Physically-Grounded Turbulence Mitigation with Frame-Shared Degradation Parameters

Dongxin Xie[1] Yan Huang[1*] Yong Xu[1,2] Hui Ji[3]

[1]School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China

[2]Guangdong Provincial Key Laboratory of Multimodal Big Data Intelligent Analysis, Guangzhou 510006, China

[3]Department of Mathematics, National University of Singapore, 119076, Singapore

202410190012@mail.scut.edu.cn,aihuangy@scut.edu.cn, yxu@scut.edu.cn, matjh@nus.edu.sg

Abstract

Atmospheric turbulence causes spatially varying distortion and blur in long-range imaging, making restoration highly challenging in real-world applications. Supervised methods rely on synthetic training data, whose simulated degradation often cannot faithfully reflect real turbulence. Existing unsupervised methods usually estimate degradation parameters for each frame independently, without exploiting the shared correlations among frames from the same scene. We propose TMFS, an optimization-based and physics-grounded method for unsupervised turbulence restoration. TMFS is based on a physically motivated tilt-then-blur degradation model and represents frame degradations through a shared turbulence structure. By decomposing the distortion and blur of each frame into a scene-shared correlation function and per-frame noise maps, TMFS enables cross-frame information sharing and alleviates the ill-posedness of framewise estimation. Experiments on synthetic and real datasets demonstrate the effectiveness of TMFS and its strong generalization to real turbulence.

1. Introduction

Atmospheric turbulence severely degrades long-range outdoor imaging. Induced by random fluctuations of the atmospheric refractive index along the propagation path, it causes spatially varying geometric distortion and blur, which substantially degrade image quality and hinder downstream vision tasks such as recognition and tracking. Developing effective turbulence restoration methods is thus important for many practical imaging applications. Existing turbulence restoration methods can be broadly categorized into single-frame and multi-frame settings [6, 17, 32, 33]. Single-frame restoration aims to recover a high-quality image from one degraded observation, whereas multi-frame restoration leverages multiple degraded frames of the same scene. This paper focuses on the multi-frame setting, in which the additional observations provide complementary information that can be exploited for restoration. However, the problem remains highly challenging because atmospheric turbulence leads to a complex coupling of geometric distortion and spatially varying blur, with stochastic variations across frames.

Recent deep learning methods have achieved promising results for turbulence restoration. However, most supervised methods rely on paired training data, which is difficult to acquire for real turbulence-degraded scenes. In addition, due to the complexity of atmospheric imaging, synthetic degradation models often fail to faithfully capture real-world turbulence [16]. This makes unsupervised restoration a compelling alternative, as it does not depend on synthetic supervision. However, the lack of ground-truth clean images also makes the problem substantially more challenging, requiring the method to effectively training the network only with degraded observations. The multi-frame setting is especially important in this context. While restoring a single turbulence-degraded image is highly ill-posed, multiple frames of the same scene provide complementary observations and cross-frame correlations that offer additional information for restoration. This makes it possible to jointly recover the latent image and the degradation process directly from the observed sequence.

Existing unsupervised methods commonly adopt an optimization-based framework with a forward model [18, 20]. In this framework, degraded observations is simulated by applying a turbulence degradation process with a set of degradation parameters on a latent clean image, and the

*Corresponding author: aihuangy@scut.edu.cn

This work was partially supported by the National Key Research and Development Program of China (Grant 2024YFE0105400), the Science and Technology Planning Project of Guangdong Province (Grant 2025A0505020016), the National Natural Science Foundation of China (Grants 62372186 and 62472179), and the Singapore MOE Academic Research Fund (AcRF) Tier 2 (WBS No. A-8004364-00-00).

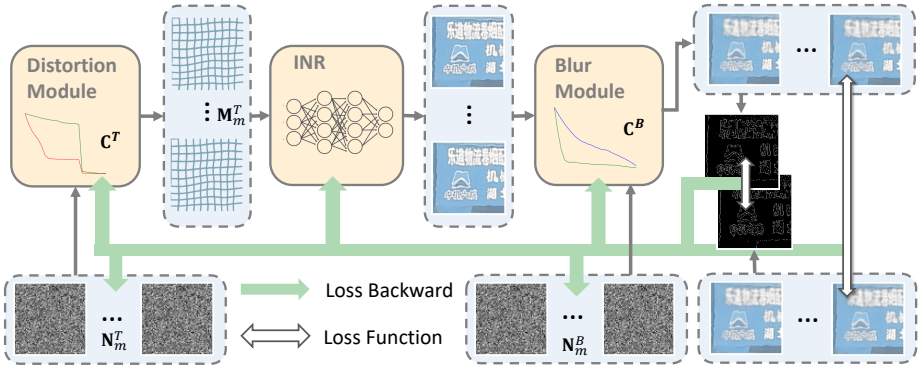


Figure 1. Overview of TMFS. The distortion module (left) predicts a distorted sampling grid from the INR features. The blur module (right) predicts Zernike coefficients to construct the PSF, which is applied to the sharp distorted image. During inference, TMFS produces the restored image using a regular sampling grid.

restoration is done by minimizing the discrepancy between the simulated and observed degraded frames. A well established turbulence degradation is Tilt-then-Blur model [7]:

$$\mathbf{J}_m = \mathcal{B}(\mathbf{I}(\mathbf{G} + \mathbf{M}_m^T), \mathbf{M}_m^B), \tag{1}$$

where $\mathbf{J}_m$ is the m-th degraded image and $\mathbf{G}$ is the uniform image grid. $\mathbf{M}_m^T$ denotes the displacement field induced by tilt, which shifts each grid location in $\mathbf{G}$ to a distorted coordinate. Thus, $\mathbf{G}+\mathbf{M}_m^T$ denotes the warped sampling grid, and $\mathbf{I}(\mathbf{G} + \mathbf{M}_m^T)$ is the distorted image. The operator $\mathcal{B}(\cdot, \mathbf{M}_m^B)$ denotes the blur process parameterized by $\mathbf{M}_m^B$. An unsupervised learning method will then reparametrize the latent image by a coordinate-based image generator $f_\theta(\cdot)$ which leads to: $f_\theta(G) = \mathbf{I}(G)$. f_θ is implemented by an implicit neural representation (INR) with a 4-layers multilayer perceptron (MLP) and position encoding.

