Diffeomorphic Template Registration for Atmospheric Turbulence Mitigation

Dong Lao
UCLA Vision Lab
lao@cs.ucla.edu

Congli Wang
UC Berkeley
congli.wang@berkeley.edu

Alex Wong
Yale Vision Lab
alex.wong@yale.edu

Stefano Soatto
UCLA Vision Lab
soatto@cs.ucla.edu

Abstract

We describe a method for recovering the irradiance underlying a collection of images corrupted by atmospheric turbulence. Since supervised data is often technically impossible to obtain, assumptions and biases have to be imposed to solve this inverse problem, and we choose to model them explicitly. Rather than initializing a latent irradiance ("template") by heuristics to estimate deformation, we select one of the images as a reference, and model the deformation in this image by the aggregation of the optical flow from it to other images, exploiting a prior imposed by Central Limit Theorem. Then with a novel flow inversion module, the model registers each image TO the template but WITHOUT the template, avoiding artifacts related to poor template initialization. To illustrate the robustness of the method, we simply (i) select the first frame as the reference and (ii) use the simplest optical flow to estimate the warpings, yet the improvement in registration is decisive in the final reconstruction, as we achieve state-of-the-art performance despite its simplicity. The method establishes a strong baseline that can be further improved by integrating it seamlessly into more sophisticated pipelines, or with domain-specific methods if so desired.

1. Introduction

Turbulent air motion, once temporally averaged and sampled to yield a digital picture, causes space- and time-varying blur. A model of the generative process for the measured data *given* the instantaneous "true" irradiance consists of simple integration over the sampling time interval resulting in discrete samples. Inferring the true irradiance is an ill-posed problem, for there is an equivalence class of unknowns that yield the same measurements. We can therefore choose *any* representative of the equivalence class, for instance one that would yield the sharpest solution. Traditionally the representative element is chosen as the (temporal) registered mean but since different images are blurred differently in different locations, the deformation is not identifiable at all locations, causing the represen-

tative to not have desirable properties, such as being blurry. Since the choice of representative is arbitrary, we can instead select the *sample* that has the most desirable properties, for instance the sharpest image ("lucky frame"), as a canonical reference from which to estimate the deformations. Thus, instead of solving the joint optimization for the deformations and the resulting average, we select the base frame $I_0 = I_\tau$ to correspond to a given image for some τ, for instance one that has the desired properties. Note that this choice is made without loss of generality for the purpose of estimating the deformations $\hat{w}_t$ mapping I_t to I_0, since the choice of canonical representative in an equivalence class is arbitrary, so the mean has no more dignity than any of the samples. Once we have estimated the deformations, however, there is a choice to be made on what criterion to use to infer the approximation of the underlying "true" irradiance I_0. We stress that I_0 is not just unknown, but unknowable, and therefore disambiguation rests on the choice of prior, or model, or complexity criterion.

Deep learning-based methods hide such arbitrary assumptions in the choice of architecture [27,33,44,48] (class of functions), or prior (loss function) [17,32,46,47]. However, inductive prior based on data used for training can be problematic for discovery, where by definition one is seeking the unknown. Generic priors may therefore be better suited for applications to computational astronomical imaging. Among non-learning methods, many directly estimate a reference image by using heuristics such as various forms of averaging [29,30,38,51], but mostly result in blurry reference images (Fig. 2). Computing deformation from this blurry reference image to other images severely impacts the quality of the deformation fields, which are sometimes of independent interest. We instead use an input image as the reference. By the Central Limit arguments, the mean of the deformation from this frame to other frames converges to the *inverse* of the deformation from the ground truth to this frame (Eq. (5)). Although straightforward, this approach involves inverting the optical flow field, which can be prone to numerical instabilities. We tackle this flow inversion problem by hard-coding an interpolation scheme, a key contributor to the performance of our method (Sect. 3.1.1).
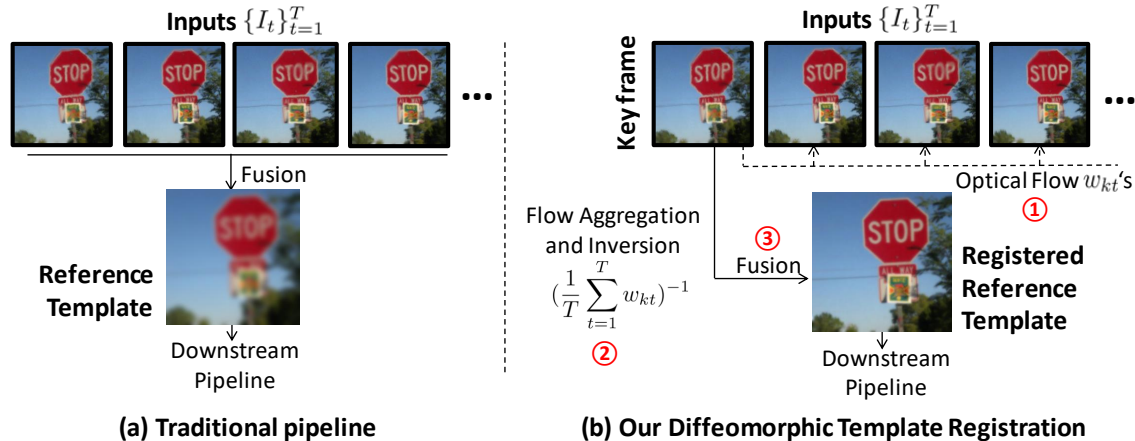
Figure 1. Schematic for Diffeomorphic Template Registration. *To mitigate atmospheric turbulence, one needs a "reference template" to estimate distortions. Existing methods use simple heuristics (e.g. averaging) to initialize this reference template and are susceptible to blurry artifacts due to a lack of registration. Instead, we select a keyframe and calculate warping from this keyframe to other frames by optical flow. By the Central Limit Theorem, the average of these optical flows converges to the warping from the keyframe to the ground truth reference template. With a novel flow inversion algorithm (detailed in Sect. 3.1.1), input frames can be registered to the reference template by Eq. (6), even without explicit access to an initialized reference template. This registered reference template can seamlessly fit into existing downstream atmospheric turbulence mitigation pipelines, such as blind deconvolution and lucky image fusion.*

The resulting method (Fig. 1) is simple and versatile since it is not subject to domain shift effects or inductive aberrations. To further emphasize the robustness of the method to the arbitrary choice of reference, in the experiments we simply use the first input frame as our reference frame. We also use a vanilla optical flow method, the good old-fashioned Horn & Schunck (L^2-regularized) optical flow [16]. Yet, despite these simplistic choices, we achieve state-of-the-art results on atmospheric turbulence mitigation benchmarks. If one longs for more sophistication, our method can be seamlessly combined with any existing sharpest image selection, optical flow estimation, downstream deconvolution, lucky-image fusion, and dynamic object filtering scheme. Our main point is that we do not need any of the heuristics underlying these methods in order to obtain state-of-the-art results. Our processing pipeline is generic and can be improved by replacing any component with better ones, including domain-specific regularization if and when appropriate or desired.

