Simulation of Anisoplanatic Turbulence for Images and Videos

D, Vint, G. Di Caterina, P. Kirkland
Centre for Signal & Image Processing
University of Strathclyde
Glasgow, United Kingdom
{david.vint, gaetano.di-caterina, paul.kirkland}@strath.ac.uk

R. A. Lamb, D. Humphreys
Airborne & Space Systems Division
Leonardo MW Ltd
Edinburgh, United Kingdom
{david.humphreys, robert.lamb}@leonardocompany.com

***Abstract*—Turbulence is a common phenomenon in the atmosphere and can generate a variety of distortions in an image. This can cause further image processing tasks to struggle due to lack of detail in the resulting turbulence affected imagery. It is therefore useful to attempt to remove such distortions as a post processing step. However, the development of such algorithms is difficult due to the complex nature of turbulence data acquisition. To alleviate these issues, this paper presents the development of a turbulence simulator that is capable of imparting the effects of a turbulent atmosphere onto clean images and videos. This work also provides a large, publicly available dataset that can be used as a benchmark. The simulator and dataset will be valuable resources in the field of turbulence mitigation. Indeed, the simulator allows researchers to simulate specific turbulent conditions for any application as required; while the dataset provides the ability to make use of turbulent data without the expensive time commitment of simulation.**

***Index Terms*—Atmospheric Turbulence, Simulation, Anisoplanatic, Wave propagation, Dataset**

I. Introduction

Atmospheric turbulence is caused by random perturbations in the refractive index of air [1], which causes light to divert from its intended path, resulting in angle of arrival fluctuations and phase alterations. This results in image warping and blurring. The processing of such a degraded image can therefore be rather challenging, as the combined warping and blurring results in a loss of image clarity and high frequency detail. Therefore, a common post-acquisition step consists in the attempt to recover such detail [2]–[4]. However, in order to design such post processing algorithms, example data is needed. Very often such post processing methods require not only the turbulence affected images, but also the clean, turbulent free images as reference. This is especially true for the case of deep learning approaches, as paired data is necessary for training purposes. The practical acquisition of such a dataset however is an extremely difficult task, as the lack of control over imaging conditions makes the capture of corresponding ground truth images nearly impossible. Such a ground truth image would need to be acquired when turbulence is not present, likely at a different time of day. The challenge then becomes the spatial alignment of the camera such that the exact same image is taken.

D.V. acknowledged support from the UK EPSRC and Leonardo MW Ltd, Edinburgh

(a) Original Image (b) $C_n^2 = 0.25\text{x}10^{-15}$ (c) $C_n^2 = 1.5\text{x}10^{-15}$

Fig. 1: Simulation sample images.

The ability to simulate the effects of turbulence in software is therefore desirable, as any atmospheric conditions can be evaluated, simulated, and applied to an image. This paper presents the development and implementation of such a simulator for generating anisoplanatic turbulent imagery. Based on the works Schmidt [5] and Hardie et al. [6], the simulator models the turbulent atmosphere as a series of phase screens. A point source is then propagated through the screens to provide a Point Spread Function (PSF), which describes how the point source has spread throughout the atmosphere. This is then applied to a clean image to provide the final turbulent output, examples of which can be seen in Fig.1. The simulator presented in this paper is able to produce turbulent images as well as videos where, for video sequences, the speed and direction of the turbulence can also be defined. This work also presents a novel dataset generated through the proposed simulator. This dataset is available for use by the general public and will be a useful resource for future research on turbulence mitigation algorithms [1].

The paper is organised as follows. Section II provides related works in the field of turbulence simulation. Section III-A details the underlying theory of turbulence and its characterisation. Section III-B then goes into the details of the simulator, describing the generation of the phase screens (III-B1) and the propagation process (III-B2). Section III-C describes the dataset presented with this work, with details

[1]All data underpinning this publication are openly available from the University of Strathclyde KnowledgeBase at https://doi.org/10.15129/1adfbe5c-68f0-49f1-9bad-e64872f9f582. The code for the simulator is available on request

This is a peer-reviewed, accepted author manuscript of the following paper: Vint, D., Di Caterina, G., Kirkland, P., Lamb, R. A., & Humphreys, D. (Accepted/In press). Simulation of anisoplanatic turbulence for images and videos. Paper presented at Sensor Signal Processing for Defence 2023, Edinburgh, United Kingdom.

on the simulator settings. Finally, Section IV provides the conclusion to the paper.

