

Accelerating Atmospheric Turbulence Simulation via Learned Phase-to-Space Transform

Zhiyuan Mao, Nicholas Chimitt, Stanley H. Chan

School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana USA

{mao114, nchimitt, stanchan@purdue.edu}

Abstract

Fast and accurate simulation of imaging through atmospheric turbulence is essential for developing turbulence mitigation algorithms. Recognizing the limitations of previous approaches, we introduce a new concept known as the phase-to-space (P2S) transform to significantly speed up the simulation. P2S is built upon three ideas: (1) reformulating the spatially varying convolution as a set of invariant convolutions with basis functions, (2) learning the basis function via the known turbulence statistics models, (3) implementing the P2S transform via a light-weight network that directly converts the phase representation to spatial representation. The new simulator offers $300\times - 1000\times$ speed up compared to the mainstream split-step simulators while preserving the essential turbulence statistics.

1. Introduction

Despite several decades of research, imaging through atmospheric turbulence remains an open problem in optics and image processing. The challenge is not only in reconstructing images from a stack of distorted frames but also in a less known image formation model that can be used to formulate and evaluate image reconstruction algorithms such as deep neural networks. Simulating images distorted by atmospheric turbulence has received considerable attention in the optics community [29, 3, 11, 24], but using these simulators to develop deep learning image reconstruction algorithms remains a challenge as there is no physically justifiable approach to synthesize large-scale datasets at a low computational cost for training and testing.

Recognizing the demand for a fast, accurate, and open-source simulator, we present a new method to generate a dense-grid image distorted by turbulence with theoretically verifiable statistics. The simulator consists of mostly optics/signal processing steps and a lightweight shallow neural network to perform a new concept called the **Phase-to-Space (P2S)** transform. By parallelizing the computation



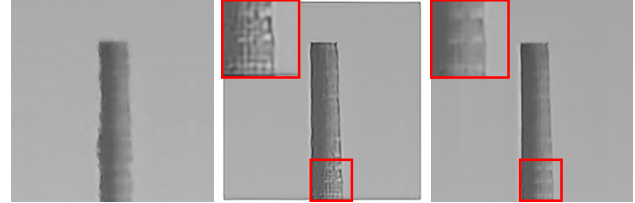
24.36 sec / frame (GPU)

(a) Hardie et al. [11]

0.026 sec / frame (GPU)

(b) Ours

Figure 1. This paper presents a new turbulence simulator that is substantially ($1000\times$) faster than the prior art, while preserving the essential turbulence statistics.



(a) Input (real)

(b) [17]+U-Net

(c) Ours+U-Net

Figure 2. Using our simulator to synthesize training set for training an image reconstruction network (U-Net [28]) offers a considerable amount of improvement in image quality. The network is identical for both (b) and (c); only the simulator used to synthesize the training data is different.

across the pixels, the simulator offers a $1000\times$ speed-up compared to the mainstream approach as shown in Figure 1. When using the new simulator to synthesize training data to train a deep neural network image reconstruction model, the resulting network outperforms the same architecture trained with data synthesized by a less sophisticated simulator, as illustrated in Figure 2.

An overview of the proposed simulator is illustrated in Figure 3. Our proposed approach is based on linking the following two ideas:

- **Convolution via basis functions** (Section 3.1). While conventional approaches model the turbulence distortion as a spatially varying convolution, we reformu-

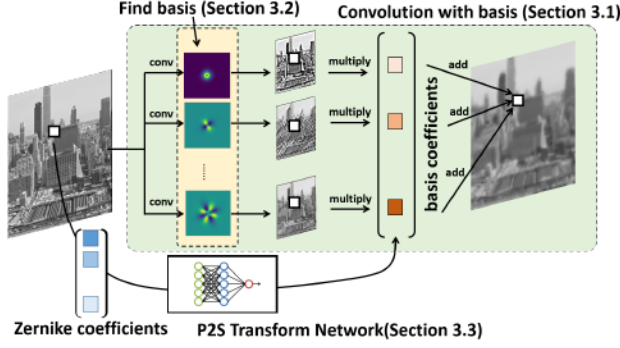


Figure 3. This paper introduces three ideas to significantly speed up the simulation. The three ideas are: (Section 3.1) Approximating the spatially varying convolution by invariant convolutions, (Section 3.2) learning the basis representation via known turbulence statistics, (Section 3.3) implementing the Phase-to-Space transform network.

late the problem by modeling the distortion as a sum of spatially invariant convolutions. The idea is to utilize a basis representation of the point spread functions (PSFs). This concept is similar to the prior work of [23], but in a different context.

- **Learning the basis functions** (Section 3.2). To enable the previous idea, we need to have the basis functions. This is done by utilizing [5] to draw Zernike samples for all *high-order* aberrations. Then, principal component analysis is used to construct the basis functions as proposed by Mao et al. [21]. This is also reminiscent to the dictionary approach proposed by Hunt et al. [13].

The missing piece between these two ideas is the relationship between the basis coefficients in the phase and spatial domains. This is an open problem, and there is no known analytic solution. We circumvent this difficulty by introducing a new concept known as the **Phase-to-Space transform** (Section 3.3). To do so, we construct a lightweight shallow neural network to transform from the phase domain to the spatial domain. Integrating this network into the two aforementioned ideas, our overall simulator adheres to the physics while offering significant speed up and additional reconstruction utility.

2. Background

In this section we provide a brief summary of the turbulence physics and prior work in turbulence simulation. The theory of imaging through atmospheric turbulence can be traced back to the work of Kolmogorov [14] and Tatarski [32], followed by a series of major breakthroughs by Fried [6, 7, 8] and Noll [22]. Readers are encouraged to check out [27, 10] for an introduction.