Recovering the latent sharp image $\mathbf{I}$ requires jointly estimating the network parameters θ, the distortion fields $\{\mathbf{M}_m^T\}_m$, and the blur parameters $\{\mathbf{M}_m^B\}_m$ from only the degraded frames $\{\mathbf{J}_m\}_m$. In the absence of ground-truth images, this problem is highly ill-posed and requires strong regularization to prevent overfitting. However, existing methods typically estimate $\mathbf{M}_m^T$ and $\mathbf{M}_m^B$ independently for each frame [18, 20], without effectively exploiting the correlations among frames from the same scene.

To overcome this limitation, TMFS introduces a frame-correlated degradation model. Specifically, we decompose the distortion and blur parameters of each frame into scene-shared correlation functions and per-frame noise maps, inspired by random-process-based turbulence simulation [8, 9, 24]. This formulation explicitly models cross-frame dependencies, reduces the ill-posedness of framewise estimation, and yields more physically plausible restoration. Together with additional physics-motivated regularization, it further improves robustness and restoration quality. An overview of TMFS is shown in Fig. 1. The main contributions of this paper are summarized below.

- We propose TMFS, an unsupervised turbulence restoration method that models the distortion and blur of all frames through correlated degradation parameters. This frame-correlated formulation enables the method to exploit cross-frame dependencies for restoration.
- We develop a physics-grounded degradation model together with physically motivated regularization. By including constraints derived from turbulence imaging physics, the proposed method guides the optimization toward more physically sound solutions.
- Extensive experiments on both synthetic and real datasets show that TMFS outperforms existing unsupervised methods and demonstrates substantially stronger robustness on real-world turbulence, where supervised methods trained on synthetic data often fail.

2. Related Work

2.1. Turbulence restoration

Before the deep learning era, conventional turbulence restoration methods [2, 4, 12] mainly relied on blind deconvolution, lucky-region selection (image regions that are less severely blurred by turbulence) and optical flow. Recent advances in deep learning have led to significant progress in turbulence restoration. Supervised methods [19, 22, 23, 32, 33] are trained on paired degraded and clean images and perform well on synthetic benchmarks. However, real paired data are extremely difficult to obtain, so these methods mainly depend on synthetic training data, whose degradation often cannot faithfully reflect real atmospheric turbulence, limiting real-world generalization.

This has motivated increasing interest in unsupervised methods. For example, [29] exploits better reference-frame selection and prior information, while [18, 20] use INR-

based image representations with a differentiable forward model. However, these methods usually estimate degradations independently for each frame, without effectively exploiting cross-frame correlations, and are only weakly constrained by turbulence physics. In contrast, TMFS adopts a physics-grounded degradation model and a frame-correlated parameterization of distortion and blur for unsupervised turbulence restoration.

2.2. Turbulence simulation

Numerous methods exist to simulate the image degradation caused by atmospheric turbulence. While split-step methods [3, 15, 27] provide high fidelity by simulating wave propagation and phase distortion, they are computationally intensive due to multiple Fourier transforms. In [8, 25], wavefront aberrations along the propagation path are represented using Zernike polynomials. Their Zernike coefficients characterize the phase distortion and can be used to derive the corresponding point spread function (PSF). Following [8], the coefficients can be separated into a low-order component that accounts for geometric distortion and a high-order component that accounts for blur.

Zernike polynomials: The second- and third-order Zernike coefficients represent wavefront tilts along the two spatial axes and mainly account for image distortion. For computational efficiency, some methods [8, 33] handle these two coefficients separately. Because turbulence-induced distortion is spatially correlated, the corresponding tilt coefficients are also correlated across pixels. Accordingly, they are modeled in [8] as samples from a spatially correlated random process, whose autocorrelation is determined by turbulence properties such as propagation distance and turbulence strength. The higher-order Zernike coefficients mainly account for blur.

P2S (Phase to Space) for blur kernel: When high-order Zernike coefficients are available at each pixel, directly transforming them into pixel-wise PSFs and applying the resulting spatially varying blur, as in [8], incurs high computational cost. To accelerate this process, P2S [24] constructs a PSF dataset, decomposes the PSFs into a set of spatial bases, and trains a shallow network to map the high-order Zernike coefficients $\mathbf{Z}$ to the coefficients of these PSF bases. This allows the blurred image to be computed by convolving with the PSF bases and then linearly combining the results using the predicted weights.

3. Method

This paper addresses the restoration of a sharp image from multiple blurred and distorted captures of the same static scene taken in a short time interval. As outlined in Eq. (1), restoration requires suitable representations for $\mathbf{M}^T$ and $\mathbf{M}^B$ within an optimizable framework. The overall pipeline is shown in Fig. 1. This section begins with relevant preliminaries, followed by the derivation of $\mathbf{M}^T$ and $\mathbf{M}^B$ shown in Fig. 2. Then it describes the optimization and additional regularization. Once trained per scene, TMFS can restore an image from a uniform grid input.

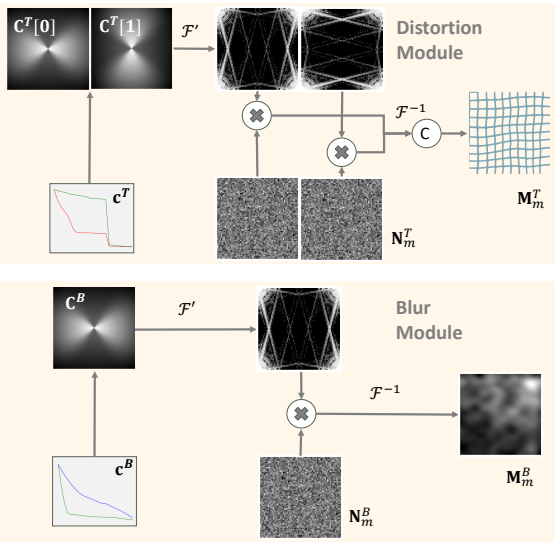


Figure 2. The procedure for generating the distortion parameter $\mathbf{M}^T$ and blur parameter $\mathbf{M}^B$.