2. Related Works

Beyond hardware methods (*e.g.* adaptive optics [42]) to mitigate atmospheric turbulent distortion, there have been numerous software-based approaches. Some based on "lucky image" selection [2, 13, 21, 22, 30] choose a set of sharp images and compose them into a common frame; [45] employed deformation-guided spatial-temporal kernel regression to fuse the registered images together; [14] found the sharpest turbulence patches and enhanced them. To handle the blur artifacts within the composite frame, blind de-

convolution [10, 14, 15, 45] is typically used, assuming the turbulence can be modeled as a spatially invariant linear filter [20]. However, this assumption is often violated in atmospheric turbulence images due to the spatial-temporal changing displacements, leading to inaccurate results.

To address this issue, Zhu *et al.* [51] proposed registering each frame using B-spline based non-rigid registration to suppress geometric deformation, and then performs a temporal regression process to produce an image for blind deconvolution and generate the final output. [9] used a convolutional neural network (CNN) to reconstruct turbulence-corrupted video sequence. [50] proposed a complex steerable pyramid framework to decompose the degraded image sequence. [1] found that complex-valued convolutions better capture phase information than typical real-valued convolution. Furthermore, [24] formulated geometric distortions removal as an unsupervised training step for a deep network, but must be repeated for each test image. [43] reconstructed the wavefront aberration phase from the distorted image of the object to perform aberration correction for mitigating atmospheric turbulence. [46] used Monte Carlo dropout to estimate uncertainty maps to guide image restoration; [47] modeled the geometric distortion and blur at each pixel via variance maps and used to to aid the restoration of face images. In the domain of fluids, [41] used recurrent layers to recover a mesh model of water surface from a monocular video, while [26] undistorted dynamic refractive effects using warping network to remove geometric distortion and a color predictor net to further refine the restoration. Additionally, [48, 49] performed a number

restoration tasks, including denoising, deblurring, and dehazing. Although untested on atmospheric turbulence mitigation, the method appears promising.

Recent studies [30, 32] have proposed physics-inspired deep learning models that aim to capture the underlying physical processes that cause atmospheric turbulence. For example, TurbuGAN [8] incorporates physically accurate simulation into the forward measurement model. These models leverage knowledge of the statistical properties of turbulence and use them to guide the restoration process.

However, it is difficult to obtain aligned clean and corrupted image pairs. Hence, there exists a very limited amount of publicly available datasets for imaging through atmospheric turbulence. Amongst them, the most widely used ones are limited to few tens of images [2, 15, 24, 30]. More recently, [31, 32] proposed standardized benchmarks to evaluate the effectiveness of existing methods. Nonetheless, there remain challenges in collecting large-scale datasets for training. The majority of them, such as [25, 35], simulate realistic atmospheric turbulence effects; yet, this process can be expensive. Another line of work follows optics and leverages split-step simulation, using ray-tracing and wave-propagation [3, 12, 36, 37]. The less expensive synthesis techniques rely on random pixel displacement and a blur model [4, 23], but are far from realistic. To bridge this gap, [6] simulated anisoplanatic turbulence by sampling intermodal and spatially correlated Zernike coefficients. [7] further improved the speed of simulation to real-time by using a multi-aperture model [6], in conjunction with the phase-to-space transform [31], and an approximation of the Zernike correlation tensor in order to provide a scalable platform for evaluation.

Overall, while there have been significant advances in software-based approaches for mitigating atmospheric turbulence, there are still many challenges that need to be addressed. These include accurately modeling the turbulence, developing efficient and effective deep learning-based methods, and creating large-scale benchmark datasets that accurately reflect real-world scenarios.

3. Method

We consider an image sequence $\{I_t\}_{t=1}^T$ with $I_t : \Omega \mapsto \mathbb{R}^n$, where Ω is the image domain and $n = 1$ for gray-scale or $n = 3$ for RGB color channels.

3.1. Modelling Deformation

We start from the standard Lambertian assumption, considering the case where there is no blur but only deformation exists in the sequence. Given the original ground truth image (without turbulence) I, for each time t, I_t is deformed from I by a diffeomorphic warping w_t, and

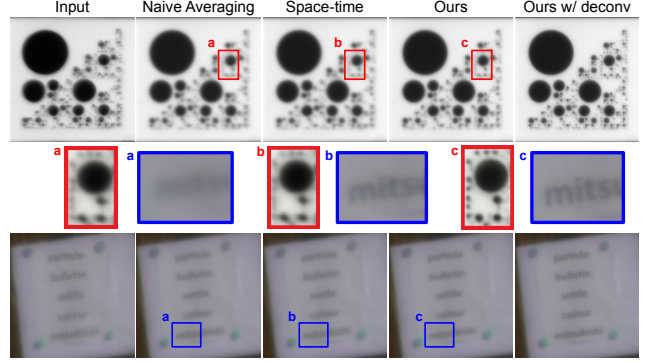$$I_t(w_t(x)) = I(x) + \eta_t(x), \quad x \in \Omega, \qquad (1)$$

Figure 2. **Sharper and more rigid reference template.** *Compared with naively averaging and space-time non-local averaging [30], our method improves the quality of the reference template. Highlighted regions are zoomed-in and sharpened for the sake of visualization, better viewed in $3\times$.*

where η denotes i.i.d. zero mean Gaussian noise. We assume each w_t is drawn from an i.i.d. distribution. With a stationary camera, by the Central Limit Theorem, $\frac{1}{T}\sum_{t=1}^T w_i \to 0$ as $T \to +\infty$ assuming the deformation is zero-mean. With Eq. (1), the turbulence mitigation problem can be written as

$$\underset{I,\{w_{t=1}^T\}}{\text{minimize}} \sum_{t=1}^T \int_{x\in\Omega} |I_t(w_t(x)) - I(x)|^2 \, \mathrm{d}x, \qquad (2)$$

in which I has an explicit optimizer

$$\hat{I}(x) = \frac{1}{T}\sum_{t=1}^T I_t(w_t(x)), \quad x \in \Omega, \qquad (3)$$

and w_t can be estimated by the optical flow from I to I_t.