II. Related Works

When simulating the effects of turbulence, there is a trade-off between simulation accuracy and processing time. To ensure accurate simulation, a large number of calculations are needed, therefore increasing simulation time. Other simulation methods instead make use of prior knowledge regarding turbulent theory that allows an estimation of turbulence. These implementations are faster, however they have limited accuracy. The distortion of an image can be described by

$$I_{out}(x,y) = H\left[I_{in}(x,y)\right] + n(x,y) \tag{1}$$

where I_{in} and I_{out} are the clean input and distorted output images respectively. Here n is additive noise and $H[\cdot]$ is a function that describes the type of distortion. In the case of turbulent imagery, the function $H[\cdot]$ can be represented by a spatially varying warp, as well as a blurring operation [7], [8]. The simplest method of turbulence simulation can be achieved by applying (1) as a set of random functions, where the distortion function can be represented by a random shift of pixels, and the noise is represented by Gaussian blur [9]. The most common approach to the simulation of turbulence is that of estimating the PSF caused by a turbulent atmosphere [7], [10]. This can be done using an optical transfer function (OTF), which describes how a source of light is affected by the turbulent atmosphere and camera system [11].

Whilst the theoretical OTF can provide an indication of a turbulent volume, the most accurate method of depicting the path of light is by implementing a propagation simulation [12]. These methods model the atmosphere using a series of complex planes called phase screens, which represent how the light wave changes path as it travels [13], [14]. Although the propagation of a single light wave can be achieved with minimal computation, the aggregate time required to perform a propagation for each pixel in an image can result in a computationally expensive simulation.

As previously described, the practical acquisition of turbulent data is a challenging task. However, by combining practical acquisition with computer simulation, real data can be utilised. A popular method of simulation is the work of Tunick et al. [15], [16], which was able to extract typical distortions caused by different levels of turbulence. This allows their simulation method to draw upon real turbulent experience when processing an image. This method has also been used in [17], [18]. Computer simulation is not the only method of recreating turbulence. Multiple works have made use of real life phenomena and real time photo capture in order to obtain turbulent data. This includes imaging through a heating vent [19], gas hobs [3] and hair dryers [20]. Whilst these methods introduce a form of control over the underlying turbulent effects, they are not true simulations of turbulence over a long propagation path and are therefore only a simplistic imitation of the real effects of turbulence.

Despite the challenges, attempts have been made to produce a real turbulence dataset, such as the work of Gilles et al. [21], which formulates ground truth images via a downsampling and registration process of the pristine image. Anantrasirichai et al. [3], [22] have collated real data for the purpose of turbulence mitigation. To overcome the lack of ground truth, a no reference image quality metric is used to evaluate the performance of their methods. Turbulent data can also be acquired for the purpose of atmosphere analysis. Such as in [1], [23], where scintillometers are used to provide a reference value of C_n^2 for the current atmosphere.

III. Turbulence Simulation

A. Turbulence Theory

Any viscous fluid can be categorised into one of two states of motion: laminar or turbulent. In the case of laminar flow, no mixing of the fluid occurs, resulting in a predictable flow that has known characteristics. The flow becomes turbulent when mixing occurs, causing the flow to break into subflows called turbulent eddies, where any predictable characteristics are therefore lost [24]. The point at which a fluid can transition from laminar to turbulent flow can be determined by the Reynolds number, Re, defined as

$$Re = \frac{Vl}{v}$$

where V, l and v are the velocity, length and kinematic viscosity of the fluid respectively. At low values of the Reynolds number, fluids tend towards laminar flow, whilst high Reynolds numbers characterise turbulent flow [24]. By considering the atmosphere as a fluid, the same principles can be applied in the case of turbulence modelling.

