proposed grid-based rendering method on two inherently different tasks: atmospheric turbulence removal and imaging through water distortions. Our code is available at: `github.com/Nianyi-Li/unsupervised-NDIR`

2. Related Work

Atmospheric turbulence removal. To resolve the distortion and blur introduced by air turbulence, conventional turbulence restoration methods leverage optical flow [3, 31, 43], lucky regions fusion [50, 41, 12, 21] and blind deconvolution [14, 59] to recover images. Methods employing image registration with deformation estimation architecture can also resolve small movements of the camera and temporal variations due to atmospheric refraction [59, 18]. In a similar turbulence removal problem (not atmospheric), Xue *et al*. [54] adapt classical optical flow to estimate small refractive distortions caused by hot air or gas.

However, many of these methods have artifacts when reconstructing dynamic scenes with large amounts of motion. To counter this, methods have been introduced such as block matching [20], enforcing temporal consistency [33], using reference frames [7], and segmenting static background from moving objects [35, 17, 1]. One promising avenue of direction has been utilizing the physics of turbulence to create accurate forward models for image formation. Mao *et al*. [29] achieve state-of-the-art performance by utilizing knowledge of atmospheric turbulence to create a physics-constrained prior for optimization.

In addition to classical methods, there have been some deep neural networks for air turbulence removal proposed. These are typically convolutional neural networks trained with synthetic or semi-synthetic turbulent data [13, 32, 4]. However, these supervised architectures have trouble with generalization outside of the training data (as do most supervised neural networks). In contrast, our neural network operates in an unsupervised fashion and does not require training data.

Imaging through turbulent water. Recovering undistorted images from underwater images has been well-studied in computer vision for various applications. Early solutions [11, 24] take the mean/median of a distorted image sequence to approximate the latent distortion-free image, although these methods are limited for large distortions. Like in the case of air turbulence, "lucky region" algorithms have also been proposed using clustering [8, 9], manifold embedding [11], and Fourier-based averaging [52]. The seminal work of [46, 47] presents a model-based tracking method to restore underwater images.

Recent advances in imaging through water distortions have leveraged deep learning for state-of-the-art performance. Li *et al*. [26] propose a generative adversarial network (GAN) to correct refractive distortions using a single image. The main drawback of Li's method was that it did not leverage the temporal consistent nature of the fluid flow. Thapa *et al*. [45] propose a two-step dynamic fluid surface reconstruction network to recover the depth and normal maps of the transparent fluid given a short sequence (3 frames) of distorted fluid images.

Unsupervised learning for image restoration. Recently, unsupervised or self-supervised learning using deep image priors [49] for image restoration tasks has enabled improved performance without the need for training data. In [49], the authors showed that a randomly-initialized neural network can be used as a handcrafted prior with excellent results in standard inverse problems such as denoising, super-resolution, and inpainting. Deep image priors have been adopted across many application domains [27, 16, 39, 51].

Recently, analysis-by-synthesis techniques have demonstrated impressive capabilities for estimating visual information, particularly for inverse graphics problems [28, 25, 34, 55, 15, 2, 48]. Mildenhall *et al*. [30] demonstrate how a multilayer perceptron (MLP) coupled with a special layer known as Fourier features [44] can estimate the 5D radiance field of a scene. More recently, [5, 6, 10, 40, 58] exploit the NeRF architecture to solve problems like view synthesis, texture completion from impartial 3D data, non-line-of-sight imaging recognition, etc. In our paper, we leverage Fourier features operator to help perform analysis-by-synthesis for our deformed images.

3. Non-Rigid Distortion Removal Network

Our problem formulation is as follows: we assume a static scene being imaged by a camera with non-rigid distortion being induced by turbulence. Given a sequence of captured non-rigidly distorted images $\{I_k | k = 1, 2, ...K\}$ and a uniform grid G_U, our goal is to recover the latent distortion-free image J as if it was unaffected by the turbulent medium.

Our key idea is to model the non-rigid distortions through grid deformation and reconstruct the distortion-free image J while estimating the distorted image sequence to be consistent with the captured data. To do so, we utilize two sub-networks in our main neural network architecture: a **grid deformer** and an **image generator**. The grid deformer $\mathcal{G}_\theta^k$ is a network to deform a uniform sampled straight grid G_U by estimating the distortion field of the captured frames I_k, and generates a deformed grid $G_k = \mathcal{G}_\theta^k(G_U)$. The image generator is a neural network acting as a parametric function $\widetilde{I} = \mathcal{I}_\phi(G)$ that maps a grid G to an image $\widetilde{I}$. When the grid G_k from the grid deformer is used as input, $\mathcal{I}_\phi$ maps its parameters ϕ to a distorted color image $\widetilde{I}_k$, which is compared to the corresponding image frame I_k. At the same time, feeding a uniform grid G_U to the network $\mathcal{I}_\phi$, we can expect $\mathcal{I}_\phi$ map ϕ to a distortion-free image J, as shown in Fig. 2. We also use the predicted distorted grids $\{G_1, ..., G_K\}$ to directly resample J and obtain another set of distorted images $\{\widetilde{J}_1^G, ..., \widetilde{J}_K^G\}$ as intermediate results to constrain the optimization procedure.

Novel to our method is its *unsupervised* learning approach, which means that our network does not require ground truth knowledge of the underlying true distortion-free image J_{true}. Instead, given an image scene, our network works as an optimizer that solves for J by minimizing the pair-wise differences among I, $\widetilde{I}$, and $\widetilde{J}^G$. To properly estimate sharp image details in the image generator, we leverage the latest positional encoding technique in [30, 44] to preserve fine-details in our recovered latent image, without the need for extra convolutional layers with many parameters, which we describe in Section 3.2. To improve the convergence of our network, especially important for learning in an unsupervised fashion, we introduce a novel two-step optimization algorithm to constrain our network described in Section 3.3.