However, without a good initialization of I, optical flow estimation for w_t fails. Meanwhile, warping between two frames w_{ij}, on the other hand, can be estimated by optical flow between I_i, I_j. By choosing a key frame I_k, through flow composition [18, 19], we have $w_t = w_{kt} \circ w_k$. In this way, we decompose the estimation of warping w_t for *each* t into a composition of the warping w_{kt} (directly estimated by optical flow) and a *single* warping I to each I_k.

Recall $\frac{1}{T}\sum_{t=1}^T w_i \to 0$. We assume that we have a sufficiently large number of frames, so for simplicity we denote $\frac{1}{T}\sum_{t=1}^T w_i \approx 0$. Therefore, given the composition of optical flow composition is a linear operation,

$$0 \approx \frac{1}{T}\sum_{t=1}^T w_t = \frac{1}{T}\sum_{t=1}^T w_{kt} \circ w_k = \left(\frac{1}{T}\sum_{t=1}^T w_{kt}\right) \circ w_k. \quad (4)$$

Therefore,

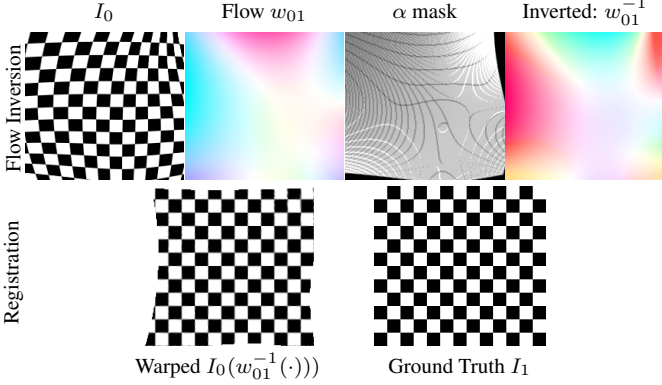$$w_k \approx \left(\frac{1}{T}\sum_{t=1}^T w_{kt}\right)^{-1}, \qquad (5)$$

Figure 3. **Example of flow inversion and registration.** *Given the flow w_{01}, we map the non-integer endpoints to the four nearest pixels, then construct an inverse flow w_{01}^{-1} by weighted averaging. When the corresponding weight is zero in α, we fill in the missing values by inpainting. This scheme successfully registers I_0 to the target frame I_1, even without having access to the ground truth I_1.*

where the exponent denotes the inverse of the diffeomorphic warping. Note that, computing Eq. (5) is non-trivial, as inverting a diffeomorphic warping (parametrized by 2-D optical flow) has no explicit solution. We describe how we implement this function in the Sect. 3.1.1.

Finally, by Eq. (3) and Eq. (5), we have

$$\hat{I}(x) \approx \frac{1}{T} \sum_{t=1}^{T} I_t(w_{kt} \circ (\frac{1}{T} \sum_{t=1}^{T} w_{kt})^{-1}(x)), \quad x \in \Omega, \ (6)$$

where we name $\hat{I}(x)$ the *template* of the sequence.

3.1.1 Flow Inversion

Inverting optical flow seems natural but is seldom practiced. When dealing with simple flow patterns like affine motion, an explicit solution is easy to compute in closed form. However, for more complex scenarios, such solutions are generally ill-posed. When pixels from one frame are mapped to non-integer positions in another frame, simply reversing the direction in the target frame is not a practical approach. Moreover, issues like occlusions at image boundaries and motion boundaries can result in missing values at certain pixel positions. One notable approach to inverting optical flow is offered by SMURF [39], which proposes learning of a flow inverting network on the fly. However, this method is not suitable for atmospheric turbulence removal problems with limited computational resources and processing time.

In this study, we propose a solution for estimating flow inversion, which is detailed in Alg. 1. Our approach involves determining the endpoint for each pixel in the reference frame and mapping the inverse flow from non-integer endpoints in the target frame to the four nearest integer

---

Algorithm 1 *Optical Flow Inversion.*

Require: Input flow field $w \in \mathbb{R}^{h \times w \times 2}$
1: Initialize: $w^{-1} = \text{zeros}(h, w, 2)$, $\alpha = \text{zeros}(h, w)$
2: **for** each grid point (i, j) in the image domain, **do**
3:      Compute endpoint $(\hat{i}, \hat{j}) = (i, j) + w(i, j)$
4:      Find four neighboring grid points of $(\hat{i}, \hat{j})$: $\mathcal{N}(\hat{i}, \hat{j}) = \{\text{floor}(\hat{i}, \hat{j}), \text{floor}(\hat{i}+1, \hat{j}), \text{floor}(\hat{i}, \hat{j}+1), \text{floor}(\hat{i}+1, \hat{j}+1)\}$
5:      **for** each $(i', j') \in \mathcal{N}(\hat{i}, \hat{j})$ **do**
6:          $w^{-1}(i', j') \mathrel{+}= -w(i, j) \times (2 - |(\hat{i}, \hat{j}) - (i', j')|_1^1)$
7:          $\alpha(i', j') \mathrel{+}= (2 - |(\hat{i}, \hat{j}) - (i', j')|_1^1)$
8: **for** each grid point (i,j) in the image domain, **do**
9:      **if** $\alpha(i, j) \neq 0$ **then** $w^{-1}(i, j) = w^{-1}(i, j)/\alpha(i, j)$
10: Inpaint $w^{-1}(i, j)$ by interpolation where $\alpha(i, j) = 0$
11: **return** Inverse flow w^{-1}

---

points where pixels are situated. Since multiple flow vectors may map to the same target pixel, we calculate a weighted average to derive the inverse flow. For pixels with zero weights, we fill in the missing values in the inverse flow using spatial interpolation (inpainting). An example is in Fig. 3. Even without having access to the true target I_1, given only the forward flow w_{01}, this scheme successfully inverts the flow and registers I_0 to the target frame.