Once a fluid transitions from laminar into turbulent flow, the resulting turbulent subflows can then be described as an energy cascade [25]. This begins with large eddies forming due to an injection of energy; these then proceed to break up into smaller eddies and continue to reduce until the eddies dissipate completely as heat. This cascade begins at a size L_0 and reduces in size to l_0. These two values are known as the inner and outer scales of turbulence. The eddies that lie within these two scales form what is known as the 'inertial subrange' [24]. Within this inertial subrange, assumptions can be made with regard to the statistics of the atmospheric field, which led Kologorov to build his statistical model of turbulence [26]. By using these assumptions and dimensional analysis, Kolmogarov derived a power spectral density (PSD) for the changes in refractive index in air:

$$\Phi_n^K(\kappa) = 0.033 C_n^2 \kappa^{-11/3}, \quad 1/L_0 \ll \kappa \ll 1/l_0 \tag{2}$$

where κ is the angular spatial frequency in rad/m and C_n^2 is the refractive index structure parameter. Typical values of C_n^2 range from 1×10^{-16} (weak) to 1×10^{-13} (strong). Other models for the refractive PSD introduce additional control parameters to better align the theoretical and analytical experiments [5]. Such a model is the modified Von Karman

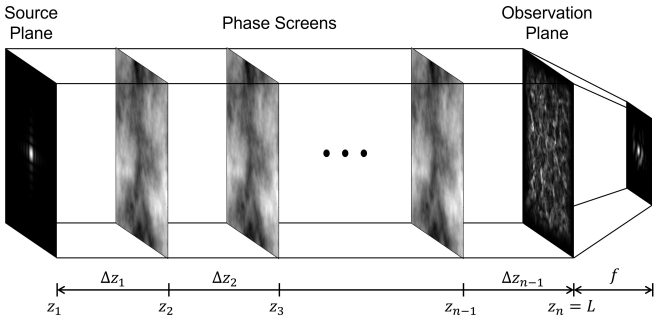


Fig. 2: Illustration of a single propagation. The point source is propagated through each phase screen until the observation plane is reached. The lens operation then converts the complex plane into a Point Spread Function.

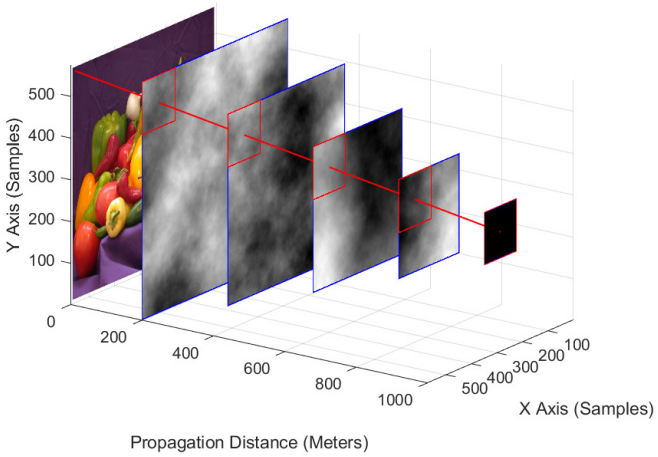


Fig. 3: Geometry of 3D space in which phase screens are situated. For each pixel, a pencil ray is traced towards the observation plane. The intersections of the pencil rays correspond to the centre points for the cropped screens used for propagation.

PSD, which is the PSD used for this simulator. It is evaluated as

$$\Phi_{\phi_i}^{mvK}(f) = \frac{0.023\mathrm{e}^{-f^2/f_m^2}}{r_{0_i}^{5/3}\left(f^2+f_0^2\right)^{11/6}} \tag{3}$$

where f is the angular spatial frequency in cycles/m. $f_m = 5.92/2\pi l_0$ and $f_0 = 1/L_0$. Unlike (2), this is evaluated with respect to the Fried parameter of the i^{th} screen, r_{0_i}, which is a measure of optical transmission quality [27].