3.1. Network structure

The overall structure of our non-rigid distortion removal network is shown in Fig. 2. Our network has two main components: the grid deformer and the image generator.

Grid deformer $\mathcal{G}_\theta$ takes a uniform grid $G_U \in \mathbb{R}^{2\times H\times W}$ as input, where W and H are the sampling number along $x-$ and $y-$axis, and outputs a deformed grid $G_k \in \mathbb{R}^{2\times H\times W}$ corresponding to the distortion field of the distorted image $I_k \in \mathbb{R}^{3\times H\times W}$, *i.e.* $G_k = \mathcal{G}^k_\theta(G_U)$, where θ is the set of trainable network parameters. $\mathcal{G}_\theta$ comprises four convolution layers, each has 256 channels and ReLU rectifier. To meet the range constraint for G_k, a tangent hyperbolic function is applied to the output layer. Note that, we train a separate $\mathcal{G}^k_\theta$ for each I_k for two reasons. First, the turbulence field, especially for the air turbulence, is random and has less temporal consistency when the image sequence or video is captured under a standard frame rate, *i.e.*, 30 fps [42, 26, 46, 36]. Using a single network to predict all these random distortion fields is challenging without empirical guidance from ground truth labels and strong temporal consistency constraints. Secondly, the network structure of $\mathcal{G}_\theta$ is simple and has few parameters, and thus we can jointly optimize $\{\mathcal{G}^k_\theta | k = 1, \dots K\}$ with low memory consumption for GPU implementation. Please find a more detailed discussion in Section 4.4.

Image generator $\mathcal{I}_\phi$ renders a color image $\widetilde{I} \in \mathbb{R}^{3\times H\times W}$ when given a grid input $G \in \{G_1, \dots G_k, G_U\}$: $\widetilde{I} = \mathcal{I}_\phi(G)$. If the input grid is a deformed grid G_k, $\mathcal{I}_\phi$ returns an image $\widetilde{I}_k$ that matches the distortion of G_k. If the input grid is a uniform grid G_U, we consider the output as a distortion-free image $J \in \mathbb{R}^{3\times H\times W}$. $\mathcal{I}_\phi$ share a similar network architecture with $\mathcal{G}_\theta$. Since the output of $\mathcal{I}_\phi$ is a color image, we apply a nonlinear Sigmoid activation function to the output layer. Please find more details about the structure of $\mathcal{G}_\theta$ and $\mathcal{I}_\phi$ in our supplementary material.

3.2. Position encoding via Fourier features

As pointed out by [38], networks which directly map xy coordinates to values typically are biased to learn lower frequency functions. To preserve high frequency content in the image, a good solution is to map the grid inputs to a higher dimensional space using high frequency functions before passing them to the network [44, 30]. In our work, we utilize Gaussian random Fourier features (GRFF) to transform the input grid to its high frequency Fourier feature domain before passing it to the image generator $\mathcal{I}_\phi$. Let $\mathbf{v} = (x, y)$ be a coordinate from the input grid. Its GRFF is computed as $\gamma(\mathbf{v}) = [\cos(2\pi\kappa\mathbf{B}\mathbf{v}), \sin(2\pi\kappa\mathbf{B}\mathbf{v})]$, where cos and sin are performed element-wise, κ is a bandwidth-related scale factor, and $\mathbf{B} \in \mathbb{R}^{128\times 2}$ is randomly sampled from a Gaussian distribution $\mathcal{N}(0, 1)$. Thus, the input grid $G \in \mathbb{R}^{2\times H\times W}$ will be mapped into Fourier Feature space $\gamma(\mathbf{v}) \in \mathbb{R}^{256\times H\times W}$.

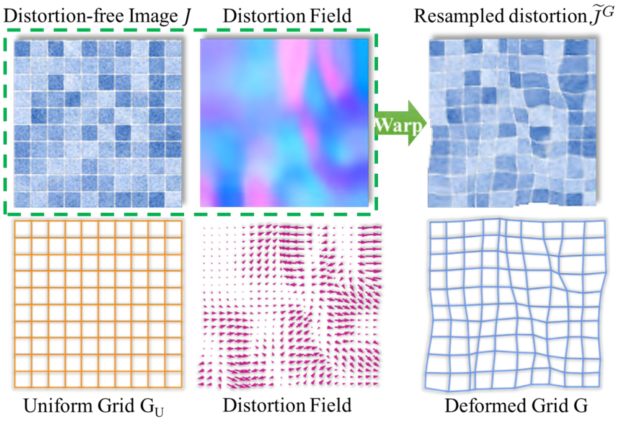


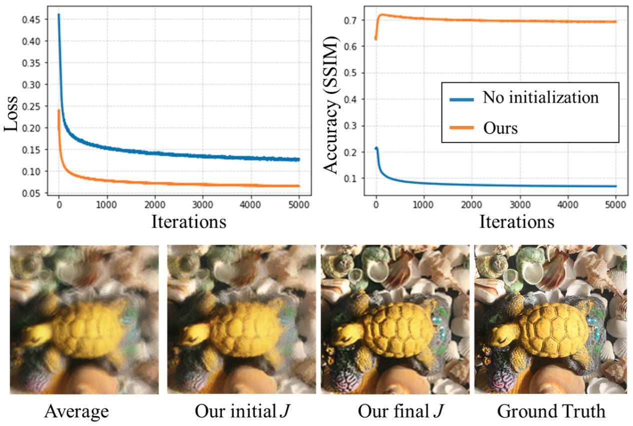
Figure 3. Distorted image generation via grid deformation.