The computation time for a 256×256 frame on a laptop CPU in MATLAB is less than 0.02 seconds, and with parallelization, we anticipate significant speed enhancements. This suggests that our approach has the potential to be applied in real-time computational imaging applications.

3.2. Modeling Blur

Temporal atmospheric turbulence causes wavefront distortion, resulting in image blurring. The image formation model has been discussed extensively in [5]. Following derivation shows that the per-frame degradation point spread function (PSF) can be approximated as a parametric Gaussian function.

3.2.1 Point Spread Function

In Fourier optics [11], the time-varying PSF $p_t(x)$ of an ideal imaging system with an effective focal length f at wavelength λ can be expressed as:

$$p_t(x) = |h_t(x)|^2, \quad h_t(x) = \mathcal{F}\left\{A(u)e^{iW_t(u)}\right\}\left(\frac{x}{\lambda f}\right). \tag{7}$$

where $A(u)$ represents the camera's entrance pupil at aperture coordinate u, $W_t(u)$ is the instantaneous wavefront that varies with time, and $\mathcal{F}$ denotes the Fourier transform. Assuming non-dispersivity, the wavefront remains unchanged with respect to the spectrum wavelength λ, and we will drop λ in the followings.

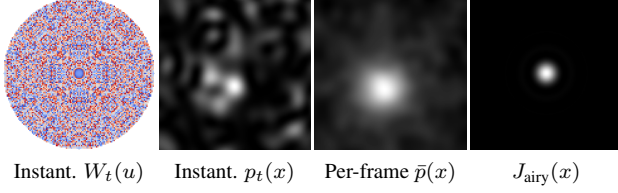| Instant. $W_t(u)$ | Instant. $p_t(x)$ | Per-frame $\bar{p}(x)$ | $J_{\text{airy}}(x)$ |

Figure 4. ***Approximating blurring kernels as Gaussians.*** *Given a set of instantaneous wavefronts (visualized within a modulo of 2π) and their PSFs, the per-frame PSF is approximated as a Gaussian function, characterized by shifts and a standard deviation.*

3.2.2 Turbulence Wavefront Decomposition

Wavefront $W_t(u)$ is typically decomposed into polynomial orders, *e.g.* the Zernike polynomials [34]. Here we are most interested in the following decomposition:

$$W_t(u) \;\; = \underbrace{a_t u}_{\text{1st order}} + \underbrace{b_t u^2}_{\text{2nd order}} + \underbrace{c_t V_t(u)}_{\geq \text{3rd orders}}, \qquad (8)$$

where a_t, b_t, and c_t are time-varying parameters due to turbulence, and $V_t(u)$ contains higher-order residuals.

3.2.3 Blurring Kernels Approximated as Gaussians

By substituting Eq. (8) into the kernel $h_t(x)$ in Eq. (7) and utilizing the convolution theorem with $\otimes$ being the convolution operator, we can rewrite $h_t(x)$ as follows:

$$h_t(x) = \mathcal{F}\left\{ A(u) e^{i a_t u} e^{i b_t u^2} e^{i c_t V_t(u)} \right\}$$
$$= \mathcal{F}\left\{ e^{i a_t u} \right\} \otimes \mathcal{F}\left\{ A(u) e^{i b_t u^2} \right\} \otimes \mathcal{F}\left\{ e^{i c_t V_t(u)} \right\}$$
$$= g_{\text{airy}}(x - \Delta x_t; \sigma_t) \otimes \mathcal{F}\left\{ e^{i c_t V_t(u)} \right\}, \qquad (9)$$

where it is assumed that $A(u)$ is a circular aperture, as in most situations. Optically, $g_{\text{airy}}(x - \Delta x_t; \sigma_t)$ is the spatially-shifted (Δx_t), defocused (σ_t), amplitude Airy disc diffraction pattern of the imaging system, and it is a real number due to symmetry regarding u. By assuming in Eq. (8):

$$c_t \sim \mathcal{N}(0, \sigma_c^2) \quad \text{and} \quad V_t(u) \ll 1,$$

we obtain:

$$\mathcal{F}\left\{ e^{i c_t V_t(u)} \right\} \approx \delta(x) + i c_t \mathcal{F}\left\{ V_t \right\}(x). \qquad (10)$$

This allows rewriting Eq. (9) to approximate $p_t(x)$ as:

$$p_t(x) = |h_t(x)|^2 = \Re\{h_t(x)\}^2 + \Im\{h_t(x)\}^2$$
$$\approx J_{\text{airy}}(x - \Delta x_t; \sigma_t) + o_t(x)$$
$$\approx J_{\text{airy}}(x - \Delta x_t; \sigma_t), \qquad (11)$$

because $o_t(x)$ is small, where:

$$J_{\text{airy}}(x - \Delta x_t; \sigma_t) = |g_{\text{airy}}(x - \Delta x_t; \sigma_t)|^2$$
$$o_t(x) = |c_t|^2 \cdot |g_{\text{airy}}(x - \Delta x_t) \otimes \mathcal{F}\{V_t\}(x)|^2 .$$

$J_{\text{airy}}(x - \Delta x_t; \sigma_t)$ is the intensity Airy disc function, with defocus σ_t and shift Δx_t. For each frame, the per-frame PSF is the aggregation of multiple instantaneous PSF $p_t(x)$, and thus we can approximate it as a Gaussian function, of mean μ and standard deviation σ:

$$\bar{p}(x) = \sum_{t \in \text{frame}} p_t(x) \approx G(x - \mu; \sigma). \qquad (12)$$

The approximation in Eq. (12) is illustrated in Fig. 4.