B. Simulator Model

To simulate the effects of a turbulent atmosphere on images and videos, the atmosphere must first be modelled in 3D space. A point source can then be propagated from the source plane, along a distance L until it reaches the camera sensor (Observation Plane). To represent a volume of turbulent atmosphere, it is common to treat the atmosphere as a series of discrete layers [5], where each layer is represented by a 2 dimensional phase screen. The result of propagation through these phase screens is a complex matrix at the observation plane. Using a lens operation, this complex plane is transformed into a single PSF [6], which can then be applied to an image via convolution. This process of propagation is illustrated in Fig.2. The propagation of a single point source through phase screens is known as isoplanatic simulation. In such a case, the resultant PSF is applied to each pixel within the source image (i.e. spatially invariant). This therefore assumes that each pixel in the image has passed through the same volume of turbulence (or that the turbulence is identical in all points in the 3D space). This however is not the case in real imagery as each source of light traces a different path through the atmosphere. The modelling of such an environment is known as anisoplanatic simulation, in which each pixel has a specific PSF based on its optical path through the atmosphere. To achieve this, the phase screens are generated at an extended size, as seen in Fig.3. Once the trajectory of a pixel (pencil ray) is traced through 3D space, the intersections with the phase screens are taken as centre points for a cropping operation. The cropped screens are then used for propagation, as in Fig.2. For the simulation of video sequences, the phase screens are again generated to an extended size. The video is then simulated one frame at a time, where the phase screens are translated laterally, to simulate turbulence motion, by a number of samples in a given direction for each new frame, resulting in frames that are temporally correlated.

In order to sample the planes such that the simulation is accurate, the actual physical dimensions of the simulation are used. In the case of this simulator, the size of the planes is related to the diameter D of the camera aperture. From this, the width X of the point source and cropped phase screens is defined as $X = sD = \Delta x N$, where s is a scaling parameter, Δx is the grid spacing and N is the sample count. Δx is calculated such that it is able to accurately represent each screen without undersampling. To ensure this, the Voel critical sampling rule is applied [5], as

$$\Delta x = \left(\frac{\lambda L}{N}\right)^{\frac{1}{2}} \tag{4}$$

allowing the sample count N to be evaluated as

$$N = \frac{X^2}{\lambda L} \tag{5}$$

The scaling parameter s is chosen such that the resulting value of N is a power of two and the screen width is at least 4 times that of the aperture diameter. The image at the source plane is sampled according to Nyquist as $\delta_o = \lambda L/(2D)$ [6].

1) Generation of Phase Screens: Each phase screen imparts an optical phase change to the incoming wave. Assuming that this phase is a Fourier transformable function, it can be written as a Fourier series [5] as

$$\phi(x,y) = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} \hat{c}_{n,m} e^{i2\pi(f_{x_n}x + f_{y_m}y)}$$

where f_{x_n} and f_{y_m} are the spatial frequencies. $\hat{c}_{n,m}$ is the random realisation of the Fourier series coefficients $c_{n,m}$, which are multiplied by a Gaussian random variable with zero mean and unit variance as

$$\hat{c}_{n,m} = \mathcal{N}\left(f_{x_n}, f_{y_m} \mid 0, 1\right) c_{n,m}$$

where the values of $c_{n,m}$ are generated from the modified von Karman PSD (3) as

$$c_{n,m} = \sqrt{\Phi_{\phi_i}\left(f_{x_n}, f_{y_m}\right)}\Delta f$$

where the frequency spacing is $\Delta f = 1/(N\Delta)x$. For the evaluation of (3), the values of r_{0_i} are first calculated using the optimisation method described in [6]. The limitation of this method is that of sampling the modified von Karman PSD at Δf, as most of the power lies at the low spatial frequencies. To access these frequencies, a larger spatial sampling rate, Δx, would be required. To overcome this, more phase screens are generated at subharmonics of Δf ($\Delta f_p = \Delta f/3^p$). These are then combined with the base screen, resulting in accurate phase representations of the turbulent atmosphere.