It is worth noting that the choice of κ in the image generator is pertinent to our network's performance. In general, large κ tends to have the network converge fast and very likely to end up at a local minimum. In this paper, we empirically pick $\kappa = 8$. We discuss the effect of GRFF in an ablative study in Section 4.4.

3.3. Two-step network optimization

As our network is unsupervised, it is highly non-convex and has enormous parameter search space. By exploiting redundant information within the deformed image sequence, we propose a two-step network optimization strategy to train a CNN at test time for a given sequence. We first initialize the parameters of $\mathcal{G}_\theta$ and $\mathcal{I}_\phi$ so that they are constrained under properties of non-rigid distortion through turbulent media. Next, we iteratively refine the initialized networks and update the estimated underlying distortion-free image using the captured input distorted images as references.

Parameter initialization. To avoid being trapped in potential saddle points and to allow faster convergence speed, we initialize the network parameters θ and ϕ by exploiting a physical property of pixel displacement caused by turbulent media: *the non-rigid distortions induced by a turbulent medium are generally locally centered* [35]. The distorted images therefore still preserve a large amount of low-frequency image structures. By extrapolating the similarities among the distorted images, we are able to remove a certain amount of non-rigid distortions and obtain a reason-



Average Our initial J Our final J Ground Truth

Figure 4. The loss and accuracy comparison with and without the initialization step (Top row). Our initialization algorithm improves our prediction performance significantly and can initialize sharper distortion-free images than the simple averaging (Second row).

able initial estimation of the distortion-free image.

The grid deformer is initialized by constraining its output to be close to the uniform grid. In this way, we can limit the grid deformation within a certain range and also preserve the order of pixels. We initialize the image generator by constraining its output to have a similar appearance as the input sequence. Specifically, we feed the uniform grid G_U to the image generator. We then compare the output image $J = \mathcal{I}_\phi(\gamma(G_U))$ with all images in $\{I_k\}$, and minimize the sum of per-pixel color differences.

We formulate the initialization procedure as:

$$\min_{\theta,\phi} \sum_k |\mathcal{G}_\theta^k(G_U) - G_U| + |\mathcal{I}_\phi(\gamma(G_U)) - I_k|, \tag{1}$$

where $|\cdot|$ represents the absolute differences (*i.e.*, the L_1 loss). Notice that we use the L_1 loss for all loss functions as it tends to be less affected by outliers. We run the optimization for a few hundreds of iterations, and use the resulting parameters θ' and ϕ' as the initialized weights.

As illustrated in Fig. 4, removing the initialization step will lead the network to converge to a wrong local minimum and fails to predict a reasonable J. In addition, our initialization produces a sharper image that is closer to the latent distortion-free image in color space than simply averaging the images together. This is because taking the average will result in the centroid of the images in RGB color space, and will be blurry since turbulence is time-varying. We discuss more in Section 4.4.

Iterative refinement. After our initialization step, we set out to learn the underlying distortion-free image through the following optimization model:

$$\min_{\theta,\phi} \sum_k |\widetilde{I}_k - I_k| + R(I_k), s.t.\ \theta^0 = \theta',\ \phi^0 = \phi', \tag{2}$$

where $\widetilde{I}_k = \mathcal{I}_\phi(\gamma(G_k))$ is the estimated distorted image, $G_k = \mathcal{G}_\theta^k(G_U)$ is the deformed grid, $R(I_k)$ is a regularizer, θ^0 and ϕ^0 are the initial weights of the network. We use $R(I_k)$ to strengthen the interconnection between the predicted distortion-free image $J = \mathcal{I}_\phi(\gamma(G_U))$ and deformed grids $\{G_k\}$:

$$R(I_k) = |\widetilde{J}_k^G - I_k| + |\widetilde{J}_k^G - \widetilde{I}_k|, \tag{3}$$

where $\widetilde{J}_k^G$ is a resampled distorted image by grid sampling the deformed grid G_k on the recovered latent image J, as shown in Fig. 3. We iteratively update the J using Eqn. 2 until networks converge.

4. Experiments

In this section, we first compare our approach to a set of state-of-the-art methods from the literature on the task of image restoration for both air and fluid turbulence. Then, we present our experimental results to validate our neural network architecture and optimization algorithm. We demonstrate that our method not only outperforms the unsupervised approaches, but even edges out other supervised algorithms that, in contrast to ours, have access to a large amount of synthetic turbulent data using sophisticated physics-based simulators at training time. For quantitative evaluation, we employ the most common metrics for image restoration, *i.e.*, the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM).

4.1. Experimental setup

Implementation details. Our network was implemented in Pytorch [37] with a desktop computer equipped with two NVIDIA GTX 1080 GPUs. Unless specially stated, the experiments follow the same setting: We use the Adam optimizer and set the learning rate as 10^{-4} for both $\mathcal{G}_\theta$ and $\mathcal{I}_\phi$. We use 1,000 iterations for parameter initialization, and in the iterative refinement stage, our network converges within 1,000 iterations, as shown in Fig. 4. We empirically pick $\kappa = 8$ as the bandwidth-related factor of the Fourier feature mapping operator for all experiments.

Memory consumption. The overall network to handle 10 input frames has around 1.53 million trainable parameters, which include 1.33 million (M) total for the grid deformers (one for each frame) and 0.2 M for the image generator. Compared to a contemporary GAN to restore imaging through water turbulence with about 50 million parameters [26], our network restores comparable high-frequency details in the predicted image with less memory footprint.