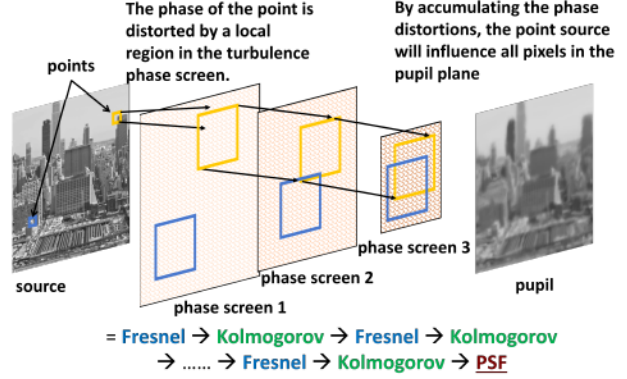


Figure 4. Split-step propagation [11] models the turbulence as a discrete set of phase screens where the wavefront distortion is caused by cropping regions of the phase screen at *every* pixel location. The key operations are Fresnel propagation and Kolmogorov phase imparting. The end result of a sequence of these operations is a PSF for one pixel. The overlaps of the phase screens create the spatial correlations. See [11] for detailed description.

2.1. Split-step simulation

The image formation process through turbulence is best described in the *phase* domain. In free space, an emitted wave propagates spherically outward and, if at a sufficiently long distance, arrives upon the aperture approximately flat. If the medium contains random fluctuations, the phase of the wavefront will be distorted along the path of propagation. We can imagine the wave leading and lagging in phase in reference to its unperturbed counterpart as a result of spatially varying indices of refraction.

The most widely used simulation approach to the above process is the split-step propagation [29, 3, 11]. The idea is to discretize the wave propagation path as illustrated in Figure 4. Split-step simulation propagates every point in the object plane through a discrete set of *phase screens*, alternating between free space propagation, given by Fresnel diffraction, and phase imparting. The statistical behavior of the phase screens is defined through its power spectral density (PSD) [11, 29], many of which are related to the Kolmogorov PSD. This sequence of operations is best described by the equation

$$\text{Fresnel} \rightarrow \text{Kolmogorov} \rightarrow \dots \rightarrow \text{Fresnel} \rightarrow \text{Kolmogorov}.$$

After passing through a turbulent medium, the point spread functions (PSFs) will be spatially varying as illustrated in Figure 5.

The benefit of split-step is two fold: (1) it is interpretable, as it mirrors the physical process, (2) spatial correlations are obtained with minimal effort, as neighboring point sources share overlapping cropped phase screens. The drawback of split-step propagation is its computational requirements: each Fresnel propagation requires a pair of

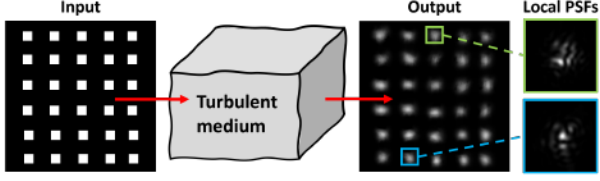


Figure 5. If we image a grid of point sources through turbulence, we will observe a set of spatially varying point spread functions (PSFs). The shape and orientation of the PSFs are determined by the phase structure of the turbulence.

Fourier transforms. This is repeated for every point and every step along the path. Moreover, performing the spatially varying convolution adds another layer of computational cost [11, 29].

2.2. Phase-over-aperture simulation

Our proposed simulator is inspired by the work of Chimitt and Chan [5]. The idea is to collapse the split-step propagation into the resultant phase across the aperture. Compared to split-step which uses *global* phase screens, the collapsed model generates the *local* phase realization directly, which we illustrate in Figure 6.

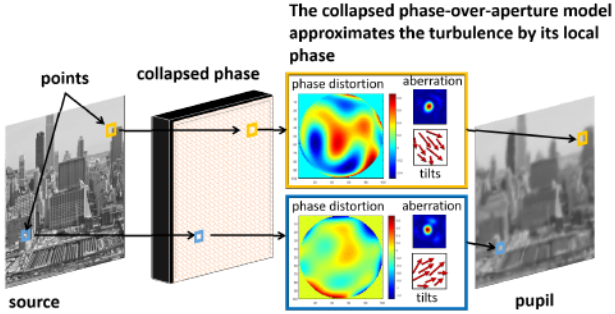


Figure 6. The collapsed phase-over-aperture model [5] replaces the global phase screens and Fresnel diffraction by local phase screens per pixel. This translates the wave propagation to a spatially varying convolution with PSFs that are characterized by the tilts and aberrations *per-pixel*. While the phase cropping and propagation of the split-step method is eliminated, Fourier transforms at every pixel location are still needed.

In the collapsed model, the local per-pixel phase is generated using Noll’s idea [22] that the phase $\phi(\rho)$ (defined over the aperture of diameter D with ρ being the coordinate) can be represented via the Zernike basis functions

$$\phi(\rho) = \sum_{j=1}^K \alpha_j Z_j(\rho), \quad (1)$$

where $Z_j(\rho)$ is the Zernike basis and α_j are the Zernike coefficients $[\alpha_1, \alpha_2, \dots, \alpha_K] = \alpha \sim \mathcal{N}(0, R_Z)$, with [22] providing the expression for R_Z . The resultant incoherent

PSF is formed via

$$h = \left| \mathcal{F} \left\{ W(\rho) e^{-j\phi(\rho)} \right\} \right|^2, \quad (2)$$

omitting a few constants for brevity, with $W(\rho)$ as the pupil function of the aperture.

The Zernike representation offers a natural grouping of terms as suggested in Figure 6: tilt and higher order aberrations. The terms α_2 and α_3 correspond to the horizontal and vertical tilt of the plane of best fit to the phase distortion ϕ . The terms $\alpha_4, \alpha_5, \dots$ correspond to the higher order aberrations and account for the complicated distortions the phase of the wave exhibits. Computationally, these two groups can be separated by generating the high order aberrations, applying the resultant PSFs to the image, then locally shifting the image according to its tilt statistics.