In the case of anisoplanatic simulation, (4) and (5) are evaluated with respect to a single propagation, resulting in a cropped screen size of X. The extended screens have a width of $\tilde{X}_i = \Delta x \tilde{N}_i$, where $\tilde{N}_i$ is evaluated based on the position of the screen along the propagation path. The largest of which (i.e. closest to the source plane) is sampled such that it is N samples larger than the source image (sampled at δ_o) for the accommodation of the corner pencil rays. Note that the PSDs for these extended screens are sampled at a frequency of $\Delta f = 1/\Delta x \tilde{N}$. An example phase screen can be seen in Fig.4a.

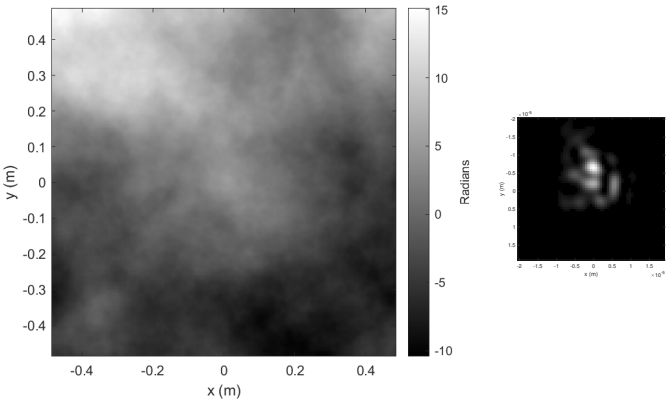
2) Propagation and Image Generation: The point source used for propagation is modelled as a 2D Gaussian windowed sinc function [5], defined as

$$U_{pt}\left(x, y\right) = \lambda L \alpha^2 e^{-i\frac{k}{2L}\left(x^2+y^2\right)} sinc\left[\alpha x, \alpha y\right] e^{-\frac{\alpha^2}{16}\left(x^2+y^2\right)}$$

where $\alpha = (4D)/(\lambda L)$. This source is designed such that if propagated through a turbulent free atmosphere, the result is a $4D$x$4D$ patch of uniform amplitude on the observation plane.

Split-step propagation of the point source is performed using the Fresnel diffraction equation; for the purposes of this paper, the derivation of this equation is omitted, and the reader is redirected to [5]. This equation begins with the point source and propagates it to the first phase screen, at which point the phase is altered as defined. This process is then repeated until the observation plane is reached. This is known as split-step propagation. An attenuating border is also introduced to the phase screens, in order to reduce any signal energy that is tending towards the simulation boundary. The split-step propagation produces a complex field $U_0\left(x, y\right)$ at the observation plane. This is then focused at the focal length with the use of a collimation operation and masked with a pupil function $a\left(x, y\right)$, defined by the camera aperture diameter D.

$$p\left(x, y\right) = a\left(x, y\right) U_0\left(x, y\right) exp\left[\frac{-i\pi\left(x^2+y^2\right)}{\lambda R}\right]$$



(a) (b)

Fig. 4: (a) Example Modified Von Karman Phase Screen with subharmonics (b) PSF after propagation and lens functions.

where $p\left(x, y\right)$ represents the amplitude distribution behind the lens. The final PSF can then be found using Fourier optics principles [28] as

$$h(x, y) = \left(\left|\mathrm{FT}\{p(x, y)\}\right|^2\right)\Big|_{u=\frac{x}{\lambda l}, v=\frac{y}{\lambda l}}$$

which is then resampled to Nyquist sample spacing [6], and normalised to have a sum of 1. An example PSF is shown in Fig. 4b. PSFs are obtained for each pixel in the image, allowing the final turbulent image to then be formulated as

$$y\left[m, n\right] = \sum_j \sum_i x\left[i, j\right] \cdot h_{m,n}\left[m - i, n - j\right]$$

where x is the clean input image.

A performance gain for the simulation process is possible at this stage. In fact, due to the nature of the pencil rays, a pixel will have a very similar path through the atmosphere as its immediate neighbours. Therefore, a skip parameter allows the pencil rays to be traced using a sparser grid of pixels. The PSFs of the remaining pixels are then estimated using bilinear interpolation.

C. Dataset Generation

The dataset generated for this work makes use of the Places dataset [29]. This is a dataset of 1,469,737 scene images covering 205 separate categories, of which 31 have been chosen as categories that could be prone to turbulent interference, such as outdoor scenes. By isolating the data to these categories, a total of 148,884 images were selected for the simulation process.