4.2. Evaluation on air turbulence

For the air turbulence, we compare with the following state-of-the-art methods: CLEAR [1], Oreifej *et al.* [35], Zhang *et al.* [56], Gao *et al.* [13], and Mao *et al.* [29].

[1, 56, 29] are physics-based approaches that use complex turbulence models. [13] is a supervised method trained on a large semi-synthetic turbulence dataset.

We compare the image restoration performance on both real and synthetic datasets. For synthetic experiments, we synthesize turbulent image sequences with different turbulence strengths. We use the turbulence strength parameter $C_n^2 = 1 \times 10^{-14}$ for the weak turbulence; $C_n^2 = 1 \times 10^{-13}$ for the medium; and $C_n^2 = 1 \times 10^{-12}$ for the strong. More details on the simulation parameters can be found in our supplementary material. The quantitative comparison results with respect to various turbulence levels are reported in Table 1. We can see that our method is robust for the strong turbulence.

Strength	Metrics	Average	Our init.	[1]	Ours
Weak	PSNR↑	25.10	**25.20**	18.31	24.29
	SSIM↑	0.941	0.95	0.856	**0.984**
Medium	PSNR↑	19.48	19.85	14.09	**20.70**
	SSIM↑	0.774	0.804	0.561	**0.904**
Strong	PSNR↑	17.08	17.12	12.51	**17.40**
	SSIM↑	0.632	0.667	0.433	**0.799**

Table 1. Quantitative comparison on air turbulence data with various strengths. We compare with the temporal average frame, our initial J and CLEAR [1].

As for the real data, we compare on two types of air-turbulence phenomena: hot-air turbulence and long-range atmospheric turbulence. For the former, we capture our data by using a gas stove to heat the air. We use a cellphone camera to capture 5 scenes around 50 meters away from the heat source. For the latter, we use data from two sources: (1) the widely adopted *Chimney* and *Building* sequences [19] and (2) our own turbulent images captured using a Nikon Coolpix P1000 camera. We mount the camera on a tripod to capture 1080p videos at 30 fps of 5 scenes at around 1-3 miles away with 125× optical zoom.

We show the comparisons with the state-of-the-arts on *Chimney* and *Building* in Fig. 5. As we don't have access to the codes of several methods [1, 56, 13], we directly take the images from their original papers. It is important to note that most of these algorithms take a longer input sequence ($\geq$ 100 frames) and has deblurring component to produce sharper images. In contrast, our network only needs 10 input frames to make a reliable prediction. As our network focuses more on distortion removal, our output may still suffer from certain amount of blurriness. We can apply off-the-shelf deblurring algorithm to further sharpen our results. Specifically, we use Xu *et al.* [53] for image deblurring. The post-deblurring results are shown as "Ours + Deblur" in Figs. 5 and 6.

We show the qualitative comparison on our real captured data in Fig. 6. Here we only compare to the methods that we have access to the codes or the authors provided us the results. Please see our supplementary material for video results on these sequences.

4.3. Evaluation on water turbulence

For the water turbulence, we compare our methods with the following state-of-the-arts: Tian *et al.* [46] and Oreifej *et al.* [36] are physics-based method. Li *et al.* [26] is a learning-based method. All provided the source codes.

We perform experiments on two water turbulent image datasets: [45] and [26]. Thapa *et al.* [45] proposed a synthetic dataset providing both the distorted image sequences and the ground truth pattern. The images are simulated using a physics-based ray tracer with different types of waves. [26] is a real captured dataset. It poses challenges such as illumination change and shadows. The distortions are also more drastic. We show the visual comparison results in Fig. 7. We can see that our method outperforms all the states-of-the-arts. To further validate the robustness, we created three synthetic sequences of water turbulence images, each contains 10 frames, caused by different types of waves using the physics-based ray tracer provided by [45]. The ocean waves are the most challenging, as they are more random and have more high-frequency turbulence components. As shown in Table 2, our method ranks higher on the Ripple and Ocean waves. Although [26] achieves higher PSNR/SSIM scores on the Gaussian wave, their results appear blurrier than ours (see visual comparisons in the supplementary material). Further, [26] requires training on ~320K images.

Types	Metrics	[46]	[36]	[26]	Ours
Ripple	PSNR↑	20.40	21.24	20.70	**23.63**
	SSIM↑	0.878	0.902	0.882	**0.970**
Ocean	PSNR↑	20.93	21.13	21.32	**22.32**
	SSIM↑	0.891	0.901	0.833	**0.964**
Gaussian	PSNR↑	17.61	17.40	**18.67**	17.50
	SSIM↑	0.787	0.789	**0.833**	0.818

Table 2. Quantitative comparison on different types of water turbulence. We compare our result with Tian *et al.* [46], Oreifej *et al.* [36], and Li *et al.* [26].

4.4. Ablation studies

We conducted a set of ablation studies to validate various design choices in our network architecture. For all these studies, we tested image restoration through simulated air turbulence, as the resulting non-rigid distortions are more random than water turbulence in general. We utilize a physics-based atmospheric turbulence simulator for 2D images [42] to generate 100 different turbulence fields with controllable turbulence strength C_n^2 that are applied to a clear image to generate distorted image sequences.