A technical challenge of the collapsed model is ensuring the Zernike coefficients are also *spatially correlated*. In [5], this correlation is enabled through the invention of a multi-aperture approximation in which the correlations could be described analytically by leveraging several classic works [2, 4, 31]. With the correlation matrix defined, the spatially correlated tilts can be generated. For the higher-order terms, it was suggested in [5] that one can define a grid of PSFs and *spatially interpolate* between them.

2.3. Limitations of phase-over-aperture

As reported in [5], the collapsed model is significantly faster than the standard split-step propagation. However, by evaluating the simulator, it is evident that there are several limiting factors:

- The collapsed model exclusively draws Zernike coefficients to create the distortion. However, even with all Zernike coefficients available, one still needs to convert them to PSFs through (2) at *every pixel*. This is the biggest bottleneck.
- It was suggested that in order to reduce the number of Fourier transforms, one can construct the PSFs for a grid of points, then interpolate between them *spatially*. However, mathematically this is incorrect, as the superposition in the spatial domain is not the same as superposition in the phase domain.
- Even if we can resolve the above two problems, to finally simulate a distorted image, we still need to perform the spatially varying convolution. This involves storing the PSFs, and executing the convolution, both of which are resource demanding.

2.4. Other simulators

Ray Tracing. An alternative to the split-step simulation is ray tracing [24, 15], which requires tracing each point sources through the propagation medium. There are also

ray tracing techniques developed in computer graphics [30]. However, the lack of quantitative evaluation based on turbulence statistics makes it difficult to assess these methods.

Warp-and-blur. For faster simulations, one can compromise the accuracy by simulating only the pixel-shifts, commonly referred to as *tilts*, and assuming a spatially-invariant blur [26, 19]. These simulations and models are widely used in the image processing literature [36, 16, 1, 20], where the goal was to provide quick evaluations of the reconstruction algorithms. However, these methods fail to match the known statistical behavior of the distortions.

3. Method

The paper includes two key building blocks: (1) reformulating the spatially varying convolution via a set of spatially invariant convolutions, (2) constructing the invariant convolutions by learning the basis functions. The major invention here is the linkage between the two for which we introduce the P2S transform to convert the Zernike coefficients to the PSF coefficients.

3.1. Idea 1: Convolution via basis functions

The turbulent distortions can be modeled as a spatially varying convolution at each pixel. Denoting $x \in \mathbb{R}^N$ as the source image, and $y \in \mathbb{R}^N$ as the pupil image, the spatially varying convolution says that y is formed by

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} = Hx = \begin{bmatrix} h_1^T x \\ \vdots \\ h_N^T x \end{bmatrix}, \quad (3)$$

where $\{h_n | n = 1, \dots, N\}$ are the N spatially varying PSFs stored as rows of the linear operator $H \in \mathbb{R}^{N \times N}$.

The first key idea of the paper is to write h_n as

$$h_n = \sum_{m=1}^M \beta_{m,n} \varphi_m, \quad (4)$$

for some basis functions φ_m (to be discussed) of the PSFs, and coefficients $\beta_{m,n}$ of the m^{th} basis at the n^{th} pixel. Then, each pixel y_n in (3) can be written as

$$y_n = \sum_{m=1}^M \beta_{m,n} \varphi_m^T x, \quad n = 1, \dots, N. \quad (5)$$

Since convolution is linear, this turns the N spatially varying convolutions $\{h_n^T x\}_{n=1}^N$ in (3) into M spatially invariant convolutions $\{\varphi_m^T x\}_{m=1}^M$ in (5). If $M \ll N$, the computational cost of (5) can be much lower.

To enable the convolution using the basis functions, there are two quantities we need to learn from the data. These are the basis functions φ_m and the coefficients $\beta_{m,n}$. If we

are able to find both, the image can be formed by a simple multiply-add between the basis convolved images $\varphi_m^T x$ and the representation coefficients $\beta_{m,n}$, as illustrated in Figure 3.

3.2. Idea 2: Learning the basis functions

To generate the basis functions φ_m , we consider the process described in [5] of forming a zero-mean Gaussian vector with a covariance matrix R_Z from [22]. The strength of correlation is dictated by the optical parameters as well as the relationship D/r_0 , where D is the aperture diameter and r_0 is the Fried parameter [7]. Figure 7 (the upper half) illustrates the generation of the tilts; removing these does not change the shape of the PSF, but instead centers it. We then seek a basis representation of the resulting centered PSFs, which we show in the lower half of Figure 7.

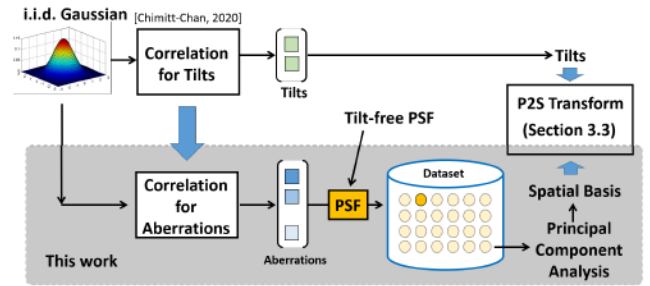


Figure 7. The basis representation is generated in two different ways. For the tilts, we follow the work of [5] to draw spatially correlated tilts by multiplying an i.i.d. Gaussian vector with the tilt correlation matrix. For the high-order aberration terms, we consider the multi-aperture concept of [5] and the analytic solution derived in [31]. Principal component analysis is conducted to extract the spatial basis functions.