The simulation parameters used for the resultant dataset are detailed in Table I, where the resulting grid spacing and sample count of the cropped phase screens were $\Delta x = 0.0064$, $N = 64$. To ensure a comprehensive dataset of diverse data, three variables were used during the simulation process. For each video, the values of C_n^2, turbulence speed and turbulence direction were randomly selected from a predefined list of values, shown in Table II. Such values allow for a total of 160 different potential turbulent simulations within the dataset.

TABLE I: Simulator settings for provided dataset.

Description	Dataset Values
Camera aperture diameter	$D = 0.1m$
Propagation distance	$L = 5km$
Number of phase screens	$nscr = 8$
Inner scale	$l_0 = 0.01m$
Outer Scale	$L_0 = 300m$
Pixel Skip	$skip = 4$
wavelength of light	$\lambda = 525 \times 10^{-9}$
Image Size	Image_Pixels $= 257 \times 257$
Video frames	frame_count $= 15$
No. of Subharmonics	Sub_Count $= 2$
Subharmonic grid size	$N_{subharm} = 4$

TABLE II: Dataset variables.

Variable	Values
C_n^2	$(0.25, 0.6525, 0.875, 1.1875, 1.5) \times 10^{-15}$
Turbulence Speed	$1, 2, 3, 4$
Turbulence Direction	$\rightarrow \searrow \downarrow \swarrow \leftarrow \nwarrow \uparrow \nearrow$

The random choice is made such that the final dataset has a uniform distribution of these 160 different classes. The range of values for the atmospheric structure parameter C_n^2 is chosen such that a range of low to high turbulence is represented, the two extremes of which are shown in Fig.1. The speed and direction of the turbulence are given as an integer value and an angle of movement. Each details the nature of movement of the enlarged phase screens in between each frame of the simulation. The speed denotes how many pixels the screens are to move, whilst the direction provides the angle in which the screens should translate. This can result in slow to fast turbulence in all the cardinal directions as well as diagonals.

IV. Conclusion

This paper has presented the details of an accurate method of turbulence simulation for optical images. This simulator is capable of imparting the realistic effects of turbulence onto datasets of clean images. Such data can then be further utilised in the development of turbulence mitigation algorithms. This paper also presents a dataset that is available for public use, allowing the slow process of propagation simulation to be avoided.

References

[1] A. Tunick, N. Tikhonov, M. Vorontsov, and G. Carhart, "Characterization of optical turbulence (cn2) data measured at the arl a_lot facility," tech. rep., US Army Research Laboratory Adelphi United States, 2005.

[2] D. Vint, G. Di Caterina, J. Soraghan, R. Lamb, and D. Humphreys, "Analysis of deep learning architectures for turbulence mitigation in long-range imagery," in *Artificial Intelligence and Machine Learning in Defense Applications II*, vol. 11543, p. 1154303, SPIE, 2020.

[3] N. Anantrasirichai, A. Achim, N. G. Kingsbury, and D. R. Bull, "Atmospheric turbulence mitigation using complex wavelet-based fusion," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2398–2408, 2013.

[4] N. Anantrasirichai, A. Achim, and D. Bull, "Mitigating the effects of atmospheric distortion on video imagery: A review," 2011.

[5] J. D. Schmidt, *Numerical Simulation of Optical Wave Propagation with examples in MATLAB*. SPIE, 2010.

[6] R. C. Hardie, J. D. Power, D. A. LeMaster, D. R. Droege, S. Gladysz, and S. Bose-Pillai, "Simulation of anisoplanatic imaging through optical turbulence using numerical wave propagation with new validation analysis," *Optical Engineering*, vol. 56, no. 7, pp. 071502–071502, 2017.

[7] G. Chen, Z. Gao, Q. Wang, and Q. Luo, "U-net like deep autoencoders for deblurring atmospheric turbulence," *Journal of Electronic Imaging*, vol. 28, no. 5, pp. 053024–053024, 2019.