Network structures of $\mathcal{G}_\theta$. For a fair comparison of the capability of different structures in encoding the deformed

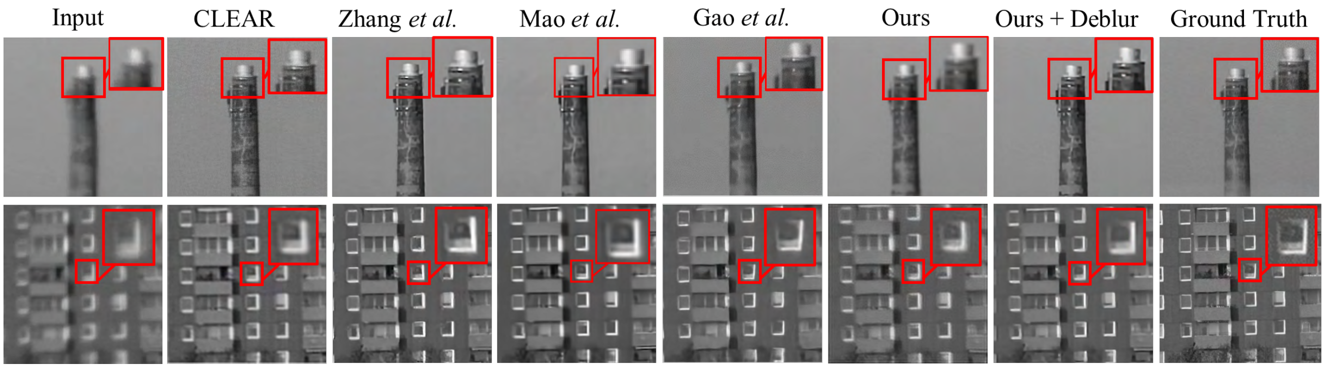


Figure 5. Comparisons on the *Building* and the *Chimney*. It's worth noting that all methods take the full sequence (100 frames), while our method only takes 10 randomly picked frames.

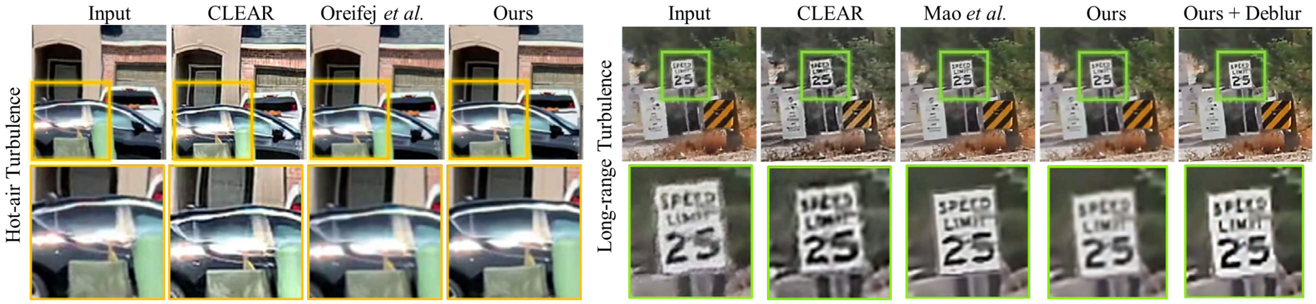


Figure 6. Visual comparison results on our real captured hot-air turbulence and long-range atmospheric turbulence images.

grid, we replace $\mathcal{G}_\theta^k$ with several different CNNs, as shown in Table 3. Specifically, Con_2, Con_4 and Con_6 are CNN structure with 2, 4, 6 convolutional layers respectively. As we have 10 frames in the input sequence, the total number of parameters is equal to 10 × the size of each $\mathcal{G}_\theta^k$. We also compare with the architecture that simply use a deep Autoencoder CNN (DAE) with skip connections [39] to predict 10 deformed grids $\{G_k\}$ at once. We demonstrate that the proposed structure (Con_4) is superior to other networks w.r.t. the restoration ability with fewer trainable parameters.

$\mathcal{G}_\theta$	10 subnets			1 network
	$Conv_2$	$Conv_4$	$Conv_6$	DAE
Total params	0.02M	1.33M	2.65M	2.35M
PSNR↑	19.23	**20.48**	20.06	16.83
SSIM↑	0.775	**0.790**	0.742	0.467

Table 3. Comparison of the performance on the restoration ability among different network structures of $\mathcal{G}_\theta$.

Number of input images. One critical design consideration for our network is the number of input images needed to generate a distortion-free image. There is a trade-off between the speed of the network in restoring images versus the visual fidelity. In Table 4, we show the average PSNR/SSIM and total running time (2,000 iterations) for 2, 5, 10, 15 and 20 frames. Please find the visual comparison results in our supplementary materials. Increasing the input number does benefit our restoration task, but we sacrifice time efficiency in order to do so. Since there are diminishing returns to the image quality of our predicted sharp images after 10 input frames, this number is chosen as the default input number throughout the following experiments.

# of inputs	2	5	10	15	20
PSNR↑	17.55	18.50	20.50	**21.43**	21.13
SSIM↑	0.556	0.666	0.793	0.827	**0.830**
Time	65s	143s	265s	403s	499s

Table 4. Average PSNR, SSIM, and running time (2,000 iterations) comparison on taking different numbers of input images.

Effect of position encoding. The Gaussian random Fourier features (GRFFs) encodes the input grid into a higher dimensional space, enabling our image generator to approximate real high-frequency sharp images. We compare our full network with the one that removes the GRFF in the image generator and simply takes $\{G_k\}$ as input. As shown in Fig. 8, with Fourier feature mapping operators in $\mathcal{I}$, we have about 30.9% improvement in SSIM and 13.9% improvement in PSNR of the recovered latent images, compared with the network variant without GRFF (No GRFF). They also help increase the convergence speed. We show the impact of the bandwidth-related scale factor κ in our supplementary material.