3.3. Blind Deconvolution of A Uniform Kernel

The degradation of each frame can thus be described as:

$$\hat{I}(x) \approx \bar{p}(x) \otimes I(x). \qquad (13)$$

With the proposed diffeomorphic template registration, the template of the sequence is obtained, and Eq. (13) is solved as a blind deconvolution problem. The problem is typically addressed using total variation (TV) regularization or other possible regularizers. The objective is to minimize:

$$\underset{I}{\text{minimize}} \int_{x \in \Omega} |\hat{I}(x) - \bar{p}(x) \otimes I(x)|^2 + \alpha |\nabla I(x)| \, dx, \qquad (14)$$

where $\alpha > 0$ is a tradeoff parameter. The problem is trivially solved using automatic differentiation engines.

There are two sources of blur in the template. The first source is caused by optical flow end-point error, which is typically assumed to be a zero-mean Gaussian distribution. With a sufficiently large number of frames, this results in a Gaussian blur being applied to the image. It can be proven that the two sources share the same Gaussian kernel. The second source of blur is the inherent blurring caused by atmospheric turbulence, which cannot be explicitly measured. Other methods simply apply a training loss. By the Central Limit Theorem, with a sufficiently large number of frames, we assume that the blur adds up to a Gaussian distribution.

3.4. Implementation Details

The full algorithm pipeline is presented in Alg. 2. Our algorithm offers flexibility in choosing the optical flow for estimating warpings. In our experiments, we use Horn-Schunk [16] optical flow as a default choice, unless specified otherwise. The computation of optical flow is efficiently parallelized using MATLAB's `parfor` function. Flow composition and image interpolation follow [19].

One notable advantage of our approach is that it does not require a predefined reference template, so one is free to select any frame from the image sequence as the keyframe. In

Algorithm 2 *Full Algorithm.*

Require: Input image sequence $\{I_i, i = 1, 2, \ldots, t\}$
1: Choose the reference frame I_{ref}
2: **for** $i = 1, \ldots, t$ **do**   ▷*Can be parallelized*
3:     Estimate $w_i = \text{Flow}(I_{\text{ref}}, I_i)$
4: Compute $\bar{w} = \frac{1}{t} \sum_{i=1}^{t} w_i$
5: Compute $\bar{w}^{-1}$ by Alg. 1
6: **for** $i = 1, \ldots, t$ **do**
7:     Compose flow: $\hat{w}_i = \bar{w}^{-1} \circ w_i$
8: Compute $I_{\text{template}} = \frac{1}{t} \sum_{i=1}^{t} \hat{w}_i^{-1}(I_i)$
9: Blind deconvolution $I = \text{Deconv}(I_{\text{template}})$
10: (Optional) $I_{\text{ref}} = I$, repeat Step 2 to 9
11: Return I

Single Frame			Multi Frame		
Method	PSNR	SSIM	Method	PSNR	SSIM
TRDN [46]	18.42	0.642	CLEAR [2]	18.91	0.757
Uformer [44]	18.68	0.658	NDIR [24]	20.08	0.763
Restormer [48]	19.12	0.684	Ours-template	20.30	0.780
TurbNet [32]	19.76	0.693	TurbRecon [30]	20.62	0.787
PiRN [17]	20.59	0.712	Ours-lucky	**20.74**	**0.789**

Table 1. **Comparison on HeatChamber dataset.** *While single-image methods can attain competitive PSNR, they exhibit low SSIM due to limited ability to handle deformation. Among multi-frame methods, our reference template (Ours-template) surpasses NDIR even without additional downstream processing. When integrated with the lucky image fusion pipeline (Ours-lucky), our approach enhances TurbRecon in both PSNR and SSIM metrics.*

our experiments, we simply use the first frame for demonstration. However, one can also choose the sharpest image in the sequence as the reference frame or iterate by using intermediate results as in Step 10.

For blind deconvolution, we have incorporated the ADMM blind deconvolution method from the public codebase in [30]. The template then be integrated with a majority of turbulence mitigation pipelines. We demonstrate this by combining our method with the lucky image fusion module of [30], as in Sect. 4. A demo code is available in the supplementary material.

4. Experiments

Experimental Settings. Due to the nature of the task of atmospheric turbulence mitigation, there is a limited choice of datasets with available ground truth for quantitative evaluation. We employed two datasets: HeatChamber [32] and CLEAR-sim [2]. HeatChamber contains 200 sequences of 90 images each at 220×220 pixel resolution, along with corresponding ground truth images. It was captured by displaying ground truth images on a screen 20 meters from the camera while inducing controlled turbulence using heat sources. To our knowledge, HeatChamber is the only real-world public dataset that provides ground truth for this purpose. In contrast, CLEAR-sim is a simulation dataset with 23 image sequences, featuring resolutions ranging from 256×256 to 1024×1024 pixels. Each sequence contains 96 to 100 frames with ground truth images.

We evaluate the results on both datasets using standard reconstruction metrics: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM). Additionally, we present quantitative results on datasets, such as the one introduced by [15], where ground truth is not available.

In our experimental comparisons, we focus on two representative methods: model-based methods TurbRecon [30], and learning-based method NDIR [24]. TurbRecon generates a reference template through space-time non-local averaging, followed by registration using optical flow and

lucky image fusion. For a fair comparison, we adopt the same optical flow method [28] as TurbRecon, and kept all the hyper-parameters consistent with it. NDIR models non-rigid distortions as deformable grids and utilizes deep neural networks to optimize both the grids and the underlying clean image simultaneously. NDIR initializes the reference template using the mean of the input images.

Quantitative Results. We present results on HeatChamber in Tab. 1 and also compare with five deep-learning-based single-image reconstruction methods. Although single-image approaches can achieve competitive PSNR due to their capability to generate realistic reconstructions, their effectiveness is limited in capturing deformation. This limitation results in generally poor SSIM scores, as inferring deformation from a single image is suboptimal.