To generate the basis functions $\{\varphi_m\}_{m=1}^M$, we use the above procedure to construct a dataset containing 50,000 PSFs from weak to strong turbulence levels. (See supplementary material for details.) Given the dataset, we perform a principal component analysis. For the numerical experiments reported in this paper, a total of $M = 100$ basis functions were used. The basis functions are then combined with the tilts, and are sent to the phase-to-space (P2S) transform to determine the basis coefficients $\{\beta_{m,n}\}$.

3.3. Idea 3: Phase-to-Space (P2S) transform

The third idea, and the most important one, is the phase-to-space transform. The goal is to define a *nonlinear mapping* that converts the per-pixel Zernike coefficients $\alpha = [\alpha_1, \dots, \alpha_K]$ to their associated PSF basis coefficients $\beta = [\beta_1, \dots, \beta_M]$, where we've dropped the pixel index subscript n for notational clarity.

At the first glance, since the basis functions $\{\varphi_m\}_{m=1}^M$ are already found, a straightforward approach is to project

the PSF h (which is defined at each pixel location) onto $\{\varphi_m\}_{m=1}^M$. However, doing so will defeat the purpose of skipping the retrieval of h from the Zernike coefficients as this is the computational bottleneck. One may also consider *analytically* describing the PSF in terms of φ_m and the Zernike coefficients,

$$h = \left| \mathcal{F} \left\{ W(\rho) e^{-j\phi(\rho)} \right\} \right|^2 \stackrel{?}{=} \sum_{m=1}^M \beta_m \varphi_m. \quad (6)$$

However, doing so (i.e., establishing the equality in (6) by writing an equation for β_m) is an open problem. Even if we focus on a special case with just a single Zernike coefficient, the calculation of the basis functions will involve non-trivial integration over the circular aperture [9].

To bypass the complication arising from (6), we introduce a computational technique. The idea is to build a *shallow* neural network to perform the conversion from $\alpha \in \mathbb{R}^K$ to $\beta \in \mathbb{R}^M$. We refer to the process as the phase-to-space transform and the network as the P2S network, as the input-output relationship is from the phase domain to the spatial (PSF) domain.

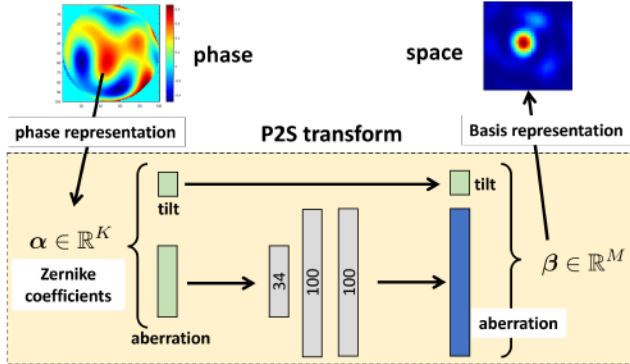


Figure 8. Illustration of the Phase-to-Space transform. We bypass the computationally expensive PSF formation process by a learned mapping between the Zernike and spatial domain. We also note the sizes of the P2S layers here.

A schematic diagram of the P2S transform is shown in Figure 8. Given the two Zernike coefficients representing the tilts and the other Zernike coefficients representing the higher-order aberrations, the P2S transform uses the first two Zernike coefficients to displace the pixels, and uses the network to convert the last $K - 2$ Zernike coefficients to M basis representations.

The architecture of the P2S transform network consists of three fully connected layers as summarized in Figure 8. In terms of training, we re-use the 50,000 PSFs generated for Idea 2 to train the P2S network. The training loss is defined as the ℓ_2 distance between the predicted basis coefficients and the true coefficients (found offline by projecting the PSF onto the learned basis functions). Note that this

network is light-weight because the P2S transform is performed *per pixel*. For an image with a large field-of-view, the P2S network can be executed in parallel. Therefore, even with a 512×512 image, the entire transformation is done in a single pass.

3.4. Interpolation across the grid

We now address the computational difficulty for generating a dense set of Zernike coefficients $\alpha \in \mathbb{R}^K$ for a high resolution image. To accomplish this goal, we partition the image into a user-defined grid of anchor points, for example, a 64×64 grid. This grid corresponds to a correlation matrix of size $64^2 \times 64^2 = 4096 \times 4096$ which can be pre-computed. Following Figure 7, 4096 sets of Zernike coefficients are drawn from the correlation matrix. To go from the grid of 64×64 anchor points to the full image, we interpolate the Zernike coefficients using bilinear interpolation.

For generation of the anchor points, we implement the angle-of-arrival statistics according to [31], in conjunction with [2, 4]. The process is mathematically tedious but conceptually simple: One just needs to rewrite the entries of the correlation matrix in [5] with the formula provided by [31]. The output of the new correlation matrix is a set of spatially correlated Zernike coefficients.

It is important to emphasize the difference between the way we interpolate and the interpolation used in [5]. In [5], the interpolation is performed in the spatial domain where two PSFs are superimposed to generate a new PSF. In our simulator, we interpolate the Zernike coefficients to superimpose two phase functions. If the phase ϕ and the PSF h is related by the P2S transform, $\phi \xleftrightarrow{\text{P2S}} h$, it is important to note that for any $0 \leq \lambda \leq 1$,

$$\lambda \phi_1 + (1 - \lambda) \phi_2 \not\xrightarrow{\text{P2S}} \lambda h_1 + (1 - \lambda) h_2.$$

Therefore, the interpolation used in [5] is less justifiable. In Figure 9 we illustrate the two interpolation schemes. We have selected a realistic and easily-observable case for illustration in which interpolation in the Zernike spaces generates a near-diffraction-limited PSF (the lucky effect [8]) but in the spatial domain is missed.

3.5. Extension to color images

Most deep neural networks today are designed to handle color images. To ensure that our simulator is compatible with these networks, we extend it to handle color.