3.1. Preliminary: Degradation model

This section will firstly introduce how Eq. (1) is inferred from traditional degradation model of image. In most types of degradation ignoring noise, the degradation operator $\mathcal{D}$ can be modeled as a spatially-varying convolution:

$$\hat{\mathbf{J}}[\mathbf{u}] = \mathcal{D}(\mathbf{I})[\mathbf{u}] = \sum_{\mathbf{u}'} k_{\mathbf{u}}(\mathbf{u} - \mathbf{u}')\mathbf{I}[\mathbf{u}'], \tag{2}$$

where $\mathbf{I}$ denotes the original image, $\hat{\mathbf{J}}$ denotes the degraded image. $k_{\mathbf{u}}$ denotes the point spread function (PSF) locating at coordinate $\mathbf{u}$ [13, 14], which can be expressed as:

$$k = |\mathcal{F}^{-1}(e^{-j2\pi\phi(\rho)})|^2, \tag{3}$$

$$\phi(\rho) = \sum_{\ell=1}^{L} \mathbf{a}[\ell] Z_\ell(\rho), \tag{4}$$

where $\mathcal{F}^{-1}$ denotes the inverse Fourier transform. $\phi(\rho)$ is the phase function, and ρ is the 2D polar coordinate in frequency space. Phase function $\phi(\rho)$ [25] is approximated by Zernike polynomials. $Z_\ell(\cdot)$ denotes the ℓ-th order Zernike polynomial with coefficients $\mathbf{a}[\ell]$. L is the maximum order of Zernike polynomial, and $L = 21$ in our case.

Following [7], the degradation due to turbulence can be decomposed into tilt and blur stages:

$$\mathcal{D}(\mathbf{I})[\mathbf{u}] = \hat{\mathcal{B}}(\hat{\mathcal{T}}(\mathbf{I}))[\mathbf{u}], \tag{5}$$

where

$$\text{Tilt:} \quad \hat{\mathcal{T}}(\mathbf{I})[\mathbf{u}] = \mathbf{I}[\mathbf{u} + \mathbf{t}_\mathbf{u}], \tag{6}$$

$$\text{Blur:} \quad \hat{\mathcal{B}}(\mathbf{I})[\mathbf{u}] = \sum_{\mathbf{u}'} k'_\mathbf{u}[\mathbf{u} - \mathbf{u}']\mathbf{I}[\mathbf{u}']. \tag{7}$$

The shift vector $\mathbf{t}_\mathbf{u}$ is positive correlated to the first two Zernike coefficients. The word tilt means the shift of pixel which causes distortion. The PSF $k'_\mathbf{u}$ is determined by the remaining Zernike coefficients: $\sum_{\ell=3}^{L} \mathbf{a}[\ell] Z_\ell(\rho)$.

Let $\{\hat{\mathbf{J}}_m\}_{m=1}^M$ denote a set of frames captured at static scene, which can be simulated by similar original image $\mathbf{I}$. For each frame, the random shifts and blurring effects caused by turbulence are parameterized by Zernike coefficients $\mathbf{A} = \{\mathbf{a}_{[0,0],m}, ..., \mathbf{a}_{\mathbf{u},m}\}_{m=1}^M$. It is shown in [8, 9, 24] that for each frame, these coefficients can be modeled as samples randomly from a normal distribution:

$$\mathbf{A}[m] \sim \mathcal{N}(0, \mathbf{\Sigma}), \mathbf{\Sigma}[\mathbf{u}, \mathbf{u}', \ell, \ell'] = \mathbb{E}(\mathbf{a}_\mathbf{u}[\ell]\mathbf{a}_{\mathbf{u}'}[\ell']). \tag{8}$$

3.2. Modeling of Turbulence via Random Process

In turbulence simulation, Zernike coefficients are often treated as samples from a scene-dependent random process. Because the covariance matrix $\mathbf{\Sigma}$ is high-dimensional, we use an autocorrelation matrix $\mathbf{C}$ instead in the sampling procedure below. The matrix $\mathbf{C}$ characterizes turbulence properties shared across frames within a scene, which is reasonable over short observation periods where the turbulence strength remains approximately stable.

Autocorrelation function: When we consider the relationship between different pixels in one image, the covariance matrix $\mathbf{\Sigma}$ will be too large. Covariance matrix $\mathbf{\Sigma}$ can be simplified [8, 9, 24] by assuming the sampling procedure of Zernike coefficients is wide-sense stationary. Now the value of $\mathbf{\Sigma}[\mathbf{u}, \mathbf{u}', :, :]$ is only related to $\xi = \mathbf{u} - \mathbf{u}'$. Then $\mathbf{\Sigma}[:, :, \ell, \ell]$ for coefficients with similar ℓ can be represented with correlation function C. This means $\mathbf{\Sigma}[\mathbf{u}, \mathbf{u}', \ell, \ell] = C_\ell(\xi)$ for any $\mathbf{u}$ and $\mathbf{u}'$ with similar distance ξ.

PSD (Power spectral Density): Directly drawing sample by rearranging autocorrelation function to covariance matrix, and than decomposing will cause high time complexity and space complexity. So refer to [8], drawing samples from the PSD of random process will make this practical if random process is wide stationary, which is the 2D Fourier transformation $\mathcal{F}$ of autocorrelation function.

3.3. Distortion simulation with random process

TMFS regards each $\mathbf{M}^T$ of each frame as a sample from scene specific random process. This random process is modeled by a 2D discrete anisotropy autocorrelation matrix pair $\mathbf{C}^T$ which only considers outer-mode correlation. TMFS draws correlated samples from random process by PSD [8]. The calculation of $\mathbf{M}^T$ is,

$$\mathbf{M}^T = \mathcal{F}^{-1}(|\mathcal{F}'(\mathbf{C}^T)| \odot \mathbf{N}^T)[: W, : H], \tag{9}$$

where $\mathcal{F}'$ is Fourier transformation with low amplitude filter introduced in regularization section. $\mathbf{C}^T$ is the discrete autocorrelation function of random process pair, with optimizable noise matrix pair $\mathbf{N}^T \in \mathbb{R}^{2\times W\times 2\times H\times 2}$ for each observed images. $\odot$ means element-wise multiplication. The reason of appearing in pair is the tilt directions in two axes. Due to the symmetry of Fourier transform for real number, the size of $\mathbf{C}^T$ and $\mathbf{N}^T$ is double to image.