In comparison to multi-frame methods, our reference template (referred to as Ours-template) outperforms NDIR even without any subsequent processing steps. Notably, the SSIM exceeds that of NDIR by 0.018, indicating that the diffeomorphic registration scheme better captures the scene's structure. We also test our method with a range of input frames as a sensitivity study (Fig. 5), where we consistently improve over temporal averaging and also NDIR for 10 input frames. As stated in Sect. 3.4, our template seamlessly integrates into a majority of existing pipelines. By replacing TurbRecon's space-time non-local averaging scheme with our approach and adopting its lucky image fusion pipeline, the results (labeled as Ours-lucky) also demonstrate improvements over the original TurbRecon in both PSNR and SSIM metrics. In addition, we present results comparing our approach with NDL [51], which also suggests a registration-deconvolution process. However, they employ a different registration strategy by using the average of input frames as a reference template and utilizing B-spline for deformation modeling instead of optical flow. Due to the extensive computational time (exceeding 1 hour for a 220×220 gray-scale image) required by NDL, we

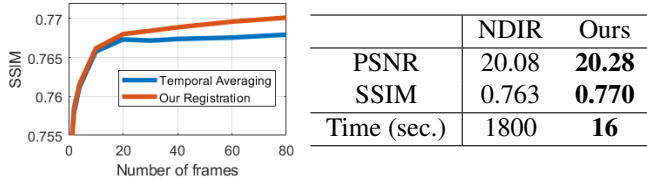|	NDIR	Ours
PSNR	20.08	**20.28**
SSIM	0.763	**0.770**
Time (sec.)	1800	**16**

Figure 5. **Sensitivity to the number of input frames.** *Left: Our method yields improvements over temporal averaging with respect to the number of frames. Right: Using 10 frames as input, ours outperforms NDIR in PSNR and SSIM.*

Key frame	Unlucky	Arbitrary (1$^{\text{st}}$ frame)	Lucky
SSIM	0.770	0.771	0.771

Table 2. **Sensitivity to frame selection.** *We evaluate the registered template (before deconvolution) using the sharpest ("lucky") and blurriest ("unlucky") and show that our method is not sensitive.*

Method	NDIR [24]	TurbRecon [30]	Ours-lucky
PSNR	24.13	24.73	**25.95**
SSIM	0.871	0.897	**0.906**
Resolution	220 × 220	1024 × 1024	1024 × 1024

Table 3. **Results on CLEAR-sim.** *Our method and TurbRecon can handle higher resolutions than NDIR by using optical flow, which avoids the computationally intensive joint optimization. We outperform NDIR and TurbRecon on both PSNR and SSIM.*

conduct comparisons on gray-scale images. Our method enhances the PSNR from 20.65 to 20.84 and SSIM from 0.715 to 0.723, while achieving a speed improvement of $\sim 100\times$.

Our experiments on the CLEAR-sim dataset consistently support our prior findings. It is important to note that CLEAR-sim and HeatChamber differ in resolution. Utilizing an Nvidia GTX 1080Ti GPU with 11 GB of memory, our experiments reveal that NDR, can only process 10 input frames at a low spatial resolution of 220×220. In contrast, both our approach and TurbRecon, which employ optical flow to explicitly address deformation, can handle the full resolution of 1024×1024. This highlights the versatility of leveraging optical flow, eliminating the need for joint optimization that is computationally expensive. The quantitative results align with those from HeatChamber, demonstrating that TurbRecon outperforms NDR, with our method yielding the best performance among the three approaches.

Qualitative Results. Fig. 6 presents qualitative results from both datasets. The images are zoomed-in to highlight intricate reconstruction details. Our approach consistently produces reconstructions with minimal distortion compared to the baselines, exhibiting a crisper appearance. It is important to note that we maintain consistent parameters with TurbRecon in optical flow and deconvolution, ensuring that the enhanced reconstruction quality is not a result of parameter selection but stems from superior registration. This also emphasizes that improved registration not only leads to rigid reconstruction with reduced distortion, but also mitigates blur caused by misalignment. A key comparison is in HeatChamber (row 1, Fig. 6), where existing methods introduce blurring of lights (highlighted in red) in different directions. Unlike them, we do not rely on an averaging scheme, which enables us to improve on the registration step to yield sharper results. Similar blurring artifacts can be observed in row 2 on the poles of the gate and row 3 (CLEAR-sim) on

the logo for both NDIR and TurbRecon.

To provide a comprehensive view, we include two additional examples from the CodedTarget[1] and the HotAir dataset [15]. While these sequences lack a ground truth for quantitative evaluation, a visual comparison indicates the consistent superiority of our method.

5. Discussions and Conclusions

Conclusions. We have presented a simple method; its simplicity makes it easy to extend, and serves as a foundation for more intricate approaches. The crux of it relies on an observation that the mean of the deformation induced by turbulence will converge to the inverse deformation of the flow by the Central Limit Theorem. This assumption makes it possible to register the frames to the latent irradiance to be solved (the "template") but without the template. Counter to current trends, our method does not rely on any data-driven processes, but rather an analytical solution. Hence, it is not subject any form of learning bias. Given that the problem is inherently ill-posed, such biases will result in hallucinations, which translates to artifacts and aberrations in the output. Yet, we do make an assumption, which is most generic, making the method widely applicable, and can be seamlessly integrated into mainstream atmospheric turbulence mitigation pipelines. Because of its simplicity, the method is robust and does not exhibit high sensitivity to the choice of frame, i.e. we achieve state-of-the-art perform by using just the 1st frame as the reference frame and the use of simple optical flow. Of course, more informed choices of frame (prior, e.g. sharpest frame) can naturally lead to improved results. We leave the trade-off between performance and generality of the method up to the user.