In principle, the spectral response of the turbulent medium is wavelength dependent, and the distortion must be simulated for a dense set of wavelengths. However, if the turbulence level is moderate, wavelength-dependent behavior of the Fried parameter is less significant for the visible spectrum (roughly 400nm to 700nm) when compared to other factors of the turbulence.

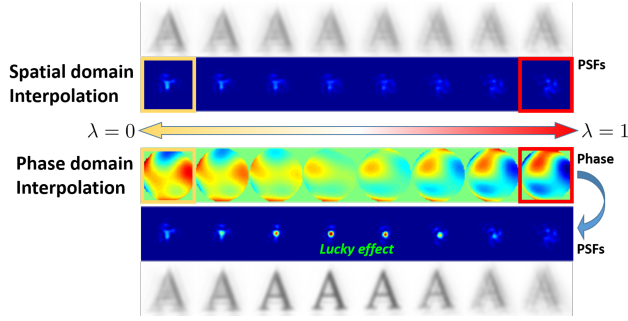


Figure 9. Comparison between the spatial interpolation scheme from [5] and our interpolation in the phase domain. For both cases, we show the PSFs and example resultant images. [Top] Spatial interpolation of two PSFs is performed via $\lambda h_1 + (1 - \lambda)h_2$, which is a superposition of the two PSFs. [Bottom] Phase interpolation is performed via $\lambda \phi_1 + (1 - \lambda)\phi_2$. In this example, the superposition of the two phase functions will lead to a PSF with very mild phase distortion known as a lucky observation [8]. This lucky observation is absent in the spatial domain interpolation.

To illustrate this observation, we show in Figure 10 the individual PSFs for several wavelength from 400nm (blue) to 700nm (red). It is evident that the shape of the PSFs barely changes from one wavelength to another. In the same figure, we simulate two color images. The first image is simulated by using a single PSF (525nm) for the color channels (and displayed as an RGB image). The second image is simulated by considering 3 PSFs with wavelengths 450nm, 540nm, and 570nm. We note that (c) is a more realistic simulation but requires $3 \times$ computation. However, the similar PSFs across the color makes difference is visually indistinguishable, as seen in (d). The small gap demonstrated in Figure 10 suggests that we can simulate the RGB channels identically in such conditions.

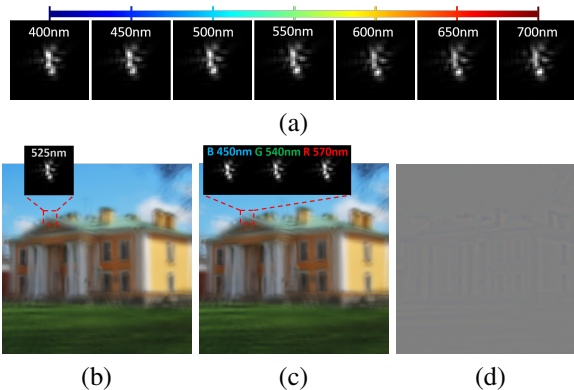


Figure 10. (a) PSFs across the visible spectrum. (b) Same distortion applied to three channels using center wavelength of the visible spectrum. (c) Wavelength dependent distortions applied to three channels. (d) Error map between (b) and (c).

4. Experimental Evaluation

Our experimental results consist of four parts: (i) Quantitative evaluation based on known turbulence statistics, (ii) visual comparison with real turbulence data, (iii) impact to deep neural network image reconstruction methods, (iv) run time comparison. Additionally, videos are included in the supplementary materials.

4.1. Quantitative evaluation

Evaluation schemes. In the turbulence simulation literature, there are two standard ways to quantitatively evaluate a simulator: (i) the Z-tilt and the differential tilt statistics, and (ii) the short and long exposure statistics. For a simulator to be valid, it is necessary to match the simulated data with the theoretical curves.

Turbulence conditions. To conduct this evaluation, we follow a similar setting as [5] and [11]. The parameters of the turbulence are listed in the supplementary material.

Evaluation 1: Tilt statistics. We first report the Z-tilt and the differential-tilt statistics. The Z-tilt and the differential-tilt statistics measure tilt correlation across the angle-of-arrivals. For example, the Z-tilt should drop as the angle-of-arrival increases, because two pixels that are far apart should have less (but non-zero) correlation. The results of the Z-tilt and the differential-tilt are shown in Figure 11. It is evident that the tilt statistics of the proposed simulator matches well with the theoretical predictions.

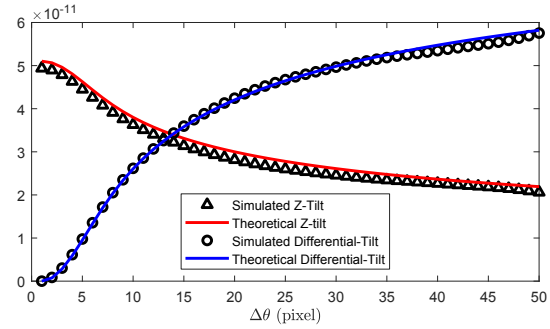
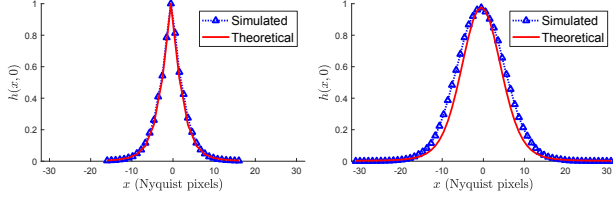


Figure 11. The Z-tilt and differential-tilt statistics produced by our simulator match with the theoretical values.