[8] M. Van Iersel and A. M. Van Eijk, "Estimating turbulence in images," in *Free-Space Laser Communications X*, vol. 7814, pp. 197–206, SPIE, 2010.

[9] W. H. Chak, C. P. Lau, and L. M. Lui, "Subsampled turbulence removal network," *arXiv preprint arXiv:1807.04418*, 2018.

[10] J. Gao, N. Anantrasirichai, and D. Bull, "Atmospheric turbulence removal using convolutional neural network," *arXiv preprint arXiv:1912.11350*, 2019.

[11] M. C. Roggemann, B. M. Welsh, and B. R. Hunt, *Imaging through turbulence*. CRC press, 1996.

[12] J. P. Bos and M. C. Roggemann, "Technique for simulating anisoplanatic image formation over long horizontal paths," *Optical Engineering*, vol. 51, no. 10, pp. 101704–101704, 2012.

[13] N. Chimitt, Z. Mao, G. Hong, and S. H. Chan, "Rethinking atmospheric turbulence mitigation," *arXiv preprint arXiv:1905.07498*, 2019.

[14] J. Beck, C. Bekins, and J. P. Bos, "Wavepy: a python package for wave optics," in *Long-Range Imaging*, vol. 9846, pp. 8–17, SPIE, 2016.

[15] E. Repasi and R. Weiss, "Analysis of image distortions by atmospheric turbulence and computer simulation of turbulence effects," in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XIX*, vol. 6941, pp. 256–268, SPIE, 2008.

[16] E. Repasi and R. Weiss, "Computer simulation of image degradations by atmospheric turbulence for horizontal views," in *Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXII*, vol. 8014, pp. 279–287, SPIE, 2011.

[17] R. Nieuwenhuizen and K. Schutte, "Deep learning for software-based turbulence mitigation in long-range imaging," in *Artificial Intelligence and Machine Learning in Defense Applications*, vol. 11169, pp. 153–162, SPIE, 2019.

[18] C. Zhang, B. Xue, F. Zhou, and W. Xiong, "Removing atmospheric turbulence effects in unified complex steerable pyramid framework," *IEEE Access*, vol. 6, pp. 75855–75867, 2018.

[19] M. Hirsch, S. Sra, B. Schölkopf, and S. Harmeling, "Efficient filter flow for space-variant multiframe blind deconvolution," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 607–614, IEEE, 2010.

[20] H. van der Elst and J. van Schalkwyk, "Modelling and restoring images distorted by atmospheric turbulence," in *Proceedings of COMSIG'94-1994 South African Symposium on Communications and Signal Processing*, pp. 162–167, IEEE, 1994.

[21] J. Gilles and N. B. Ferrante, "Open turbulent image set (otis)," *Pattern Recognition Letters*, vol. 86, pp. 38–41, 2017.

[22] N. Anantrasirichai, A. Achim, and D. Bull, "Atmospheric turbulence mitigation for sequences with moving objects using recursive image fusion," in *2018 25th IEEE international conference on image processing (ICIP)*, pp. 2895–2899, IEEE, 2018.

[23] R. K. Saha, E. Salcin, J. Kim, J. Smith, and S. Jayasuriya, "Turbulence strength c n2 estimation from video using physics-based deep learning," *Optics Express*, vol. 30, no. 22, pp. 40854–40870, 2022.

[24] L. C. Andrews, *Laser Beam Propagation Through Random Media*. SPIE, 2005.

[25] A. McFadden, "Creating a simulator of visual turbulence in long distance imaging," Master's thesis, University of Strathclyde, 2019.

[26] A. N. Kolmogorov, "The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers," *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, vol. 434, no. 1890, pp. 9–13, 1991.

[27] N. Chimitt and S. H. Chan, "Simulating anisoplanatic turbulence by sampling intermodal and spatially correlated zernike coefficients," *Optical Engineering*, vol. 59, no. 8, pp. 083101–083101, 2020.

[28] J. W. Goodman, *Introduction to Fourier optics*. Roberts and Company publishers, 2005.

[29] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," *Advances in neural information processing systems*, vol. 27, 2014.