Limitations and future works. While we are able to achieve state-of-the-art performance on standard benchmarks, admittedly the run time remains a limitation of our approach. The main speed bottleneck of our approach is in the optical flow estimation. The processes of flow inversion and image aggregation involve primarily straightforward interpolations without iterative optimization and contribute negligible computational overhead. Presently, when utilizing the same optimization-based optical flow as TurbRecon,
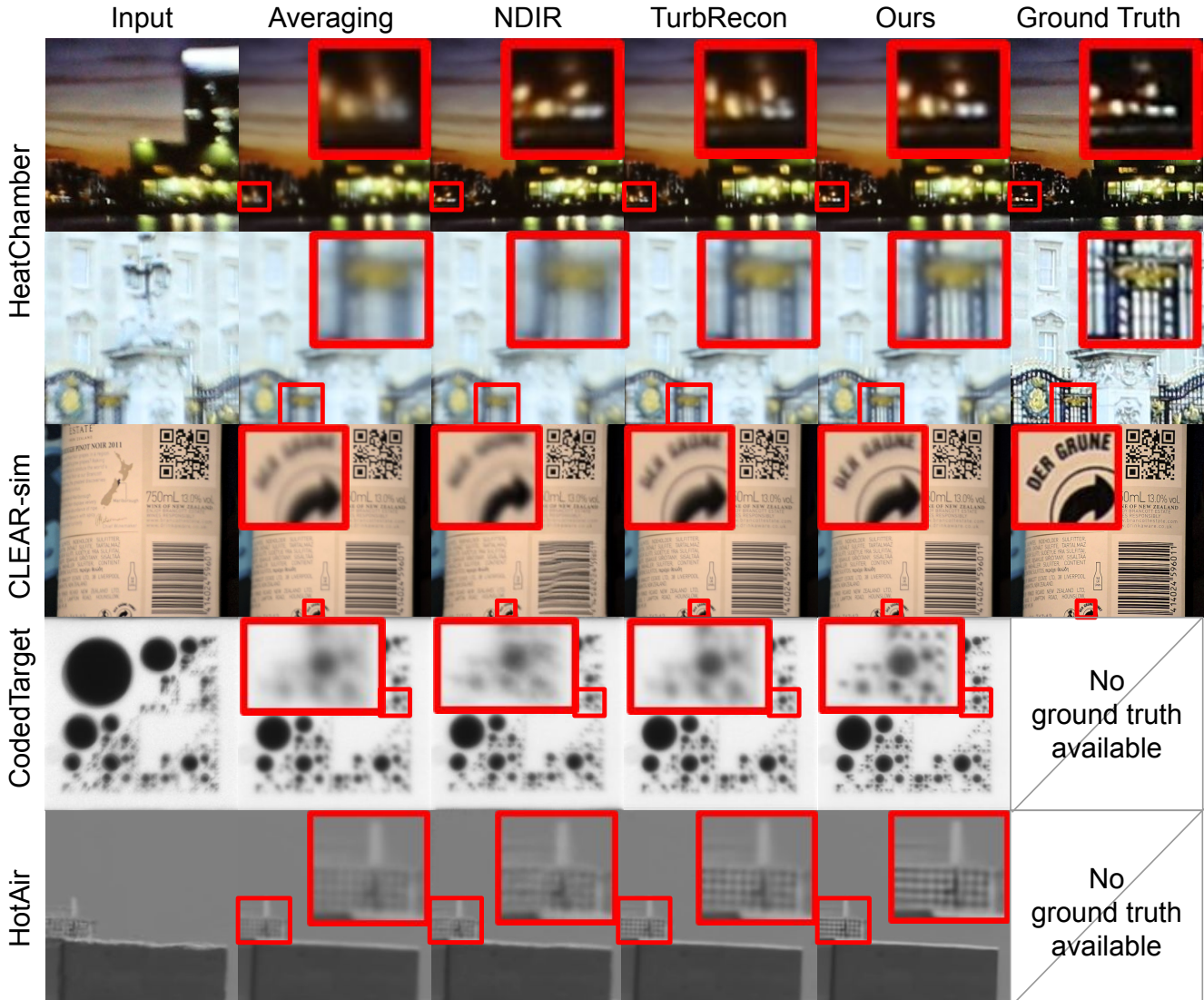
---

[1]https://cvpr2023.ug2challenge.org/

Figure 6. **Representative qualitative results on various datasets.** *Our approach yields crisper and less distorted reconstructions. Note that enhanced registration not only results in reduced distortion, but also mitigates blur caused by misalignment.*

the processing time for 100 frames sized at 220×220 is 17 seconds on an Intel i9-8950hk laptop CPU. When combined with ADMM blind deconvolution, the total processing time is 31 seconds. We anticipate a substantial speed acceleration by employing contemporary deep learning methods, such as RAFT [40], as a substitute for Horn & Schunck. This is expected to reduce the total processing time by half. For reference, TurbRecon requires an additional spatial-time nonlocal averaging module, and the total processing time is 42 seconds. NDIR requires 30 minutes to handle a duration of 10 minutes.

At the core of our method is optical flow, yet mainstream optical flow methods are not specifically tailored for atmospheric turbulence data. Nonetheless, as turbulence simulators advance [32], training a flow customized for air turbu-

lence becomes feasible. This limitation is anticipated to be transformed into a strength, since flow integrates as a plug-and-play module, allowing for straightforward upgrades to our method without alterations to the pipeline.

Finally, we currently do not consider dynamic scenes. Although occlusion induced by motion violates the diffeomorphic assumptions, in Eq. (13) there is in fact no restriction posed on w. By strategies like outlier rejection, adjustments can be made to Eq. (4), enabling the formulation to effectively accommodate scenarios involving either a moving camera or moving objects. Nevertheless, the main goal of this paper is to show that registration can be achieved without the need for an explicit template. Challenges related to dynamic scenes and moving objects will be addressed in future extensions.

References

[1] Nantheera Anantrasirichai. Atmospheric turbulence removal with complex-valued convolutional neural network. *Pattern Recognition Letters*, 171:69–75, 2023. 2

[2] 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. 2, 3, 6

[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] Wai Ho Chak, Chun Pong Lau, and Lok Ming Lui. Subsampled turbulence removal network. *Mathematics, Computation and Geometry of Data*, 1(1):1–33, 2021. 3

[5] Stanley H Chan. Tilt-then-blur or blur-then-tilt? Clarifying the atmospheric turbulence model. *IEEE Signal Processing Letters*, 29:1833–1837, 2022. 4

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

[7] 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. 3

[8] Brandon Y Feng, Mingyang Xie, and Christopher A Metzler. Turbugan: An adversarial learning approach to spatially-varying multiframe blind deconvolution with applications to imaging through turbulence. *IEEE Journal on Selected Areas in Information Theory*, 3(3):543–556, 2022. 3

[9] Jing Gao, Nantheera Anantrasirichai, and David Bull. Atmospheric turbulence removal using convolutional neural network. *arXiv preprint arXiv:1912.11350*, 2019. 2