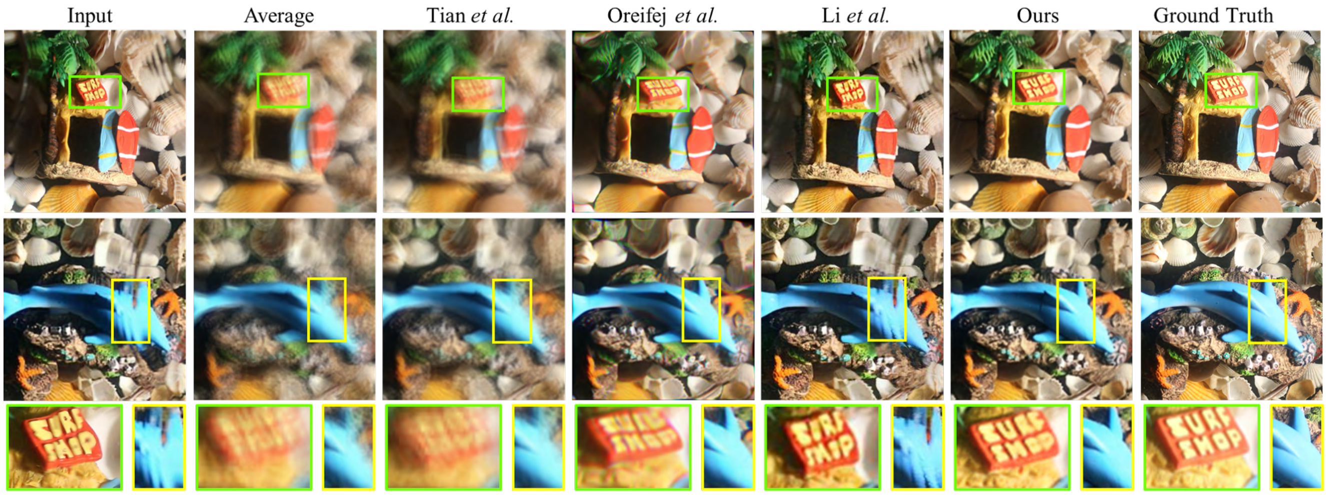


Figure 7. Visual comparisons on real water turbulence images provided by Li *et al.* [26], which proposed a supervised GAN model to restore water turbulence.

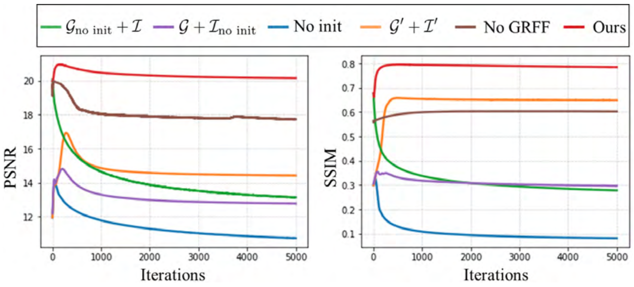


Figure 8. Ablation study on different variants of our proposed network. We show the PSNR and SSIM vs. the number of iteration curves for comparison.

Effect of initialization step. To evaluate the effects of the network initialization, we created four variants of the network for comparison: 1) $\mathcal{G}_{\text{no init}} + \mathcal{I}$, that removes the initialization step of grid deformer $\mathcal{G}$; 2) $\mathcal{G} + \mathcal{I}_{\text{no init}}$, that removes the initialization step of image generator $\mathcal{I}$; 3) *No init*, that has no initialization step at all; and 4) $\mathcal{G}' + \mathcal{I}'$, that adds the initialization losses to the iterative refinement step. As shown in Fig. 8, taking out the initialization step from either the $\mathcal{G}_\theta$ and $\mathcal{I}_\phi$, the overall network has subpar optimization performance and fails to predict a reasonably sharp image. However, simply adding the initialization losses to the main optimization loop can degrade our restoration performance, as these losses can lead the network to converge to some local minimum, as discussed in Section. 3.3.

Effect of $R(I_k)$. The term $R(I_k)$ in Eq. 2 is used for regulating the resampled distortion images $\tilde{J}^G$. It enforces the grid deformer network to output a warp motion that could be used to generate a plausible distortion-free image by both direct resampling and the image generator. We perform ablation experiment by removing the term $R(I_k)$. Qualitative and quantitative comparison results are included in the supplementary material. We can see that the output image is more blurry when $R(I_k)$ is taken out. Quantitatively, with $R(I_k)$ included in the objective function, both the PSNR and SSIM values are apparently improved.

5. Conclusions and Discussions

We have presented an unsupervised non-rigid image distortion removal network via grid-deformation given a short sequence of turbulent images. Our network architecture can jointly estimate the sharp latent image as well as the non-rigid distortion. Our proposed two-step optimization framework can significantly improve the performance of the unconstrained non-rigid distortion restoration model. Our network does not require ground truth turbulence models as guidance, and thus can be generalized to handle most non-rigid distortions, for both air and fluid turbulence.

Limitations and future directions. As our method does not have any physics-based constraints and it takes only 10 frames as input, a good initialization is critical for our algorithm to reach optimizing results. Further, our method does not solve cases where there is large motion in the scene in addition to turbulence distortions. For future directions, we hope to tackle these issues as well as make our method applicable to more general non-rigid distortions.