In [8], the autocorrelation function is determined by physical turbulence parameters. Directly treating these parameters as optimization variables would lead to high computational cost. Following [8], TMFS represents C using two 1D functions: an isotropic correlation term c_1 and an anisotropic correlation term c_2:

$$w(\theta) = (0.5cos(2\theta) + 0.5), \tag{10}$$

$$C^T(\rho, \theta)[0] = w(\theta) \odot c_1(\rho) + (1 - w(\theta)) \odot c_2(\rho), \tag{11}$$

$$C^T(\rho, \theta)[1] = (1 - w(\theta)) \odot c_1(\rho) + w(\theta) \odot c_2(\rho), \tag{12}$$

where ρ is the distance between two points and θ is the angle between the line connecting two points and the horizon line. Function w is used to mix isotropic and anisotropic correlation. In implementation for efficiency, TMFS simply uses arrays $\mathbf{c}_1$, $\mathbf{c}_2$ as the discretization of 1D functions c_1 and c_2, and gets value by linear interpolation. They model 2D autocorrelation matrix $\mathbf{C}^T$.

It is worth mentioning that the usage of mixing weight is different to [8], for the convenience of constraint. Monotonicity is enforced by defining $\mathbf{c}_1$ and $\mathbf{c}_2$ as inverted cumulative sums of learnable non-negative sequences. Instead of optimizing $\{\mathbf{M}_m^T\}_{m=0}^M$ for M frames, TMFS optimizes a scene-level parameter $\mathbf{c}^T = \{\mathbf{c}_1, \mathbf{c}_2\}$ and per-frame variables $\{\mathbf{N}_m^T\}_{m=0}^M$.

3.4. Blur Simulation with Random Process

TMFS also treats the blur parameter $\mathbf{M}^B$ of each frame as a sample from a scene-specific random process. In contrast to distortion modeling, sampling the higher-order Zernike coefficients for blur requires accounting for two types of correlations. One is the inter-mode correlation, namely the correlation among coefficients of different orders ℓ at the same pixel. The other is the spatial correlation, namely the correlation among coefficients of the same order ℓ across different pixels.

For outer-pixel relationship, TMFS uses similar procedure as distortion simulation Eq. (9). And C^B is a set of autocorrelation functions corresponds to each higher-order Zernike coefficient. TMFS also discretizes C^B as $\mathbf{C}^B$ which is represented by $\mathbf{c}^B$. Different to $\mathbf{c}^T$, $\mathbf{c}^B$ has $L - 2$ pairs of arrays. Now out-pixel sample does not directly correspond to coefficient image, but the noise image which is used in inter-pixel sample.

Inter-pixel relationship is modeled by a fixed covariance matrix $\mathbf{\Sigma}^B$ defined in [8]. As the size of covariance matrix

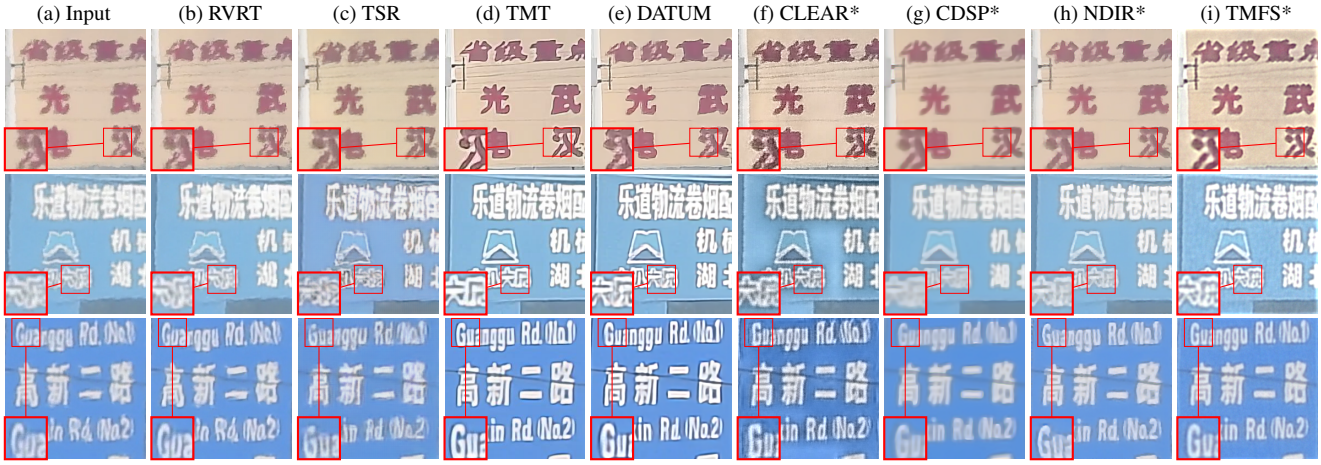


Figure 3. The restoration results on dataset RLR-AT [29]. The symbol of * means unsupervised method.

is only the square of the number of Zernike coefficients, TMFS directly decomposes it and then multiplies it with noise. The final calculation is,

$$\mathbf{M}^B[\mathbf{u}] = \mathbf{R} \times (\mathcal{F}^{-1}(|\mathcal{F}'(\mathbf{C}^B)| \odot \mathbf{N}^B)[:W,:H])[\mathbf{u}], \tag{13}$$

where $\mathbf{\Sigma}^B = \mathbf{R}\mathbf{R}^T$, with optimizable noise matrix $\mathbf{N}^B \in \mathbb{R}^{2\times W\times 2\times H\times (L-2)}$.

To efficiently handle spatially varying blur, TMFS adopts P2S [24], which uses precomputed PSF bases for acceleration. To better approximate near-delta PSFs, where P2S is less accurate, we further incorporate a delta-function basis into the model. As a result, instead of directly optimizing $\{\mathbf{M}_m^B\}_{m=0}^M$ for all M frames, TMFS optimizes a scene-level parameter $\mathbf{c}^B$ and per-frame variables $\{\mathbf{N}_m^B\}_{m=0}^M$.