Evaluation 2: Long and short exposure. We also analyze the long and short exposure (LE and SE, respectively) behavior of the generated PSFs. The LE PSF is a standard temporal average over the PSF realizations, while the SE is a temporal average over the centered PSFs. Since the LE includes pixel shifts, the spread of the LE PSF is larger than its SE counterpart. Furthermore, the SE is a valuable metric as it quantifies the blur the system experiences regardless of its shift behavior. We present these results in Figure 12, where we again see a match between the simulated and theoretical behavior.



(a) Short exposure

(b) Long exposure

Figure 12. The short and long exposure PSFs produced by our simulator match with the theoretical PSFs.

4.2. Visual comparison with real data

We emphasize the results of the previous quantitative discussion are significant statistically. However, visual comparisons with real data, while subjective, are an important consideration and serve as a useful reality check. In the following discussion, we present data simulated at the same optical parameters as those provided and show their real counterparts for visual comparison.

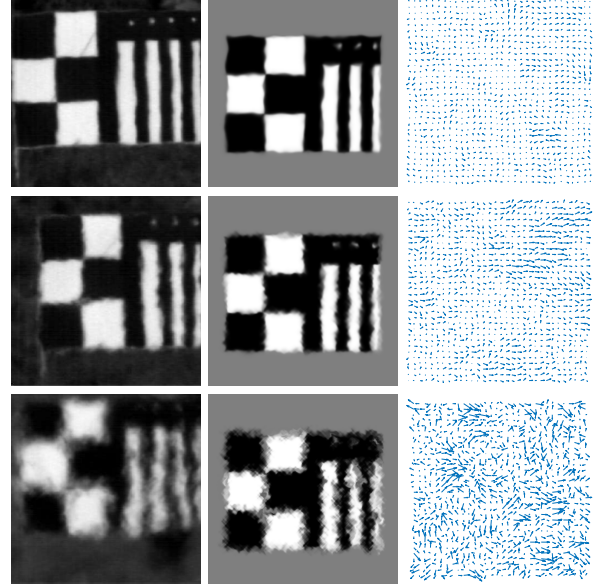
NATO dataset. The carefully recorded NATO RTG40 dataset [25, 33] contains both optical and estimated turbulence parameters. For these particular sets of images, the target is 1 km from the imaging system using passive visible light for imaging. Turbulence parameters were measured that help to evaluate what the appropriate turbulence level was at the time of taking the images. We select these parameters to use in our simulation technique, with comparisons shown in Figure 13.

In comparing simulated against their real counterparts, we can see a match in blur and shifting effects. At higher turbulence levels, there are some small observable differences, though we argue this is inherent to modeling just the phase in this type of problem as well as differences in illumination (e.g. digital representation of target pattern vs. illumination by the sun).

Datasets used in [12] and [1]. In addition to the NATO dataset, there are also those used in [12] and [1]. The images in Figure 14 show a method of collecting turbulence data that uses stream of gas in front of the camera to produce images at different turbulence levels. While this is a different scenario than the typical long-distance imaging sequences, this data serves as a decent proxy and is useful as it is easier to collect and can provide ground truth by simply turning the gas system off. We present for visual comparison the results in Figure 14 and note the similarity in random draws vs. observations.

4.3. Impact on training deep networks

We conduct an experiment to demonstrate the impact of the proposed simulator on a multi-frame turbulence image reconstruction task. The goal of this experiment is to show that a deep neural network trained with the data synthesized by the proposed simulator outperforms the same network



(a) real

(b) simulated

(c) tilt map

Figure 13. Contrast balanced NATO RTG-40 dataset reported by [25, 33]. The optical parameters are listed in supplementary materials.

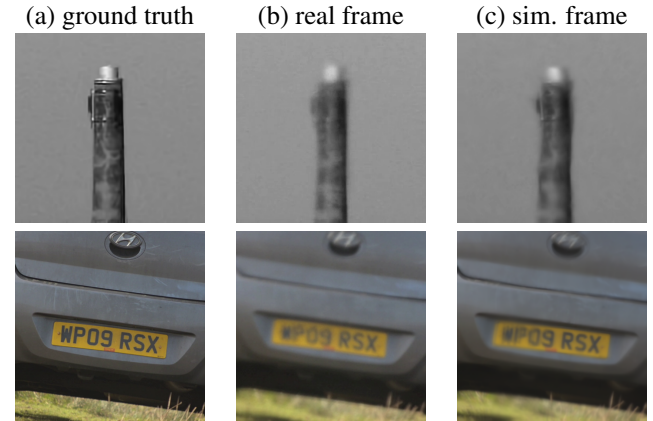


Figure 14. Visual comparison of simulated and real turbulence data. With comparing individual frames, we can see similar blurring and warping effects.

trained with the data generated by simulators that are less physically justified.

To demonstrate the impact of the simulator, we do not use any sophisticated network structure or training strategy. Our network has a simple U-Net architecture [28] with 50 input channels and is trained with an MSE loss for 200 epochs. The network is trained with 5000 simulated sequences, where each sequence contains 50 degraded frames. The ground truth images used for simulation are obtained from the Places dataset [35]. The sequences are simulated with a turbulence level D/r_0 uniformly sampled from [1, 8].