Acknowledgements. Li, Thapa and Ye are supported by NSF CRII-1948524, Louisiana Board of Regents grant LEQSF(2018-21)-RD-A-10, and a gift from DGene. Whyte and Jayasuriya are supported by NSF IIS-1909192. Reed was supported by a DoD NDSEG Fellowship.

References

[1] Nantheera Anantrasirichai, Alin Achim, and David Bull. Atmospheric turbulence mitigation for sequences with moving objects using recursive image fusion. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2895–2899. IEEE, 2018. 3, 5, 6

[2] Dejan Azinovic, Tzu-Mao Li, Anton Kaplanyan, and Matthias Niessner. Inverse path tracing for joint material and lighting estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2447–2456, 2019. 3

[3] Tufan Çaliskan and Nafiz Arica. Atmospheric turbulence mitigation using optical flow. In *2014 22nd International Conference on Pattern Recognition*, pages 883–888. IEEE, 2014. 2

[4] Gongping Chen, Zhisheng Gao, Qiaolu Wang, and Qingqing Luo. U-net like deep autoencoders for deblurring atmospheric turbulence. *Journal of Electronic Imaging*, 28(5):1 – 14, 2019. 3

[5] Wenzheng Chen, Fangyin Wei, Kiriakos N Kutulakos, Szymon Rusinkiewicz, and Felix Heide. Learned feature embeddings for non-line-of-sight imaging and recognition. *ACM Transactions on Graphics (TOG)*, 2020. 3

[6] Julian Chibane and Gerard Pons-Moll. Implicit feature networks for texture completion from partial 3d data. In *European Conference on Computer Vision*. Springer, 2020. 2, 3

[7] Nicholas Chimitt, Zhiyuan Mao, Guanzhe Hong, and Stanley H Chan. Rethinking atmospheric turbulence mitigation. *arXiv preprint arXiv:1905.07498*, 2019. 3

[8] Arturo Donate, Gary Dahme, and Eraldo Ribeiro. Classification of textures distorted by waterwaves. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 2, pages 421–424. IEEE, 2006. 3

[9] Arturo Donate and Eraldo Ribeiro. Improved reconstruction of images distorted by water waves. In *VISAPP (1)*, pages 228–235, 2006. 3

[10] Emilien Dupont, Miguel Bautista Martin, Alex Colburn, Aditya Sankar, Josh Susskind, and Qi Shan. Equivariant neural rendering. In *International Conference on Machine Learning*. PMLR, 2020. 2, 3

[11] Alexei Efros, Volkan Isler, Jianbo Shi, and Mirkó Visontai. Seeing through water. In *Advances in Neural Information Processing Systems*, pages 393–400, 2005. 3

[12] David L Fried. Probability of getting a lucky short-exposure image through turbulence. *JOSA*, 68(12):1651–1658, 1978. 2

[13] Jing Gao, Nantheera Anantrasirichai, and David Bull. Atmospheric turbulence removal using convolutional neural network. *arXiv preprint arXiv:1912.11350*, 2019. 1, 3, 5, 6

[14] Jérôme Gilles, Tristan Dagobert, and Carlo De Franchis. Atmospheric turbulence restoration by diffeomorphic image registration and blind deconvolution. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 400–409. Springer, 2008. 2

[15] Ioannis Gkioulekas, Shuang Zhao, Kavita Bala, Todd Zickler, and Anat Levin. Inverse volume rendering with material dictionaries. *ACM Transactions on Graphics (TOG)*, 32(6):162, 2013. 3

[16] Kuang Gong, Ciprian Catana, Jinyi Qi, and Quanzheng Li. Pet image reconstruction using deep image prior. *IEEE transactions on Medical Imaging*, 38(7):1655–1665, 2018. 3

[17] Kalyan Kumar Halder, Murat Tahtali, and Sreenatha G Anavatti. Moving object detection and tracking in videos through turbulent medium. *Journal of Modern Optics*, 63(11):1015–1021, 2016. 3

[18] Renjie He, Zhiyong Wang, Yangyu Fan, and David Fengg. Atmospheric turbulence mitigation based on turbulence extraction. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1442–1446. IEEE, 2016. 3

[19] Michael Hirsch, Suvrit Sra, Bernhard Schölkopf, and Stefan Harmeling. Efficient filter flow for space-variant multiframe blind deconvolution. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 607–614. IEEE, 2010. 6

[20] Claudia S Huebner. Turbulence mitigation of short exposure image data using motion detection and background segmentation. In *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXIII*, volume 8355, page 83550I. International Society for Optics and Photonics, 2012. 3

[21] Sarah John and Mikhail A Vorontsov. Multiframe selective information fusion from robust error estimation theory. *IEEE Transactions on Image Processing*, 14(5):577–584, 2005. 2

[22] Andrey Nikolaevich Kolmogorov. Dissipation of energy in the locally isotropic turbulence. In *Dokl. Akad. Nauk SSSR A*, volume 32, pages 16–18, 1941. 1

[23] Andrey Nikolaevich Kolmogorov. The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers. *Cr Acad. Sci. URSS*, 30:301–305, 1941. 1

[24] 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. 3

[25] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 37(6):222:1–222:11, 2018. 3

[26] 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. 1, 3, 4, 5, 6, 8

[27] Jiaming Liu, Yu Sun, Xiaojian Xu, and Ulugbek S Kamilov. Image restoration using total variation regularized deep image prior. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7715–7719. IEEE, 2019. 3

[28] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7708–7717, 2019. 3

[29] Zhiyuan Mao, Nicholas Chimitt, and Stanley H Chan. Image reconstruction of static and dynamic scenes through anisoplanatic turbulence. *IEEE Transactions on Computational Imaging*, 6:1415–1428, 2020. 3, 5, 6