3.5. Optimization and Regularization

In practice, TMFS uses INR to represent image which is more suitable than directly interpolation. The parameter θ of INR is initialized by fitting the mean of the observed frames. The total loss function is defined as:

$$L^{total} = \lambda_1 L^{recon} + \lambda_2 L^{grad} + \lambda_3 L^{lucky}. \tag{14}$$

The set of λ is hyper parameter that weights each loss term. Three loss functions are introduced below. The primary reconstruction loss is:

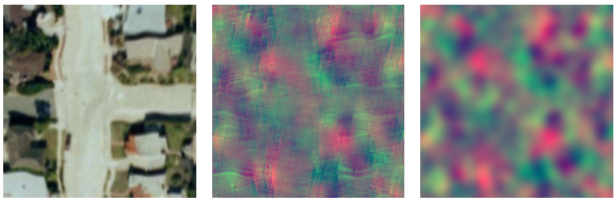
$$L^{recon} = \sum_{m=1}^{M} \|\mathbf{J}_m - \hat{\mathbf{J}}_m\|_1. \tag{15}$$

The optimizable parameters are θ, $\{\mathbf{N}_m^T\}_{m=0}^M$ in Eq. (9), $\mathbf{c}^T$ in Eq. (12), $\{\mathbf{N}_m^B\}_{m=0}^M$ in Eq. (13) and $\mathbf{c}^B$.

And in order to sharpen restored image, TMFS adds gradient loss according to lucky effect, that the original image should has higher gradient than observed image at same edge.

$$L^{grad} = \sum_{m=1}^{M} \sum_{\mathbf{u}\in edge(\mathbf{J}_m)} \|ReLU(\nabla\mathbf{J}_m[\mathbf{u}] - \nabla\hat{\mathbf{J}}_m[\mathbf{u}])\|_1, \tag{16}$$

where ∇ is the calculation of 2D gradient for image. And $edge$ is the location of edge in observed image calculated by Canny edge detection [5]. During optimizing L^{grad}, $\{\mathbf{M}_m^B\}_{m=0}^M$ are frozen.



(a) Degraded image (b) Without filter (c) With filter

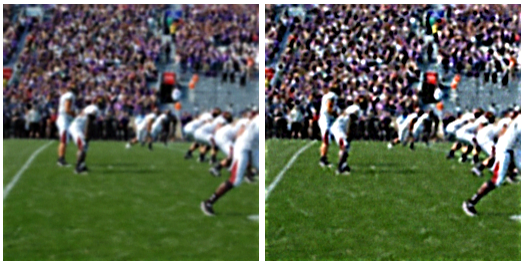
Figure 4. The comparison for the results of $\mathbf{M}^T$ without or with low amplitude filter.

Low amplitude filter: Filtering frequency components of PSD with low amplitude, can limit the fitting ability of distortion simulation to avoid local optimal solution.

$$\tau(\mathbf{P})[\mathbf{u}] = \begin{cases} 0 & \text{if } \mathbf{P}[\mathbf{u}] \le \alpha_p \\ 1 & \text{if } \mathbf{P}[\mathbf{u}] > \alpha_p \end{cases}, \tag{17}$$

$$\mathcal{F}'(\mathbf{C}^T) = \mathcal{F}(\mathbf{C}^T) \odot \tau(\mathcal{F}(\mathbf{C}^T)). \tag{18}$$

Function τ generates mask to filter frequency components with hyperparameter threshold α_p. Without limit on fitting ability, the tilt matrix will be affected by the content of image shown in Fig. 4. The comparison shows that the tilt image $\mathbf{M}^T$ will contain the shape of corresponding observed image without low amplitude filter.



(a) With L^{lucky} (b) Without L^{lucky}

Figure 5. The comparison for restoration results with or without lucky region regularization.

Lucky region: The following regularization encourages each pixel to be sharp in at least one frame:

$$\mathbf{K}[\mathbf{u}] = \max_{m} k_{\mathbf{u},m}([0,0]), \tag{19}$$

$$L^{\text{lucky}} = \sum_{\mathbf{u}} \begin{cases} 1 - \mathbf{K}[\mathbf{u}], & \mathbf{K}[\mathbf{u}] < \alpha_l, \\ 0, & \mathbf{K}[\mathbf{u}] \geq \alpha_l, \end{cases} \tag{20}$$

where $k_{\mathbf{u},m}([0,0])$ is the central PSF value at location $\mathbf{u}$ in frame m. A larger central value indicates weaker blur, so this term encourages each pixel to belong to a lucky region in at least one frame. Here $\alpha_l = 0.8$ controls how closely the PSF should approximate a delta function. The contribution of this regularization is shown in Fig. 5. We further observe that this term can produce a degenerate solution where lucky regions collapse into a single frame and the PSF for other frames becomes overly smooth. To avoid this, TMFS upweights the frame with the highest reconstruction loss during optimization.

3.6. High-Resolution Image Patch Processing

TMFS partitions the high-resolution image into overlapping patches and processes them independently. The patch-wise predictions are then merged to reconstruct the full-resolution output. Since the correlation decays with spatial distance, patch-wise processing still preserves the dominant correlation structure, especially with overlapping patches. As shown in Figure 6, the correlation function estimated on the RLR-AT dataset approaches zero at large distances, supporting this design. Additional full-resolution restoration results in the supplementary material show no visible stitching artifacts.

4. Experiments

In this section, we evaluate TMFS from three aspects. First, we compare it with existing restoration methods on both synthetic and real turbulence datasets. Second, we further assess its performance on a water-turbulence dataset. Third, we conduct ablation studies to analyze the contribution of each component.