For comparison, we train the same network using a simu-

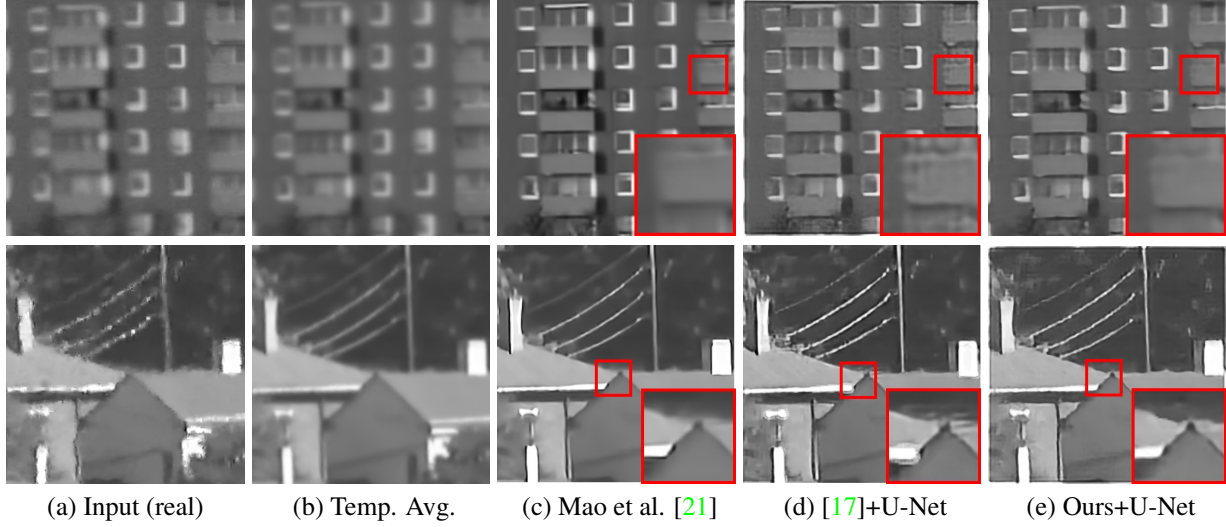


Figure 15. Image reconstruction using **real** data (so ground truth is not available). For (d) and (e), we train a UNet using data synthesized by [17] and our simulator, respectively. Notice the artifacts in (d).

D/r_0	Mao et al. [21]	Ours+U-Net	[17]+U-Net
1.5	27.33dB	27.18dB	26.59dB
3.0	27.04dB	26.98dB	26.11dB
4.5	25.85dB	26.01dB	25.40dB

Table 1. PSNR values of the reconstruction results, averaged over 30 testing sequences. The testing data is **synthesized** by the split-step propagation method [11].

lation technique proposed by Lau et al. [17]. This simulator has been used in several recent works [18, 34]. To ensure a fair comparison, we perform a uniform sweep for the Gaussian blur (σ^2 sampled from [1, 3]) and tilt strength (sampled from [0.1, 0.4]). As a reference, we also report the results of a deterministic (non-learning based) state-of-the-art reconstruction method by Mao et al. [21].

Two qualitative reconstruction results are shown in Figure 15. It can be seen that the network trained with proposed simulator has performance close to state-of-the-art. Visible artifacts are generated from the network trained with [17]. We also include a quantitative evaluation, where a split-step simulator [11] is used to generate 30 testing sequences under low, medium, and high ($D/r_0 = 1.5, 3$, and 4.5). PSNR values are reported in Table 1. It is worth nothing that the network trained with the data synthesized by our simulator achieves a comparable performance to the state-of-the-art.

4.4. Run time

Finally, we compare the run time of the proposed method with several existing methods [5, 11, 17]. The simulators are run on a computing cluster node with Intel Xeon “Sky Lake” processors (16 cores) and a Tesla V100 GPU. We use 16×16 PSF grid for [5], which is comparable to our initial PSF grid. The for-loop in [17] is executed 1000 times as suggested by the authors. The run time of [11] is reported

Reference	Method	CPU (s)	GPU (s)
Hardie et al. [11]	split-step	119.63	24.36
Chimitt-Chan [5]	collapsed	5.88	N/A
Lau et al. [17]	subsampling	3.13	N/A
Ours	P2S	0.35	0.026

Table 2. Average run time for each method to process a 256×256 frame. Unit are in seconds.

by the authors. The required time to process a 256×256 frame is reported in Table 2. The proposed method offers $300 \times -1000 \times$ speed up compared to Hardie et al. [11].

5. Conclusion

The simulation approach towards imaging through atmospheric turbulence we have presented in this work has desirable advantages over existing methods. The key innovation of the P2S transform network allows for significant speedup and additional reconstruction utility. With respect to deep-learning based reconstruction, the outlined approach allows for the generation of large amounts of training data not previously feasible. Additionally, the ability to use the simulation approach as a differentiable module in a neural-network suggests additional benefit towards reconstruction. Finally, we expect the ability to produce statistically accurate data far more efficiently will allow for further statistical analysis of turbulent imaging properties through numerical analysis methods not previously possible.

Acknowledgement

The work is supported, in part, by the National Science Foundation under the grants CCF-1763896 and ECCS-2030570.