[30] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2, 3, 4

[31] Robert Nieuwenhuizen, Judith Dijk, and Klamer Schutte. Dynamic turbulence mitigation for long-range imaging in the presence of large moving objects. *EURASIP Journal on Image and Video Processing*, 2019(1):2, 2019. 2

[32] Robert Nieuwenhuizen and Klamer Schutte. Deep learning for software-based turbulence mitigation in long-range imaging. In *Artificial Intelligence and Machine Learning in Defense Applications*, volume 11169, page 111690J. International Society for Optics and Photonics, 2019. 1, 3

[33] Robert PJ Nieuwenhuizen, Adam WM van Eekeren, Judith Dijk, and Klamer Schutte. Dynamic turbulence mitigation with large moving objects. In *Electro-Optical and Infrared Systems: Technology and Applications XIV*, volume 10433, page 104330S. International Society for Optics and Photonics, 2017. 3

[34] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 38(6), Dec. 2019. 3

[35] Omar Oreifej, Xin Li, and Mubarak Shah. Simultaneous video stabilization and moving object detection in turbulence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2):450–462, 2012. 2, 3, 4, 5

[36] Omar Oreifej, Guang Shu, Teresa Pace, and Mubarak Shah. A two-stage reconstruction approach for seeing through water. In *CVPR 2011*, pages 1153–1160. IEEE, 2011. 4, 6

[37] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[38] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019. 4

[39] Dongwei Ren, Kai Zhang, Qilong Wang, Qinghua Hu, and Wangmeng Zuo. Neural blind deconvolution using deep priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3341–3350, 2020. 3, 7

[40] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *European Conference on Computer Vision*, pages 623–640. Springer, 2020. 2, 3

[41] Michael C Roggemann, Craig A Stoudt, and Byron M Welsh. Image-spectrum signal-to-noise-ratio improvements by statistical frame selection for adaptive-optics imaging through atmospheric turbulence. *Optical Engineering*, 33(10):3254–3265, 1994. 2

[42] Armin Schwartzman, Marina Alterman, Rotem Zamir, and Yoav Y Schechner. Turbulence-induced 2d correlated image distortion. In *2017 IEEE International Conference on Computational Photography (ICCP)*, pages 1–13. IEEE, 2017. 4, 6

[43] Masao Shimizu, Shin Yoshimura, Masayuki Tanaka, and Masatoshi Okutomi. Super-resolution from image sequence under influence of hot-air optical turbulence. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 2

[44] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020. 2, 3, 4

[45] Simron Thapa, Nianyi Li, and Jinwei Ye. Dynamic fluid surface reconstruction unsing deep neural network. In *Conference on Computer Vision and Pattern Recognition*, 2020. 3, 6

[46] Yuandong Tian and Srinivasa G Narasimhan. Seeing through water: Image restoration using model-based tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 2303–2310. IEEE, 2009. 1, 3, 4, 6

[47] Yuandong Tian and Srinivasa G Narasimhan. A globally optimal data-driven approach for image distortion estimation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1277–1284. IEEE, 2010. 3

[48] Chia-Yin Tsai, Aswin C. Sankaranarayanan, and Ioannis Gkioulekas. Beyond volumetric albedo — A surface optimization framework for non-line-of-sight imaging. In *IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[49] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018. 3

[50] Mikhail A Vorontsov and Gary W Carhart. Anisoplanatic imaging through turbulent media: image recovery by local information fusion from a set of short-exposure images. *JOSA A*, 18(6):1312–1324, 2001. 2

[51] Oleg Voynov, Alexey Artemov, Vage Egiazarian, Alexander Notchenko, Gleb Bobrovskikh, Evgeny Burnaev, and Denis Zorin. Perceptual deep depth super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5653–5663, 2019. 3

[52] ZY Wen, Donald Fraser, A Lambert, and HD Li. Reconstruction of underwater image by bispectrum. In *2007 IEEE International Conference on Image Processing*, volume 3, pages III–545. IEEE, 2007. 3

[53] Li Xu, Shicheng Zheng, and Jiaya Jia. Unnatural l0 sparse representation for natural image deblurring. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1107–1114, 2013. 6

[54] 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 *European Conference on Computer Vision*, pages 767–782. Springer, 2014. 3

[55] Cheng Zhang, Lifan Wu, Changxi Zheng, Ioannis Gkioulekas, Ravi Ramamoorthi, and Shuang Zhao. A differential theory of radiative transfer. *ACM Transactions on Graphics (TOG)*, 38(6):1–16, 2019. 3

[56] Chao Zhang, Bindang Xue, Fugen Zhou, and Wei Xiong. Removing atmospheric turbulence effects in unified complex steerable pyramid framework. *IEEE Access*, 6:75855–75867, 2018. 5, 6

[57] Mingjie Zhang, Xing Lin, Mohit Gupta, Jinli Suo, and Qionghai Dai. Recovering scene geometry under wavy fluid via distortion and defocus analysis. In *European Conference on Computer Vision*, pages 234–250. Springer, 2014. 1

[58] Xiuming Zhang, Sean Fanello, Yun-Ta Tsai, Tiancheng Sun, Tianfan Xue, Rohit Pandey, Sergio Orts-Escolano, Philip Davidson, Christoph Rhemann, Paul Debevec, et al. Neural light transport for relighting and view synthesis. *ACM Transactions on Graphics (TOG)*, 2021. 2, 3

[59] Xiang Zhu and Peyman Milanfar. Removing atmospheric turbulence via space-invariant deconvolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):157–170, 2012. 2, 3