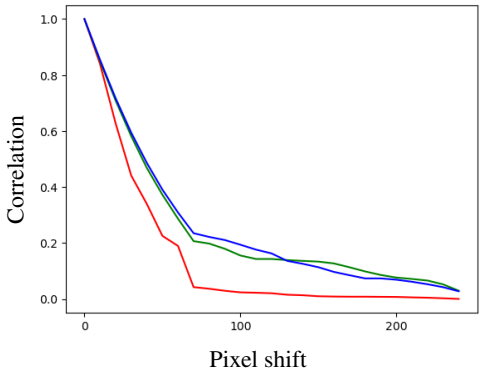


Figure 6. Isotropic correlation as a function of pixel shift for the third to fifth Zernike coefficients, computed from a restoration result on the RLR-AT dataset. The correlations for different coefficients are plotted in distinct colors.

Dataset: We use both synthetic and real datasets in the evaluation. Following [17, 32], the synthetic dataset is generated using the turbulence simulator of [24] under three turbulence levels, namely weak, medium, and strong. The clean images are taken from the UC Merced Land Use Dataset [30]. For real turbulence evaluation, we use the fixed-pattern subset of the OTIS dataset [11], the Heat Chamber dataset [24], and the small version of RLR-AT [29], which contains turbulence videos at 1920×1080 resolution. Both OTIS and RLR-AT are captured under natural atmospheric turbulence, whereas Heat Chamber is generated in a controlled setting and may not fully reflect real atmospheric turbulence [16]. The OTIS and Heat Chamber datasets both provide reference images for evaluation.

Methods for comparison: We compare TMFS with general restoration methods, supervised turbulence restoration methods, and unsupervised methods. RVRT [22] is a general multi-frame restoration method. TSR [19], TMT [33], and DATUM [32] are supervised methods specifically designed for turbulence restoration, all trained on synthetic data. For unsupervised baselines, we consider CLEAR [1], CDSP [29], and NDIR [20]. As NDIR only proposes the restoration of distortion and directly uses L0-sparse [28] for image deblurring. And CDSP only provides code for geometric distortion, so we also use L0-sparse. In the following comparison, all NDIR and CDSP are combined with L0-sparse.

The experiments are conducted on one NVIDIA RTX 4090 GPU with 20 observed 256×256 images as input unless specified. All scenes are static and multi-frame. For high resolution dataset such as RLR-AT, TMFS split the images into patches.

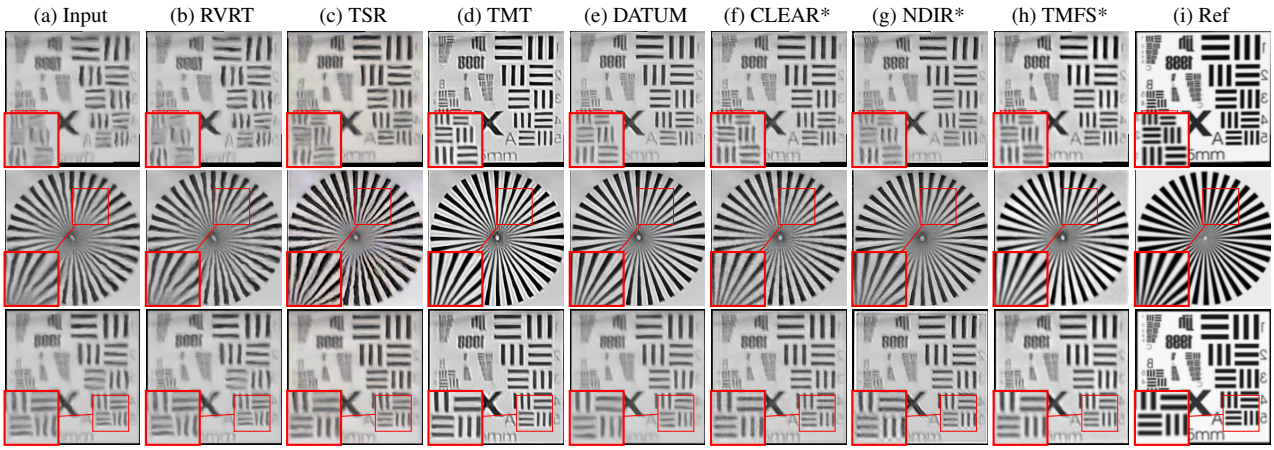


Figure 7. Visualization of the results from different methods on dataset OTIS. The symbol of * means unsupervised method.

DataSet	Metrics	Supervised				Unsupervised			
		RVRT	TSR	TMT	DATUM	CLEAR	CDSP	NDIR	TMFS
Weak($D/r_0 = 1$)	SSIM↑	0.4333	0.4516	**0.7405**	0.7112	0.7008	0.4961	0.5841	**0.7505**
	PSNR↑ (dB)	19.53	23.90	**24.21**	24.01	21.51	19.70	21.59	**25.70**
Middle($D/r_0 = 2$)	SSIM↑	0.3290	0.3278	0.4637	**0.58737**	0.3631	0.4019	0.4235	**0.5786**
	PSNR↑ (dB)	17.31	**21.77**	19.26	21.63	17.17	18.66	19.37	**22.10**
Strong($D/r_0 = 3$)	SSIM↑	0.2909	0.2816	0.3396	**0.4064**	0.1886	0.3380	0.3608	**0.4331**
	PSNR↑ (dB)	16.19	**20.35**	17.25	18.59	15.11	17.91	18.24	**19.671**
Heat$*$	SSIM↑	0.6838	0.7073	0.6940	**0.7298**	0.6792	0.6282	0.6847	**0.7080**
	PSNR↑ (dB)	20.21	**25.96**	19.71	20.877	20.16	19.76	19.89	**20.17**

Table 1. Performance comparison of different restoration methods over different datasets. The Heat dataset contains real-world images degraded by a physical heat chamber. The left part consists of supervised methods. And the right part consists of unsupervised methods. Bold means best result at that class.

4.1. Quantitative Comparison

The performance comparison is shown in Tab. 1. The metrics used in the comparison include peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). On most datasets, TMFS achieves the best performance among unsupervised methods. On the synthetic dataset under weak turbulence, it attains the best overall results. Under medium turbulence, its performance is surpassed by some supervised methods. Under strong turbulence, although TMFS achieves the best SSIM, the overall restoration quality remains limited, as indicated by the low PSNR values. Its performance on the TMT and Heat Chamber datasets is similar to that in the medium-turbulence setting.