[10] 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

[11] Joseph W Goodman. *Introduction to Fourier optics*. Roberts and Company Publishers, 2005. 4

[12] 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

[13] Russell C Hardie, Michael A Rucci, Alexander J Dapore, and Barry K Karch. Block matching and wiener filtering approach to optical turbulence mitigation and its application to simulated and real imagery with quantitative error analysis. *Optical Engineering*, 56(7):071503–071503, 2017. 2

[14] Renjie He, Zhiyong Wang, Yangyu Fan, and David Fengg. Atmospheric turbulence mitigation based on turbulence extraction. In *IEEE International Conference on Acoustics,*

[15] Michael Hirsch, Suvrit Sra, Bernhard Schölkopf, and Stefan Harmeling. Efficient filter flow for space-variant multiframe blind deconvolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 607–614. IEEE, 2010. 2, 3, 6, 7

[16] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981. 2, 5

[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] Dong Lao and Ganesh Sundaramoorthi. Minimum delay moving object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4250–4259, 2017. 3

[19] Dong Lao and Ganesh Sundaramoorthi. Extending layered models to 3D motion. In *Proceedings of the European Conference on Computer Vision*, pages 435–451, 2018. 3, 5

[20] Chun Pong Lau, Carlos D Castillo, and Rama Chellappa. Atfacegan: Single face semantic aware image restoration and recognition from atmospheric turbulence. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(2):240–251, 2021. 2

[21] Chun Pong Lau, Yu Hin Lai, and Lok Ming Lui. Restoration of atmospheric turbulence-distorted images via RPCA and quasiconformal maps. *Inverse Problems*, 35(7):074002, 2019. 2

[22] Nicholas M Law, CD Mackay, Richard G Dekany, M Ireland, JP Lloyd, AM Moore, JG Robertson, P Tuthill, and HC Woodruff. Getting lucky with adaptive optics: Fast adaptive optics image selection in the visible with a large telescope. *The Astrophysical Journal*, 692(1):924, 2009. 2

[23] Kevin R Leonard, Jonathan Howe, and David E Oxford. Simulation of atmospheric turbulence effects and mitigation algorithms on stand-off automatic facial recognition. In *Optics and Photonics for Counterterrorism, Crime Fighting, and Defence VIII*, volume 8546, pages 182–198. SPIE, 2012. 3

[24] 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. 2, 3, 6, 7

[25] Yi Li, Eric Perlman, Minping Wan, Yunke Yang, Charles Meneveau, Randal Burns, Shiyi Chen, Alexander Szalay, and Gregory Eyink. A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence. *Journal of Turbulence*, (9):N31, 2008. 3

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

[27] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021. 1

[28] Ce Liu. *Beyond pixels: Exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009. 6

[29] Yifei Lou, Sung Ha Kang, Stefano Soatto, and Andrea L Bertozzi. Video stabilization of atmospheric turbulence distortion. *Inverse Problems and Imaging*, 7(3):839–861, 2013. 1

[30] 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. 1, 2, 3, 6, 7

[31] 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. 3

[32] Zhiyuan Mao, Ajay Jaiswal, Zhangyang Wang, and Stanley H Chan. Single frame atmospheric turbulence mitigation: A benchmark study and a new physics-inspired transformer model. In *Proceedings of the European Conference on Computer Vision*, pages 430–446. Springer, 2022. 1, 3, 6, 8

[33] Armin Mehri, Parichehr B Ardakani, and Angel D Sappa. Mprnet: Multi-path residual network for lightweight image super resolution. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2704–2713, 2021. 1

[34] Robert J Noll. Zernike polynomials and atmospheric turbulence. *Journal of the Optical Society of America*, 66(3):207–211, 1976. 5

[35] Eric Perlman, Randal Burns, Yi Li, and Charles Meneveau. Data exploration of turbulence simulations using a database cluster. In *Proceedings of the ACM/IEEE Conference on Supercomputing*, pages 1–11, 2007. 3

[36] Michael C Roggemann and Byron M Welsh. *Imaging through turbulence*. CRC press, 2018. 3

[37] Jason D Schmidt. *Numerical simulation of optical wave propagation with examples in MATLAB*. SPIE, 2010. 3

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

[39] Austin Stone, Daniel Maurer, Alper Ayvaci, Anelia Angelova, and Rico Jonschkowski. Smurf: Self-teaching multi-frame unsupervised raft with full-image warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3887–3896, 2021. 4

[40] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision*, pages 402–419. Springer, 2020. 8

[41] Simron Thapa, Nianyi Li, and Jinwei Ye. Dynamic fluid surface reconstruction using deep neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21–30, 2020. 2

[42] Congli Wang, Qiang Fu, Xiong Dun, and Wolfgang Heidrich. Megapixel adaptive optics: Towards correcting large-scale distortions in computational cameras. *ACM Transactions on Graphics*, 37(4):1–12, 2018. 2

[43] Kaiqiang Wang, MengMeng Zhang, Ju Tang, Lingke Wang, Liusen Hu, Xiaoyan Wu, Wei Li, Jianglei Di, Guodong Liu, and Jianlin Zhao. Deep learning wavefront sensing and aberration correction in atmospheric turbulence. *PhotoniX*, 2(1):1–11, 2021. 2

[44] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17683–17693, 2022. 1, 6

[45] Yuan Xie, Wensheng Zhang, Dacheng Tao, Wenrui Hu, Yanyun Qu, and Hanzi Wang. Removing turbulence effect via hybrid total variation and deformation-guided kernel regression. *IEEE Transactions on Image Processing*, 25(10):4943–4958, 2016. 2

[46] Rajeev Yasarla and Vishal M Patel. Learning to restore images degraded by atmospheric turbulence using uncertainty. In *IEEE International Conference on Image Processing*, pages 1694–1698. IEEE, 2021. 1, 2, 6

[47] Rajeev Yasarla and Vishal M Patel. Cnn-based restoration of a single face image degraded by atmospheric turbulence. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 4(2):222–233, 2022. 1, 2

[48] 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. 1, 2, 6

[49] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14821–14831, 2021. 2

[50] 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. 2

[51] 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. 1, 2, 6