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, June 2013. 4, 7
- [2] Santasri Basu, Jack E. McCrae, and Steven T. Fiorino. Estimation of the path averaged atmospheric refractive index structure constant from time lapse imagery. In *Proc. SPIE 9465, Laser Radar Technology and Applications XX; and Atmospheric Propagation XII*, pages 1–9, May 2015. 3, 5
- [3] Jeremy P. Bos and Michael C. Roggemann. Technique for simulating anisoplanatic image formation over long horizontal paths. *Optical Engineering*, 51(10):1 – 9 – 9, 2012. 1, 2
- [4] Gary A. Chanan. Calculation of wave-front tilt correlations associated with atmospheric turbulence. *Journal of Optical Society of America A*, 9(2):298–301, Feb. 1992. 3, 5
- [5] Nicholas Chimitt and Stanley H. Chan. Simulating anisoplanatic turbulence by sampling intermodal and spatially correlated Zernike coefficients. *Optical Engineering*, 59(8):1 – 26, 2020. 2, 3, 4, 5, 6, 8
- [6] David L. Fried. Statistics of a geometric representation of wavefront distortion. *Journal of the Optical Society of America*, 55(11):1427–1435, Nov. 1965. 2
- [7] David L. Fried. Optical resolution through a randomly inhomogeneous medium for very long and very short exposures. *Journal of Optical Society of America*, 56(10):1372–1379, 1966. 2, 4
- [8] David L. Fried. Probability of getting a lucky short-exposure image through turbulence*. *J. Opt. Soc. Am.*, 68(12):1651–1658, Dec 1978. 2, 5, 6
- [9] Joseph W. Goodman. *Introduction to Fourier Optics*. Roberts and Company, Englewood, Colorado, 3 edition, 2005. 5
- [10] Joseph W. Goodman. *Statistical Optics*. John Wiley and Sons Inc., Hoboken, New Jersey, 2 edition, 2015. 2
- [11] 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):1 – 16, 2017. 1, 2, 3, 6, 8
- [12] Michael Hirsch, Suvrit Sra, Bernhard Schölkopf, and Stefan Harmeling. Efficient filter flow for space-variant multi-frame blind deconvolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 607–614, June 2010. 7
- [13] Bobby R. Hunt, Amber L. Iler, Christopher A. Bailey, and Michael A. Rucci. Synthesis of atmospheric turbulence point spread functions by sparse and redundant representations. *Optical Engineering*, 57(2):1–11, Feb. 2018. 2
- [14] Andrei N. Kolmogorov. The Local Structure of Turbulence in Incompressible Viscous Fluid for Very Large Reynolds' Numbers. *Akademiia Nauk SSSR Doklady*, 30:301–305, 1941. 2
- [15] Svetlana L. Lachinova, Mikhail A. Vorontsov, Grigorii A. Filimonov, Daniel A. LeMaster, and Matthew E. Trippel. Comparative analysis of numerical simulation techniques for incoherent imaging of extended objects through atmospheric turbulence. *Optical Engineering*, 56(7):071509.1–11, 2017. 3
- [16] Chun Pong Lau, Yu Hin Lai, and Lok Ming Lui. Restoration of atmospheric turbulence-distorted images via RPCA and quasiconformal maps. *Inverse Problems*, Mar. 2019. 4
- [17] Chun Pong Lau and Lok Ming Lui. Subsampled turbulence removal network. *Mathematics, Computation and Geometry of Data*, 1(1):1–33, 2021. 1, 8
- [18] Chun Pong Lau, Hossein Souri, and Rama Chellappa. At-facegan: Single face semantic aware image restoration and recognition from atmospheric turbulence. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, pages 1–1, 2021. 8
- [19] Kevin R. Leonard, Jonathan Howe, and David E. Oxford. Simulation of atmospheric turbulence effects and mitigation algorithms on stand-off automatic facial recognition. In *Proc. SPIE 8546, Optics and Photonics for Counterterrorism, Crime Fighting, and Defence VIII*, pages 1–18, Oct. 2012. 4
- [20] Yifei Lou, Sung Ha Kang, Stefano Soatto, and Andrea Bertozzi. Video stabilization of atmospheric turbulence distortion. *Inverse Problems and Imaging*, 7(3):839–861, Aug. 2013. 4
- [21] 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. 2, 8
- [22] Robert J. Noll. Zernike polynomials and atmospheric turbulence. *J. Opt. Soc. Am.*, 66(3):207–211, Mar 1976. 2, 3, 4
- [23] Timothy Popkin, Andrea Cavallaro, and David Hands. Accurate and efficient method for smoothly space-variant gaussian blurring. *IEEE Transactions on Image Processing*, 19(5):1362–1370, 2010. 2
- [24] Guy Potvin, Luc Forand, and Denis Dion. A simple physical model for simulating turbulent imaging. In *Proceedings of SPIE*, volume 8014, pages 80140Y.1–13, 2011. 1, 3
- [25] Endre Repasi and Robert Weiss. Analysis of image distortions by atmospheric turbulence and computer simulation of turbulence effects. *Proc SPIE*, 6941, 05 2008. 7
- [26] Endre Repasi and Robert Weiss. Computer simulation of image degradations by atmospheric turbulence for horizontal views. In *Proc. SPIE 8014, Defense, Security, and Sensing*, May 2011. 4
- [27] Micheal C. Roggemann and Byron M. Welsh. *Imaging through Atmospheric Turbulence*. Laser & Optical Science & Technology. Taylor & Francis, 1996. 2
- [28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. 1, 7
- [29] Jason D. Schmidt. *Numerical simulation of optical wave propagation: With examples in MATLAB*. SPIE Press, Jan. 2010. 1, 2, 3

- [30] Armin Schwartzman, Marina Alterman, Rotem Zamir, and Yoav Y. Schechner. Turbulence-induced 2D correlated image distortion. In *Proc. International Conference on Computational Photography*, pages 1–12, 2017. [4](#)
- [31] Naruhisa Takato and Ichirou Yamaguchi. Spatial correlation of Zernike phase-expansion coefficients for atmospheric turbulence with finite outer scale. *Journal of Optical Society of America A*, 12(5):958–963, May 1995. [3](#), [4](#), [5](#)
- [32] Valeryan I. Tatarski. *Wave Propagation in a Turbulent Medium*. New York: Dover Publications, 1961. [2](#)
- [33] David Tofsted, Sean O’Brien, Jimmy Yarbrough, David Quintis, and Manuel Bustillos. Characterization of atmospheric turbulence during the NATO RTG-40 land field trials. In Cynthia Y. Young and G. Charmaine Gilbreath, editors, *Atmospheric Propagation IV*, volume 6551, pages 199 – 208. International Society for Optics and Photonics, SPIE, 2007. [7](#)
- [34] Rajeev Yasarla and Vishal M. Patel. Learning to restore a single face image degraded by atmospheric turbulence using cnns. *ArXiv*, abs/2007.08404, 2020. [8](#)
- [35] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. [7](#)
- [36] Xiang Zhu and Peyman Milanfar. Removing atmospheric turbulence via space-invariant deconvolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):157–170, Jan. 2013. [4](#)