Although the methods such as CLEAR attains relatively high SSIM on some synthetic datasets, its results are often over-sharpened, which is also reflected by lower PSNR and the qualitative comparisons. While supervised methods generally achieve strong performance on synthetic data, the qualitative results below suggest that their advantage does not fully transfer to visual improvement in real-world data.

4.2. Comparison of Visual Quality

Visual comparison on the real-world RLR-AT dataset are shown in Fig. 3. Most supervised methods, such as RVRT and TSR, still exhibit noticeable residual distortions, suggesting limited generalization from synthetic data to real-world data. Although DATUM achieves strong quantitative performance, it introduces visible artifacts and over-sharpening in the first and last rows of Fig. 3. CLEAR also tends to produce overly sharp results while failing to fully correct the distortions as seen in the last row. In contrast our method achieves more effective distortion removal, which we attribute to the proposed random-process-based distortion modeling rather than the INR-based grid parameterization used in NDIR. This advantage is particularly evident in the third row of Fig. 3. Qualitative comparisons on the OTIS dataset in Fig. 7 further support the same observation. We note that the official implementation of CDSP occasionally produces invalid results on OTIS, possibly due to implementation issues. Additional results on RLR-AT and OTIS are provided in the supplementary material.

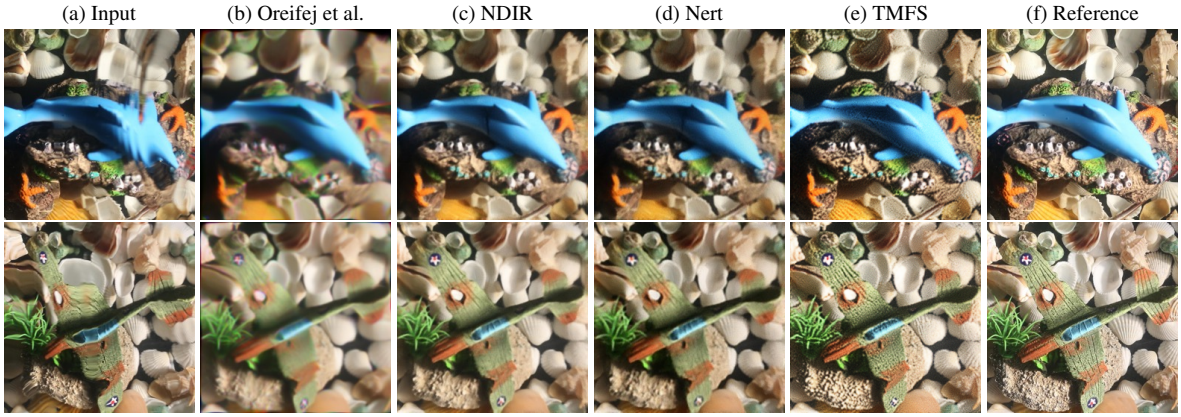


Figure 8. The restoration result of the real water turbulence tank datasets[21].

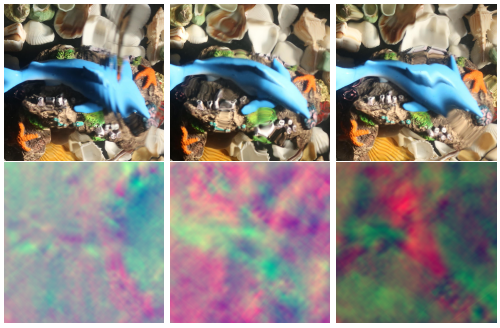


Figure 9. Visualization of tilt images. The first row shows the observed images, and the second row shows the corresponding tilt images.

DataSet	Metrics	T+L0	NDIR	CDSP
Middle	SSIM↑	**0.4802**	0.4235	0.4019
	PSNR↑ (dB)	**20.96**	19.37	18.66
Heat	SSIM↑	**0.7001**	0.6847	0.6282
	PSNR↑ (dB)	**20.09**	19.89	19.76

Table 2. Ablation study for traditional deblurring. The best result in each category is highlighted in bold.

4.3. Ablation Study

Unsupervised methods CDSP and NDIR only provide distortion mitigation code and we use deblurring method L0-sparse [28] following [20] for them. This section combines distortion mitigation module of TMFS and L0-sparse, which called T+L0. The combination is compared with CDSP and NDIR. As shown in Tab. 2, this combination still surpasses CDSP and NDIR in these dataset. It means the distortion mitigation module of our method surpasses CDSP and NDIR. The supplementary material includes a comparison with other blind deblurring methods [10, 31], in which TMFS also achieves the best performance.

4.4. Additional Application: Seeing through Water

The proposed TMFS is not only applicable to restore images degraded by atmospheric turbulence, but also can be used for other image restoration tasks. Here we apply TMFS on restoration of under-water images. The experiment is conducted on the real water-turbulence dataset of [21]. The dataset was captured using a water tank with an agitating pump to simulate real-world water turbulence. Since NERT [18] does not provide code, we only reproduce the result shown in its paper for visual comparison in Fig. 8. We also compare with the classical method of Oreifej et al. [26]. As shown in Fig. 8, although the reference image differs in illumination and is provided only for visual reference, TMFS recovers finer textures and sharper edges than competing methods. This suggests that, without relying on training data, our method can capture degradation characteristics shared across different turbulence domains. We further visualize the estimated tilt images in Fig. 9.

5. Conclusions

We presented TMFS, an unsupervised method for multi-frame turbulence restoration. By leveraging tilt-then-blur degradation model with a frame-correlated parameterization of distortion and blur, TMFS exploits the correlations among degradation processes of image frames to achieve better performance. Experiments show that TMFS outperforms existing unsupervised methods and generalizes better on real-world data than supervised methods trained on synthetic data. Future works will address two limitations of TMFS, computational cost due to per-sample training and unsatisfactory performance of restoring images degraded by severe turbulence distortion.

References

[1] Nantheera Anantrasirichai, Alin Achim, Nick G Kingsbury, and David R Bull. Atmospheric turbulence mitigation using complex wavelet-based fusion. *IEEE Transactions on Image Processing*, 22(6):2398–2408, 2013. 6

[2] Mathieu Aubailly, Mikhail A Vorontsov, Gary W Carhart, and Michael T Valley. Automated video enhancement from a stream of atmospherically-distorted images: the lucky-region fusion approach. In *Atmospheric Optics: Models, Measurements, and Target-in-the-Loop Propagation III*, pages 104–113. SPIE, 2009. 2

[3] Jeremy P Bos and Michael C Roggemann. Technique for simulating anisoplanatic image formation over long horizontal paths. *Optical Engineering*, 51(10):101704–101704, 2012. 3

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

[5] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986. 5

[6] Wai Ho Chak, Chun Pong Lau, and Lok Ming Lui. Subsampled turbulence removal network. *arXiv preprint arXiv:1807.04418*, 2018. 1

[7] Stanley H Chan. Tilt-then-blur or blur-then-tilt? clarifying the atmospheric turbulence model. *IEEE Signal Processing Letters*, 29:1833–1837, 2022. 2, 3

[8] Nicholas Chimitt and Stanley H Chan. Simulating anisoplanatic turbulence by sampling intermodal and spatially correlated zernike coefficients. *Optical Engineering*, 59(8): 083101–083101, 2020. 2, 3, 4

[9] Nicholas Chimitt, Xingguang Zhang, Zhiyuan Mao, and Stanley H Chan. Real-time dense field phase-to-space simulation of imaging through atmospheric turbulence. *IEEE Transactions on Computational Imaging*, 8:1159–1169, 2022. 2, 4

[10] Yuning Cui, Syed Waqas Zamir, Salman Khan, Alois Knoll, Mubarak Shah, and Fahad Shahbaz Khan. Adair: Adaptive all-in-one image restoration via frequency mining and modulation. In *13th international conference on learning representations, ICLR 2025*, pages 57335–57356. International Conference on Learning Representations, ICLR, 2025. 8

[11] Jérôme Gilles and Nicholas B Ferrante. Open turbulent image set (otis). *Pattern Recognition Letters*, 86:38–41, 2017. 6

[12] 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

[13] Joseph W Goodman. *Introduction to Fourier optics*. Roberts and Company publishers, 2005. 3

[14] Joseph W Goodman. *Statistical optics*. John Wiley & Sons, 2015. 3

[15] Russell C Hardie, Jonathan D Power, Daniel A LeMaster, Douglas R Droege, Szymon Gladysz, and Santasri Bose-Pillai. Simulation of anisoplanatic imaging through optical turbulence using numerical wave propagation with new validation analysis. *Optical Engineering*, 56(7):071502–071502, 2017. 3

[16] Paul Hill, Nantheera Anantrasirichai, Alin Achim, and David Bull. Deep learning techniques for atmospheric turbulence removal: a review. *Artificial Intelligence Review*, 58(4):101, 2025. 1, 6

[17] Ajay Jaiswal, Xingguang Zhang, Stanley H Chan, and Zhangyang Wang. Physics-driven turbulence image restoration with stochastic refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12170–12181, 2023. 1, 6

[18] Weiyun Jiang, Yuhao Liu, Vivek Boominathan, and Ashok Veeraraghavan. Nert: Implicit neural representations for general unsupervised turbulence mitigation. *arXiv preprint arXiv:2308.00622*, 2023. 1, 2, 8

[19] Darui Jin, Ying Chen, Yi Lu, Junzhang Chen, Peng Wang, Zichao Liu, Sheng Guo, and Xiangzhi Bai. Neutralizing the impact of atmospheric turbulence on complex scene imaging via deep learning. *Nature Machine Intelligence*, 3(10):876–884, 2021. 2, 6

[20] Nianyi Li, Simron Thapa, Cameron Whyte, Albert W Reed, Suren Jayasuriya, and Jinwei Ye. Unsupervised non-rigid image distortion removal via grid deformation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2522–2532, 2021. 1, 2, 6, 8

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

[22] Jingyun Liang, Yuchen Fan, Xiaoyu Xiang, Rakesh Ranjan, Eddy Ilg, Simon Green, Jiezhang Cao, Kai Zhang, Radu Timofte, and Luc V Gool. Recurrent video restoration transformer with guided deformable attention. *Advances in Neural Information Processing Systems*, 35:378–393, 2022. 2, 6

[23] Jingyun Liang, Jiezhang Cao, Yuchen Fan, Kai Zhang, Rakesh Ranjan, Yawei Li, Radu Timofte, and Luc Van Gool. Vrt: A video restoration transformer. *IEEE Transactions on Image Processing*, 2024. 2

[24] Zhiyuan Mao, Nicholas Chimitt, and Stanley H Chan. Accelerating atmospheric turbulence simulation via learned phase-to-space transform. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14759–14768, 2021. 2, 3, 4, 5, 6

[25] Robert J Noll. Zernike polynomials and atmospheric turbulence. *JOsA*, 66(3):207–211, 1976. 3

[26] 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. 8

[27] Jason D Schmidt. Numerical simulation of optical wave propagation with examples in matlab. *(No Title)*, 2010. 3

[28] Li Xu, Shicheng Zheng, and Jiaya Jia. Unnatural l0 sparse representation for natural image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1107–1114, 2013. 6, 8

[29] Shengqi Xu, Run Sun, Yi Chang, Shuning Cao, Xueyao Xiao, and Luxin Yan. Long-range turbulence mitigation: a large-scale dataset and a coarse-to-fine framework. In *European Conference on Computer Vision*, pages 311–329. Springer, 2024. 2, 5, 6

[30] Yi Yang and Shawn Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 270–279, 2010. 6

[31] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5728–5739, 2022. 8

[32] Xingguang Zhang, Nicholas Chimitt, Yiheng Chi, Zhiyuan Mao, and Stanley H Chan. Spatio-temporal turbulence mitigation: A translational perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2889–2899, 2024. 1, 2, 6

[33] Xingguang Zhang, Zhiyuan Mao, Nicholas Chimitt, and Stanley H Chan. Imaging through the atmosphere using turbulence mitigation transformer. *IEEE Transactions on Computational Imaging*, 2024. 1, 2, 3, 6